PHANTOM GAINS: AUDITING SELF-IMPROVEMENT AGAINST A MEASURED NULL

Anonymous authors
Paper under double-blind review

ABSTRACT

Whether a language model has improved itself is increasingly judged not by mean accuracy but by which individual problems it gains and loses. Tracking these transitions means differencing two noisy estimates, leaving them vulnerable to measurement artifacts. Auditing three rounds of rank-32 LoRA self-training on Qwen3-8B against a frozen control pushed through the identical pipeline, we identify seven measurement failures, each of which inverts a reported finding when its control is absent. Several are standard practice. A ledger built on a single greedy decode manufactures capability changes on an untrained model, largely an artifact of inference batching; the expansion statistic separating acquisition from sharpening assigns that same model a rate of 0.280. The natural threshold repair does not survive replication: estimated across the frozen comparisons such a design already contains, its null stays non-zero. We replace it with a per-problem exact test against a pooled baseline under false-discovery-rate control, which detects nothing on any held-out replicate and is unchanged under the multiple-testing rule, error rate and pool size. Applied to a ladder of arms matched in stream, volume and evaluation, the audit finds that external distillation improves problems the base model rarely reaches while three forms of self-training do not; a regression rejects this asymmetry as a by-product of distillation's larger overall gain ($p < 10^{-8}$). On the far smaller set of problems the base model never reaches, the evidence is inconclusive, while self-training corrupts problems solved at baseline at rates well above the measured floor. Transition-level auditing therefore requires a separately measured null for every statistic it reports: nulls that cost no new experiments, built from baseline replicates a multi-arm study already owns, though not from as few as most possess. Code and evaluation artifacts are included as supplementary material.

1 INTRODUCTION

A language model that improves itself without new supervision would be a considerable thing, and a growing family of methods reports exactly that (Fang et al., 2025; Tao et al., 2024). STaR fine-tunes on its own correct rationales (Zelikman et al., 2022); TTRL replaces labels with a majority vote over its own samples (Zuo et al., 2025); more elaborate schemes generate their own curricula (Zhao et al., 2025; Chen et al., 2025; Huang et al., 2026) or learn to edit their own weights (Zweiger et al., 2025).

The evidence offered used to be a single number: accuracy before, accuracy after (Atreja, 2025). That is now widely accepted to be insufficient, since an aggregate cannot distinguish a problem the model newly learned to solve from one it could already solve on a good day (Yu et al., 2026b; Yue et al., 2025), nor show what was lost while the mean rose. A second generation of analyses has therefore moved to individual problems, tracking which enter and leave a model's reachable set (Yuan et al., 2026; Wu et al., 2026; Mayilvahanan et al., 2026) and counting those destroyed under self-training (Lin et al., 2026; Yu et al., 2026a).

This is the right move, and it creates a new problem. A transition is a difference between two estimates, each carrying sampling error, and the events of interest sit exactly where that error bites hardest. Auditing self-training on Qwen3-8B across three evaluation sets, we isolate *seven* measurement failures, each of which produces a confident and wrong conclusion wherever its control is absent (Figure 1). Two are standard practice, and one lies inside the correction for another.

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.

A single decode is not a state. Temperature-zero decoding is not deterministic under batching (He & Lab, 2025), so a ledger built on one greedy sample per problem gives an untrained model, re-evaluated against itself, 6 apparent learnings and 9 apparent corruptions. That is a corruption-to-learning ratio of 1.5: the shape of the alarming result this literature reports, manufactured out of nothing. Serializing the requests removes three-quarters of those flips and confirms the mechanism, but not all of them: even one request at a time, a frozen model changes 2% of its greedy verdicts.

An expansion statistic needs its own null, and one measurement of a null is not a null. Following Mayilvahanan et al. (2026), a problem counts as *expanded* when the base model produced no correct sample in k draws and the trained model produces at least one. That rule compares two noisy binomial estimates through asymmetric thresholds, and what it returns when nothing has changed has not been measured. It returns 0.280: a frozen Qwen3-8B evaluated twice at k=128 "expands" 7 of the 25 AIME problems it had not reached. All seven are single lucky samples, which appears to license requiring $m \geq 2$ successes and so to drive the null to zero. It does not: that conclusion rests on one frozen comparison, and over the 110 our experiments contain the null is $0.058\,[0.038, 0.078]$, larger than the value it was supposed to certify. The repair is to stop thresholding: a per-problem exact test against a baseline pooled over every independent evaluation of the untrained model gives all arms a common denominator, returns no detection on any held-out replicate, and holds under every multiple-testing rule we try.

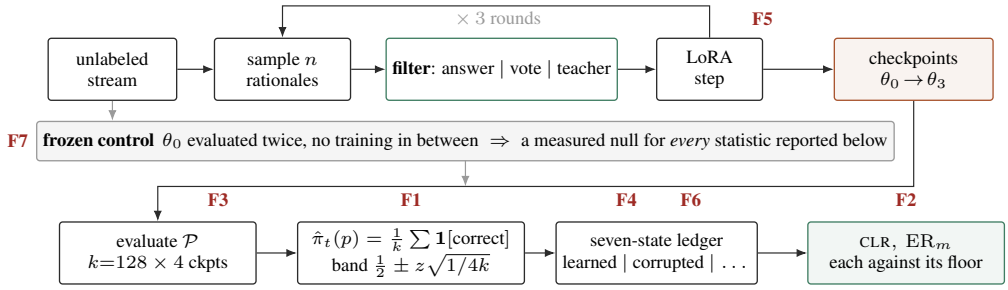
Applying the controlled audit, with distillation from a stronger external teacher as a positive control matched in stream, retained volume and evaluation, we find a dissociation. Distillation improves 8–11 of the 22 problems the base model reaches at most five times in 1,408 draws, against 0–2 for three forms of self-training, and a logistic model rejects the reading that this merely follows from the teacher's larger overall gain ($\beta = 1.91$, $p < 10^{-8}$). We are careful about what this licenses: on the ten problems the base model never reaches at all the comparison is not significant, so the claim is about rarely-reached problems and not about expansion. It aligns with the sharpening account of Huang et al. (2025); Yue et al. (2025) and with Strozzi (2026), on a different method family and with a positive control neither runs. We also quantify what is destroyed: both methods corrupt 88–106 of 1,163 band problems against a design-matched floor of 8, over half of those events larger than any solve-rate change a frozen model produced across twenty independent comparisons.

Contributions. (i) Seven measurement failure modes for transition-level audits, each demonstrated on a run where it reverses a conclusion; the last arises inside the correction for the second (Section 4). (ii) A threshold-free replacement for the expansion statistic, whose null we measure rather than assume and whose comparisons survive the multiple-testing rule, the error rate and the pool size (Section 4.2). (iii) The observation that makes measured nulls nearly free, namely that every arm's checkpoint 0 is an independent evaluation of the untrained model, together with the replicate count at which one becomes trustworthy, which is more than a three-arm study owns. (iv) A transition ledger with a benchmark-specific noise floor and effect-size accounting that separates threshold flicker from real capability loss, and a policy-gradient arm on which three seeds disagree about when the model collapses but not that it does (Section 5).

2 Related Work

What self-improvement can add. Huang et al. (2025) model self-improvement as *sharpening*: the model acts as its own verifier and concentrates mass on sequences it already assigns high likelihood, improving first-sample accuracy without introducing anything new; Song et al. (2025) formalize the same limit through a generation–verification gap. Empirically, Yue et al. (2025) report that RLVR-trained models are outperformed by their base models at large k, Wu et al. (2026) argue such training cannot place mass outside the base support, and Yuan et al. (2026) trace the effect to diversity collapse, while Liu et al. (2025a) argue the opposite for prolonged training. MATH-Beyond (Mayilvahanan et al., 2026) turns the debate into a measurement and supplies the expansion statistic we audit in Section 4.2; Qi et al. (2026) benchmark closed-loop self-evolution against oracle supervision and find a persistent gap.

Concurrent work on transition-level analysis. Strozzi (2026) ask our headline question, whether teacher-free self-training acquires capability or only expresses existing capability, and report a pass@K crossover on a DSL domain with a free verifier. We read our result as a replication in a



F1 a single decode in place of the estimator: the frozen model scores CLR=1.5. **F2** a one-success expansion threshold: the frozen model "expands" 0.280 of its unreached problems. **F3** a fixed token cap under style drift: the most effective arm is scored the most destructive. **F4** transition metrics carry ∼10× less power than accuracy: a 112-problem pilot cannot resolve the comparison. **F5** training-seed variance: CLR spans 0.55–1.53 for one method. **F6** an underpowered probe: a phantom 10-point safety drop. **F7** a null measured once: the "corrected" expansion threshold has a null of 0.058, not 0.

Figure 1: **The audit pipeline, and the seven stages at which a measurement failure enters it.** *Top:* an evolution arm draws candidates from an unlabeled stream, filters them by its own criterion (ground-truth answer for STaR, majority vote for the self-training arm, an external teacher for the distillation control) and takes a LoRA step, for three rounds. *Middle:* a frozen copy of θ_0 is pushed through the identical evaluation path, so every statistic is read against a null measured on the same problems with a model that did not train. *Bottom:* evaluation draws k=128 samples per problem per checkpoint, converts them to a solve rate rather than a single decode, discretizes with hysteresis, and reads off a seven-state ledger. F7 sits on the control path itself: the null each statistic is read against is an estimate, and measuring it once is the same error one level up. Every mode is measured here, inverts a reported conclusion when uncontrolled, and is caught by a control at the stage marked (Section 4).

different domain and supervision regime, with a positive control they do not run. Yuan et al. (2026) track per-problem boundary entry and exit across training, structurally our ledger, under an *analytic* binomial null. Ours is measured, and we can now say what that buys: for the estimator we recommend the two agree closely, and they part company where the noise is not binomial, which is how we found the expansion statistic's null to be non-zero at any threshold below m=5 (Appendix E.2).

Everything else. Appendix E.1 places this work against the self-evolution methods it audits, the degradation and model-collapse literature that motivates the corruption half, and the statistical-evaluation work our controls build on.

3 A TRANSITION-LEVEL AUDIT, AND WHAT IT MUST CONTROL FOR

The ledger. Fix an evaluation set $\mathcal{P}$ and checkpoints $t = 0, \ldots, T$, with $t = 0$ the untrained base model. For each problem p and checkpoint we draw k samples and record the solve rate $\hat{\pi}_t(p)$. Discretizing into a binary state $s_t(p)$ gives a trajectory per problem, and every problem falls into exactly one of seven categories: *stable-correct*, *stable-incorrect*, *learned*, *corrupted*, *recovered*, *transient*, *oscillating*. These partition $\mathcal{P}$, so counts need no normalization. The headline quantity is the **corruption-to-learning ratio**,

$$\mathrm{CLR} = \frac{|\{p : s_0(p) = \mathsf{solved},\ s_T(p) = \mathsf{unsolved}\}|}{|\{p : s_0(p) = \mathsf{unsolved},\ s_T(p) = \mathsf{solved}\}|}, \tag{1}$$

problems destroyed per problem gained; CLR > 1 removes more capability than it adds, whatever happens to mean accuracy. Equation 1 is undefined when nothing is learned, as on AIME, where we report the counts instead. We define it on endpoints, and report trajectory-based counts alongside (Appendix C.1); they agree to within three problems.

Correctness as an estimator, not a draw. The obvious definition of $s_t(p)$ is a single greedy sample, which modern inference does not support (Section 4.1). We instead treat $\hat{\pi}_t(p)$ as an estimator

of the model's pass@1 probability and discretize with hysteresis:

$$s_t(p) = \begin{cases} \textsf{solved} & \hat{\pi}_t(p) \geq \theta_{\text{hi}}, \\ \textsf{unsolved} & \hat{\pi}_t(p) < \theta_{\text{lo}}, \\ s_{t-1}(p) & \text{otherwise}, \end{cases} \qquad \theta_{\text{hi/lo}} = \tfrac{1}{2} \pm z\sqrt{\tfrac{1}{4k}}, \tag{2}$$

with $z = 2$, $k = 128$ throughout, giving a band of $[0.41, 0.59]$. The half-width is the worst-case standard error of a binomial proportion at k samples, so a problem must move further than ordinary jitter before it is recorded as changing state. The band degenerates at $k \leq z^2$, where it covers all of $[0, 1]$ and no problem can change state.

The baseline state carries no hysteresis. Equation 2 is undefined at $t = 0$, where there is no previous state to hold; we assign $s_0(p) = \textsf{solved}$ iff $\hat{\pi}_0(p) \geq \frac{1}{2}$. A problem whose baseline lies inside the band is therefore unprotected at that one checkpoint, which is why the floors of Table 10 are not zero on a set whose mean solve rate is 0.552 (Appendix C.1).

Expansion, with an explicit evidence threshold. Whether a newly solved problem represents new capability is a separate question that pass@1 cannot answer. Following Mayilvahanan et al. (2026), a problem that becomes pass@1-correct while the base model already solved it at least once in k samples has been *sharpened*; one the base model never reached has been *expanded*. We make the threshold m explicit:

$$\text{ER}_m = \frac{|\{p : \hat{\pi}_0(p) = 0 \ \wedge \ k\hat{\pi}_T(p) \geq m\}|}{|\{p : \hat{\pi}_0(p) = 0\}|}. \tag{3}$$

Prior work uses $m = 1$ implicitly; Section 4.2 shows that no threshold repairs this statistic and replaces it. Note what $\hat{\pi}_0(p) = 0$ does not mean: no successes in 128 draws bounds the base pass@1 probability at $q \leq 0.023$ with 95% confidence, it does not establish $q = 0$. We write "base-unreached at k" rather than "unsolvable" throughout, and every claim below is about what a stated sampling budget reaches, not about the support of the base distribution.

A control for every statistic. Transitions are differences between noisy measurements, so some appear when nothing has changed. We measure how many by re-evaluating a *frozen* model at every checkpoint and computing every statistic exactly as for a trained one; each transition it reports is an artifact by construction. No claim is admitted unless its counts exceed this floor. That floor is measured separately per statistic and per benchmark, and, as Section 4.3 establishes, with an interval and a matched number of checkpoints.

Benchmarks. Corruption is observable only on problems the model already solves, expansion only on problems it does not reach; the two populations are disjoint, so we use three sets (Figure 4). **MATH-500** (Lightman et al., 2024), subsampled to 200 problems by a fixed stratified draw over (subject, level), is thoroughly contaminated and hosts corruption rather than expansion. **AIME 2025 and 2026** (60 problems) are the least contaminated sets available to us, and AIME 2026 post-dates the backbone's March 2025 cutoff (Yang et al., 2025) outright, so we report that half separately wherever a claim rests on it. The base model reaches 0.175–0.180 of AIME, leaving 22 problems it reaches at most five times in 1,408 pooled draws (only 10 of them never) and 8 that can be corrupted. A **difficulty band** of 1,163 problems, profiled from 10,999 held-out MATH problems (Hendrycks et al., 2021) at k=8, is enriched near the decision boundary so that corruption is observable at all; its counts are therefore not a forgetting rate for a natural distribution, and it contains no base-unreached problems by construction (Appendix B.1).

We audit **STaR** (Zelikman et al., 2022) (filter by ground-truth answer), **majority-vote self-training** in the manner of TTRL (Zuo et al., 2025) (filter by majority vote over 8 samples, pooled by mathematical equivalence so $1/2$ and 0.5 do not split a vote), and **distillation** from `gpt-oss-120b` (OpenAI et al., 2025) as a positive control filtered by exactly STaR's criterion. All arms share the backbone (Qwen3-8B, Yang et al., 2025), rank-32 LoRA, three rounds of 256 problems from one stream, and evaluation at k=128, T=0.8, top-p 0.95 plus one greedy sample per problem per checkpoint. The stream is verified disjoint from every evaluation set by identifier and by normalized text hash, and answers are compared by symbolic equivalence (Kydlíček, 2025). Appendices A.1 and A.2 give the configuration and the grading rules in full.

Table 1: **Seven measurement failures, the conclusion each would have supported, and the control that caught it.** Every row is measured on a run from this study rather than posited, and F7 is the failure that arises inside the correction for F2. Cost is the marginal spend on the control; three of the seven cost nothing at all, being re-analyses of records the study already holds, and two more would cost nothing in any study that reports more than one arm.

	failure	would have concluded	control that catches it	cost
F1	single-decode ledger	"a frozen model has CLR $= 1.5$"	frozen floor + estimator	\$27
F2	one-success ER threshold	"maj.-vote expands 0.143 of AIME"	frozen floor for ER	\$0
F3	fixed token cap	"distillation is the most destructive"	log truncation per ckpt	\$0
F4	underpowered ledger	"the two methods do not differ"	prior power analysis	\$88
F5	single training seed	"maj.-vote corrupts less than STaR"	≥ 3 seeds	\$282
F6	underpowered probe	"self-training erodes refusal by 10 pts"	size the probe first	\$2
F7	a null measured once	"ER_2 has a null of zero"	every baseline is a replicate	\$0

Two deviations from TTRL as published. TTRL is reinforcement learning with a majority-vote reward, whereas the arm labeled *majority-vote SFT* below is cross-entropy fine-tuning on the rationales that agree with the vote. The two differ in what happens to the samples that lose the vote, pushed down under the first and discarded under the second. We therefore also run a genuine policy-gradient arm (Section 5) and confine claims about a second method family to it. TTRL is also normally applied to the unlabeled target test set, which trains on the problems we score. That on-set configuration is reported throughout as an unmatched baseline, and every claim rests on the matched ladder of Section 5, whose training stream is disjoint from the evaluation set.

4 Seven ways to mismeasure self-improvement

Each of the following is demonstrated on a run from this study and would, uncorrected, support a confident and incorrect conclusion; Table 1 summarizes them. The seventh arises *inside the correction for the second*, which is why we state it as a failure mode in its own right rather than folding it into the discussion.

4.1 F1. Without a floor, a frozen model appears to corrupt

Under a single greedy sample, an untrained model re-evaluated twice on MATH-500 shows 9 corruptions against 6 learnings, a CLR of 1.5. The solve-rate estimator on the same samples reports 1 and 0. The ratio is two single-digit counts, so we report the rate alongside it: $7.5\%\,[4.6, 12.0]$ of problems change verdict, itself unstable across three independent executions of the same two-pass comparison at 7.5%, 8.0% and 4.5%.

The mechanism, tested rather than cited. These flips are conventionally attributed to batching (He & Lab, 2025) without a direct test. One is available: sample the same 200 problems greedily twice over, once with 64 requests in flight and once strictly serialized. Serializing cuts verdict flips from $16/200$ to $4/200$ ($8.0\%\,[5.0, 12.6]$ against $2.0\%\,[0.8, 5.0]$, Fisher $p = 0.005$) and extracted-answer changes from 25 to 5 ($p < 10^{-4}$). Batching is therefore the dominant cause, but not the only one. A fully serialized frozen model still flips 2% of its greedy verdicts, enough to fabricate a CLR from nothing on a small evaluation set. No care with our own request pattern removes it: whatever remains is server-side batching across tenants or nondeterminism below the API. A single-decode ledger cannot be repaired by decoding more carefully; it has to be replaced by an estimator. On the $1{,}163$-problem band the gap is wider still, 18.9% of problems changing state under greedy decoding against 0.8% under the estimator (Figure 7, Table 10).

4.2 F2. No threshold repairs the expansion statistic

Equation 3 at $m = 1$ requires the base model to produce *zero* correct samples and the evolved model *at least one*. That is a comparison of two binomial estimates through asymmetric thresholds, and its null is not zero. On the frozen AIME control it gives $\mathrm{ER}_1 = 7/25 = \mathbf{0.280}$: a model that did not train "expands" more than a quarter of the problems it had not reached, and pooled over all 110

frozen comparisons the rate is 0.176. On the matched ladder of Section 5, STaR scores $\mathrm{ER}_1 = 0.320$ and majority-vote self-training 0.364, 0.167 and 0.192 across three seeds. Those values straddle the pooled null and lie inside the 0.000–0.364 range a frozen model produces across its own evaluations. The correct reading is not that they are nil, but that ER_1 overstates them by about a factor of two and cannot separate them from a model that never trained. It does separate distillation, at 0.545–0.667.

The obvious repair does not work. All seven frozen "expansions" are exactly one correct sample in 128, so $m = 2$ removes every one and its null on that comparison is exactly zero. The appearance rests on a single realization over 25 problems. Every arm's checkpoint 0 is an independent k=128 evaluation of the same untrained model, and with the ladder in place AIME carries *eleven* of them, giving 110 ordered pairs at no cost beyond experiments already run. Across those pairs ER_2 pools to $\mathbf{0.058}$ $[0.038, 0.078]$ and reaches 0.136 on the worst pair. The value we measure for majority-vote self-training, $1/21 = 0.048$, is therefore *indistinguishable* from its own null (Wilson $[0.008, 0.23]$) rather than below it. Raising the threshold does not rescue the statistic: ER_3 has a null of 0.023 $[0.009, 0.038]$ and only ER_5 approaches zero, at 0.003 $[0.000, 0.009]$; on MATH-500 the m=2 null is 0.128. The threshold that appeared to fix the statistic was fitted to the noise it was fitted on, and m is benchmark- and k-specific in exactly the way the statistic it corrects is.

What does work is to stop thresholding. Pool every independent evaluation of the untrained model into one baseline: eleven on AIME, $1{,}408$ draws per problem, one denominator shared by all arms rather than a noise-selected set per arm. Then test each problem's endpoint count against it with a one-sided Fisher exact test under FDR control. The evidence threshold disappears and the arms become comparable. Four choices remain, the error rate, the one-sidedness, the pool size and the strata, and Appendix D.1 varies each (Table 16). One belongs here, because a false-discovery-rate procedure is adaptive: an arm containing many true effects raises its own admissible cutoff, so detection counts need not be comparable across arms. Here they are: the cutoffs actually attained are 0.0095 for distillation's 36 detections and 0.0094 for majority-vote self-training's 19, and every comparison below is unchanged under Bonferroni, under one joint procedure across arms and problems, and at fixed thresholds. The null is measured the same way as everything else here: hold out one base evaluation, call it the "after", and run the identical test. It returns **zero** detections on all eleven held-out replicates, at every base rate (Figure 2, Table 19); zero is itself an estimate, and the Clopper–Pearson bounds are 0.24 per replicate and 0.005 per test.

Under that test the dissociation is sharper than any thresholded version. On the matched ladder, distillation detects 36 of 60 problems and majority-vote self-training 19; of the 21 they disagree about, 19 favor distillation, a paired exact $p = 2.2 \times 10^{-4}$ against the $p = 0.0015$ an unpaired test on mismatched denominators returns on the same data. Restricted to the 22 problems whose pooled base rate is at most $5/1{,}408$, distillation detects 8–11 per seed and the self-training arms 0–2, the frozen control 0 on all eleven replicates. We combine the per-seed comparisons rather than pooling detections across seeds, which is the operation F5 says three seeds cannot support: $p = 2.0 \times 10^{-6}$ against majority-vote SFT, 0.0078 against STaR and 0.0019 against the policy-gradient arm. Those 22 are not all expansion candidates, however. Only 10 have a pooled base rate of exactly zero, and there the union over seeds is 5 against 2, which is not significant ($p = 0.35$). Neither is the comparison on the ten low-base problems of the uncontaminated 2026 half (6 against 2, $p = 0.17$). Section 5 keeps the two populations apart.

The small non-zero counts are stated rather than rounded away: applied to a stream of problems it can solve, self-training does occasionally raise a rarely-reached problem above the noise, at roughly a fifth of the teacher's rate. The comparison is in any case conservative in the teacher's disfavor, since the distillation arms truncate 36–46% of completions even at an 8,192-token cap (Appendix D.1).

4.3 The remaining failures

Three of the remaining five are stated in summary and developed in Appendices C.3 and D.4. **F3**: a fixed token cap scores the most effective arm as the most destructive, because the distillation student adopts its teacher's verbose style and truncation rises from 16.5% to 65.3%; that arm's corruption counts are excluded from the capability analysis, since severe generation-length drift under a fixed cap confounds capability loss with stylistic verbosity. **F4**: transition metrics carry an order of magnitude less power than accuracy, and a 112-problem pilot yields 24 events against the

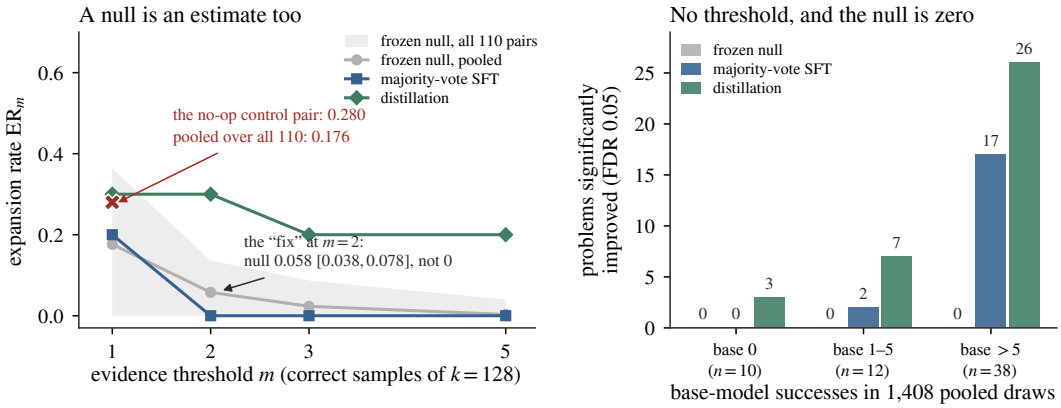


Figure 2: **The expansion statistic needs a null, and one measurement of a null is not one.** *Left:* ER_m against the evidence threshold. At $m{=}1$, the criterion in use, a single frozen comparison sits at 0.280, above majority-vote self-training. The shaded band is the range over all 110 frozen comparisons the records contain, and it does not approach zero until $m{=}5$: at $m{=}2$, the natural repair, the pooled null is 0.058 and the value it was meant to certify as null is 0.048. *Right:* the threshold-free test, by pooled base rate. Detections against a shared 1,408-draw baseline under FDR control; the frozen null is zero in every stratum on all eleven held-out replicates.

$\sim$249 per arm the comparison needs. **F6**: a 10-prompt refusal probe yields a 10-point safety finding that 50 prompts dissolve.

F5. The comparison between methods is not stable across training seeds. Two further seeds on a matched 175-problem subsample invert the single-seed comparison (Table 13). STaR is stable (CLR of 0.70, 0.74, 0.45, accuracy up $+5.6$ to $+6.9$ points on every seed) and majority-vote self-training is not: 0.55, 1.53, 1.33, with accuracy changes of $+4.0$, -2.8 and -1.0. A powered comparison at seed 0 finds no difference ($p = 0.406$); pooling the two matched replicates for a Fisher test gives $p = 0.0076$, and we do not report it, because pooling seeds as a fixed effect is what the same paragraph says three seeds cannot support. The defensible reading is narrower and more useful: *a majority-vote method's outcome is not determined by its design alone*. Same method, same data, same budget, run twice, landing on either side of CLR $= 1$.

F7. A null measured once is a point estimate, not a parameter. Everything above compares a measured quantity to a measured null, and a null estimated once carries the same sampling error one level up. This design admits three instances: ER_2's null of $0/25$ on a single frozen pair becomes $0.058\,[0.038, 0.078]$ over 110 frozen comparisons, the band's corruption floor of 5 becomes a median of 8 once the control is given as many checkpoints as the arms, and the 0.188 envelope on frozen solve-rate movement becomes 0.234 over twenty pairs. None moves a headline, but the first removes the correction F2 would otherwise license. The remedy needs no additional experiments, and generalizes: *every arm's checkpoint* 0 *is an independent evaluation of the untrained model*. It needs a number attached, and ours is that four is not enough: over every subset of our eleven baselines the ER_2 null still ranges 0.022–0.098 at $G = 4$, enough to certify a null that is not one, and only at $G = 9$ does it hold within ± 0.02 of the value eleven give.

5 What the controlled audit shows

A matched comparison. Compared under their native protocols, self-training and distillation differ in more than the correctness filter: TTRL as published is applied to the 60 AIME problems themselves and retains 50 examples per round, where distillation draws from a disjoint stream and retains 247, a $4.9\times$ volume gap on top of a stream mismatch. To eliminate the volume discrepancy and isolate the filter, we evaluate a ladder on one shared stream at matched volume, three seeds, so that only the correctness filter varies: the model's own majority vote, the ground-truth answer, an

Table 2: **The matched AIME ladder: one stream, one retained volume, one evaluation, and only the source of the correctness filter changing.** Detections are problems whose solve rate rises significantly against a pooled 1,408-draw baseline of the untrained model, pooled over 11 independent evaluations (per-problem exact test, FDR 0.05), split by how often that baseline reaches the problem at all. The frozen null is zero in every column. Expansion is only interpretable in the two low-base columns: there the self-training arms reach 0–2 problems and the external teacher 8–11, on every seed. Majority-vote accuracy also moves $+3.3$, -1.5 and $+0.7$ points across its three seeds while the teacher's moves $+13$ to $+16$: the instability of F5, on a second benchmark and under the matched protocol.

filter	update	seed	accuracy	Δ (pts)	retained	base 0	base 1–5	base > 5
majority vote	SFT	0	0.180 → 0.213	↑3.3	250	0	2	17
		1	0.178 → 0.163	↓1.5	251	1	0	3
		2	0.179 → 0.186	↑0.7	251	0	0	4
ground truth	SFT	0	0.177 → 0.195	↑1.8	220	1	1	9
external teacher†	SFT	0	0.180 → 0.310	↑13.0	247	3	7	26
		1	0.177 → 0.324	↑14.7	248	4	7	26
		2	0.177 → 0.335	↑15.8	246	2	6	25
majority vote	policy grad.‡	0	0.176 → 0.192	↑1.6	1984	0	1	11
frozen control	none	—	—	—	—	**0**	**0**	**0**

†positive control, not a self-evolution method. The frozen null is the same test with one base evaluation held out as the "after", over all 11 such replicates. ‡this arm collapsed into unbounded repetition on all three seeds and is 76.4% truncated at the checkpoint shown; its detections are retained as a floor, since truncation can only suppress them, and its transition counts are excluded (Appendix A.3). *Retained* is the first round's count (Table 3 gives all three): training examples for the SFT arms, rollouts for the policy-gradient arm, of which 135, 143 and 217 of ∼250 carried any gradient.

external teacher (Table 2), together with a policy-gradient arm using the same vote as a *reward*, as TTRL specifies.

The ladder yields three results the unmatched comparison cannot. First, the dissociation survives matching: on the low-base pool the teacher reaches 8–11 problems per seed and every self-training arm 0–2. On the ten problems of that pool the base model never reaches at all, however, the split over seeds is 5 against 2 and is not significant, so what we claim is a low-base result and not an expansion result. Second, the dissociation is not an artifact of the teacher's larger gain. A logistic model on the per-problem counts, with terms for arm, stratum and their interaction, gives the teacher a further $\beta = 1.91\,[1.25, 2.56]$ on low-base problems beyond its overall lift ($p < 10^{-8}$), where majority-vote SFT gets $0.21\,[-0.60, 1.03]$ and the policy-gradient arm -0.49. Uniform multiplicative improvement accounts for the self-training arms and not for the teacher (Appendix D.1). Third, the seed instability of F5 replicates on a second benchmark: majority-vote accuracy moves $+3.3$, -1.5 and $+0.7$ points across seeds, one of three net negative, while the teacher's moves $+13$ to $+16$ on all three.

Sharpening, at its true evidential weight. Across every self-training arm, all problems newly solved at pass@1 were already reachable at pass@k; the informative evidence is nine problems rather than four hundred, since the band excludes base-unreached problems by construction (Appendix C.1). The threshold-free test carries that claim at scale, and its null result is a bound rather than an absence. With eleven pooled baselines the test has 80% power against a true post-training rate of 0.024, and no self-training arm produced more than one correct sample in 128 on any base-0 problem. None lifted such a problem above roughly a 2.4% pass rate.

Reinforcement rather than filtering, and an arm outside the capability analysis. Both self-training arms above are rejection-sampling SFT, whereas TTRL as published pushes the losing samples down. A policy-gradient arm on the band gives 19 learned against 9 corrupted (CLR $= 0.47$) and behaves; on AIME it does not survive at any of the three seeds F5 demands. The runs see identical data in identical order and agree at round one to within a percentage point: the vote is 77.0–77.9% correct, and 135–136 groups of roughly 250 carry any gradient. They then diverge completely, collapsing into unbounded repetition after one update, two, and three, at 100%, 95.6% and 76.4% truncation. Every seed ends there, so the collapse belongs to the configuration and the

round at which it arrives does not; the arm's transition counts are excluded under F3 exactly as distillation's are. What it is reported for is its own diagnostic. On seed 2, between the update that damaged the model and the one that finished it, the majority vote became *more* accurate and *more* unanimous (77.9 to 85.1%, share 0.78 to 0.86, over 248 intact majorities) while the evaluation went from 47% truncated to 96%. A supervision signal that reads healthy while the model it supervises is dying is the failure this paper is about (Table 5, Appendix A.3).

What is removed is large, not a threshold flicker. On the band both methods destroy problems solved at baseline, 106 for STaR and 88 for majority-vote SFT, while STaR's mean accuracy rises by 5.6 points on that same run. Aggregate improvement and substantial destruction are the same run here, not alternatives. The floor needs the treatment the expansion null needed. The conventional two-checkpoint control gives 4 learned and 5 corrupted, where an arm has four checkpoints. A design-matched four-checkpoint no-op, built from the band's five independent evaluations, gives a median of 10 and 8 over 120 orderings (Figure 8), so the arms clear it by 11–13× rather than 18–21×. Nor is what they destroy a threshold flicker. Measured over all 20 frozen pairs, a model that never trained never moved a solve rate by more than 0.234. Yet 58 of STaR's 106 corruptions and 60 of majority-vote SFT's 88 exceed that: more than half, and more still under a stated quantile in place of the maximum, which is the less conservative choice (Table 23, Figure 10).

Two things we can only bound. The discretization rule moves the counts without moving the conclusion, but under a pure endpoint rule the two self-training methods coincide exactly and at $z = 0$ their order reverses, so we rest nothing on the between-method comparison (Appendix C.2). And the refusal probe cannot resolve what it was built to measure: three replicates read -0.015, -0.115 and -0.005, the middle one significant and replicated by neither neighbor, so we report a bound (any change here is smaller than about eleven points) rather than a null result (Appendix D.4).

6 Discussion

A reporting standard. Seven practices follow, each of which caught an inverted result here: report a no-op control for *every* statistic and *with an interval*; match the control's design to the arm's; define correctness through a solve-rate estimator rather than a single decode; prefer a per-problem test against a pooled baseline to any thresholded expansion statistic; log generation length per checkpoint; size transition comparisons from a prior power analysis; and report at least three training seeds, scored on the same problems. As our own policy-gradient arm shows, the third seed can be the one that changes the finding. Five need no new experiments, being re-analyses of records a careful study already holds, but none of them on four frozen replicates, which is not enough for the null they would be read against. We report roughly a dozen p-values across seven failure modes and correct within each family rather than across them, which is stated here rather than left to be noticed (Appendix D.1).

Limitations. Three rounds and roughly 270 optimizer steps of rank-32 LoRA, far short of the schedules we position against: what we establish is that *this* regime does not expand, not that self-training cannot. Expansion rests on 22 low-base problems on one benchmark, and on the 10 the base model never reaches the comparison is not significant. Corruption rests on a band drawn from MATH *training* problems almost certainly in the backbone's pretraining data and built to make corruption observable. One backbone family carries every claim (Appendix E.3).

Conclusion. Transition-level auditing is the right response to the fact that accuracy conceals what a self-improving model gains and what it destroys. It is also a measurement problem in its own right: we isolate seven ways to get it wrong, two of them standard practice and one inside the correction for another. The expansion statistic in current use assigns a model that never trained a rate of 0.280; the natural threshold repair takes that null to 0.058 rather than zero, and calculating it would have returned 0.044 and certified the same false conclusion. The replicates that bound a null already exist in any multi-arm study, provided there are enough of them.

Ethics Statement

This study evaluates language models on public mathematics benchmarks and releases only model-generated text about those problems; it involves no human subjects and no personal data. Two aspects warrant comment. The first is the out-of-domain refusal probe. Its prompts are deliberately mild and non-operational, asking for the shape of a harmful action rather than the means to carry one out, because their purpose is to elicit a refusal and not anything usable. The probe is released so that its result can be re-scored under a rule other than ours. What that result supports is a bound and not a clearance: we decline to claim that self-training causes no safety erosion here, and the data are consistent both with no effect and with one of practical size (Appendix D.4). Nothing in this paper should be read as evidence that a self-training method is safe to deploy without supervision. The second is that this paper reports, for every failure mode it documents, both the conclusion an uncontrolled analysis supports and the marginal cost of the control that rules it out. Presenting both is deliberate: a literature in which nulls are never re-measured is precisely the failure this paper is about. The distillation control uses an openly released teacher within the terms of its license, and Appendix A.4 reports the study's compute and monetary cost in full.

Reproducibility Statement

Records for every run are included as supplementary material, reduced to per-problem counts: for each checkpoint, sampling mode and problem, the samples drawn, the samples correct, the completions truncated, and the tokens and cost they consumed. Every number and figure in this paper is a function of those counts and is recomputed from them by released code; none is transcribed by hand. The per-sample records they are reduced from — one JSON line per generation, carrying the generated text and the extracted answer — total roughly 15 GB, are impractical to distribute and matter only for re-grading under a different verifier; they are available on request, and the reduction that produced the released files is released with them. To support re-scoring under alternative refusal rules, the out-of-domain refusal probe similarly records individual outputs per generation; those records are what establish that the probe's reading does not reproduce across replicates (Appendix D.4). The evaluation subsets are released with the code, including the constructed difficulty band and the selection profile used to build it, together with the noise-floor runs for each benchmark. Section 3 and Appendix A.1 state the training and sampling configuration in full, including temperature, top-p, token caps, learning rate, adapter rank and per-round training-example counts. The audited methods are reimplementations against Tinker[1], a hosted LoRA training and sampling service, and are released; the policy-gradient arm uses that service's importance-sampling objective on group-centered advantages, with the group mean as baseline, no clipping and no off-policy correction beyond a single epoch per round.

Use of Large Language Models

An LLM-based coding assistant was used to implement the evaluation harness and to run and monitor experiments, and assist with language editing and proofreading. All experimental design decisions, all interpretation of results, and all claims in this paper are the authors' own. Every number reported is computed by released code from released per-sample records; none was produced or transcribed by a language model. We have reviewed all AI-assisted work and take full responsibility for it.

References

Emre Can Acikgoz, Cheng Qian, Heng Ji, Dilek Hakkani-Tür, and Gokhan Tur. Self-improving llm agents at test-time, 2025. URL https://arxiv.org/abs/2510.07841.

Sina Alemohammad, Josue Casco-Rodriguez, Lorenzo Luzi, Ahmed Imtiaz Humayun, Hossein Babaei, Daniel LeJeune, Ali Siahkoohi, and Richard Baraniuk. Self-consuming generative models go mad. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 53581–53608, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/ebc042e767de551803ccfcc45e2454f5-Paper-Conference.pdf.

[1] https://thinkingmachines.ai/tinker/

Huan ang Gao, Jiayi Geng, Wenyue Hua, Mengkang Hu, Xinzhe Juan, Hongzhang Liu, Shilong Liu, Jiahao Qiu, Xuan Qi, Qihan Ren, Yiran Wu, Hongru WANG, Han Xiao, Yuhang Zhou, Shaokun Zhang, Jiayi Zhang, Jinyu Xiang, Yixiong Fang, Qiwen Zhao, Dongrui Liu, Cheng Qian, Zhenhailong Wang, Minda Hu, Huazheng Wang, Qingyun Wu, Heng Ji, and Mengdi Wang. A survey of self-evolving agents: What, when, how, and where to evolve on the path to artificial super intelligence. *Transactions on Machine Learning Research*, 2026. ISSN 2835-8856. URL https://openreview.net/forum?id=CTr3bovS5F. Survey Certification.

Dhruv Atreja. Alas: Autonomous learning agent for self-updating language models, 2025. URL https://arxiv.org/abs/2508.15805.

Dan Biderman, Jacob Portes, Jose Javier Gonzalez Ortiz, Mansheej Paul, Philip Greengard, Connor Jennings, Daniel King, Sam Havens, Vitaliy Chiley, Jonathan Frankle, Cody Blakeney, and John Patrick Cunningham. LoRA learns less and forgets less. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=aloEru2qCG. Featured Certification.

Lawrence D. Brown, T. Tony Cai, and Anirban DasGupta. Interval Estimation for a Binomial Proportion. *Statistical Science*, 16(2):101 – 133, 2001. doi: 10.1214/ss/1009213286. URL https://doi.org/10.1214/ss/1009213286.

Lai Chen, Yao Zhu, Xin Ye, Peng Lin, Xiangyang Ji, and Xiang Tian. Enhancing reasoning in large language models via entropy-aware self-evolution, 2026. URL https://openreview.net/forum?id=nXENWUSRMw.

Lili Chen, Mihir Prabhudesai, Katerina Fragkiadaki, Hao Liu, and Deepak Pathak. Self-questioning language models, 2025. URL https://arxiv.org/abs/2508.03682.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 6621–6642. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/chen24j.html.

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

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models, 2025. URL https://arxiv.org/abs/2505.22617.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping, 2020. URL https://arxiv.org/abs/2002.06305.

B. Efron. Bootstrap Methods: Another Look at the Jackknife. *The Annals of Statistics*, 7(1): 1 – 26, 1979. doi: 10.1214/aos/1176344552. URL https://doi.org/10.1214/aos/1176344552.

Jinyuan Fang, Yanwen Peng, Xi Zhang, Yingxu Wang, Xinhao Yi, Guibin Zhang, Yi Xu, Bin Wu, Siwei Liu, Zihao Li, Zhaochun Ren, Nikos Aletras, Xi Wang, Han Zhou, and Zaiqiao Meng. A comprehensive survey of self-evolving ai agents: A new paradigm bridging foundation models and lifelong agentic systems, 2025. URL https://arxiv.org/abs/2508.07407.

R. A. Fisher. On the interpretation of χ^2 from contingency tables, and the calculation of p. *Journal of the Royal Statistical Society*, 85(1):87–94, 1922. ISSN 09528385. URL http://www.jstor.org/stable/2340521.

Aryo Pradipta Gema, Joshua Ong Jun Leang, Giwon Hong, Alessio Devoto, Alberto Carlo Maria Mancino, Rohit Saxena, Xuanli He, Yu Zhao, Xiaotang Du, Mohammad Reza Ghasemi Madani, Claire Barale, Robert McHardy, Joshua Harris, Jean Kaddour, Emile Van Krieken, and Pasquale Minervini. Are we done with MMLU? In Luis Chiruzzo, Alan Ritter, and Lu Wang (eds.), *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5069–5096, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. ISBN 979-8-89176-189-6. doi: 10.18653/v1/2025.naacl-long.262. URL https://aclanthology.org/2025.naacl-long.262/.

Matthias Gerstgrasser, Rylan Schaeffer, Apratim Dey, Rafael Rafailov, Tomasz Korbak, Henry Sleight, Rajashree Agrawal, John Hughes, Dhruv Bhandarkar Pai, Andrey Gromov, Dan Roberts, Diyi Yang, David L. Donoho, and Sanmi Koyejo. Is model collapse inevitable? breaking the curse of recursion by accumulating real and synthetic data. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=5B2K4LRgmz.

Halil Alperen Gozeten, Muhammed Emrullah Ildiz, Xuechen Zhang, Mahdi Soltanolkotabi, Marco Mondelli, and Samet Oymak. Test-time training provably improves transformers as in-context learners. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 20266–20295. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/gozeten25a.html.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu,

Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645 (8081):633–638, September 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL http://dx.doi.org/10.1038/s41586-025-09422-z.

Mohsen Hariri, Amirhossein Samandar, Michael Hinczewski, and Vipin Chaudhary. Don't pass@k: A bayesian framework for large language model evaluation. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 148539–148579, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/f04edfc65463d020629673a4bc4c58e7-Paper-Conference.pdf.

Horace He and Thinking Machines Lab. Defeating nondeterminism in llm inference. *Thinking Machines Lab: Connectionism*, 2025. doi: 10.64434/tml.20250910. URL https://thinkingmachines.ai/blog/defeating-nondeterminism-in-llm-inference/.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In J. Vanschoren and S. Yeung (eds.), *Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks*, volume 1, 2021. URL https://datasets-benchmarks-proceedings.neurips.cc/paper_files/paper/2021/file/be83ab3ecd0db773eb2dc1b0a17836a1-Paper-round2.pdf.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network, 2015. URL https://arxiv.org/abs/1503.02531.

Arian Hosseini, Xingdi Yuan, Nikolay Malkin, Aaron Courville, Alessandro Sordoni, and Rishabh Agarwal. V-STar: Training verifiers for self-taught reasoners. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=stmqBSW2dV.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 8003–8017, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.507. URL https://aclanthology.org/2023.findings-acl.507/.

Edward J Hu, yelong shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=nZeVKeeFYf9.

Audrey Huang, Adam Block, Dylan Foster, Dhruv Rohatgi, Cyril Zhang, Max Simchowitz, Jordan Ash, and Akshay Krishnamurthy. Self-improvement in language models: The sharpening mechanism. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 76687–76739, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/bee8c2bc757f6bbc3efd7cf1b979f0c9-Paper-Conference.pdf.

Chengsong Huang, Wenhao Yu, Xiaoyang Wang, Hongming Zhang, Zongxia Li, Ruosen Li, Jiaxin Huang, Haitao Mi, and Dong Yu. R-zero: Self-evolving reasoning llm from zero data. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 130770–130790, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/d49b9aacebda61051166335af6fd3061-Paper-Conference.pdf.

Jiaxin Huang, Shixiang Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. Large language models can self-improve. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 1051–1068, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.67. URL https://aclanthology.org/2023.emnlp-main.67/.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Yu, Xinying Song, and Denny Zhou. Large language models cannot self-correct reasoning yet. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 32808–32824, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/8b4add8b0aa8749d80a34ca5d941c355-Paper-Conference.pdf.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. doi: 10.1073/pnas.1611835114. URL https://www.pnas.org/doi/abs/10.1073/pnas.1611835114.

Hynek Kydlíček. Math-verify: Math verification library. GitHub repository, 2025. URL https://github.com/huggingface/Math-Verify.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/aca97732e30bcf1303bc22ac3924fd16-Paper-Conference.pdf.

Hongxiang Lin, Zhirui Kuai, Erpeng Xue, and Lei Wang. Detecting and mitigating the correct-answer extinction window in test-time reinforcement learning with majority voting, 2026. URL https://arxiv.org/abs/2605.19444.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 17998–18031. Curran Associates, Inc., 2025a. doi: 10.52202/085713-0608. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/1a22b912945fb7c0bdd079e792b31b6f-Paper-Conference.pdf.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. In *Second Conference on Language Modeling*, 2025b. URL https://openreview.net/forum?id=5PAF7PAY2Y.

Jianqiao Lu, Wanjun Zhong, Wenyong Huang, Yufei Wang, Qi Zhu, Fei Mi, Baojun Wang, Weichao Wang, Xingshan Zeng, Lifeng Shang, Xin Jiang, and Qun Liu. Self: Self-evolution with language feedback, 2024. URL https://arxiv.org/abs/2310.00533.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *IEEE Transactions on Audio, Speech and Language Processing*, 33:3776–3786, 2025. doi: 10.1109/TASLPRO.2025.3606231. URL https://doi.org/10.1109/TASLPRO.2025.3606231.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback. In A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 46534–46594. Curran Associates, Inc., 2023. doi: 10.52202/

075280-2019. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/91edff07232fb1b55a505a9e9f6c0ff3-Paper-Conference.pdf.

Prasanna Mayilvahanan, Ricardo Dominguez-Olmedo, Thaddäus Wiedemer, and Wieland Brendel. Math-beyond: A benchmark for rl to expand beyond the base model. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 68471–68489, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/6fdf57c71bc1f1ee29014b8dc52e723f-Paper-Conference.pdf.

Evan Miller. Adding error bars to evals: A statistical approach to language model evaluations, 2024. URL https://arxiv.org/abs/2411.00640.

Mohammad Mahdi Moradi, Hossam Amer, Sudhir Mudur, Weiwei Zhang, Yang Liu, and Walid Ahmed. Continuous self-improvement of large language models by test-time training with verifier-driven sample selection. In *AI That Keeps Up: NeurIPS 2025 Workshop on Continual and Compatible Foundation Model Updates*, 2025. URL https://openreview.net/forum?id=6ahliSpvQ0.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20275–20321, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1025. URL https://aclanthology.org/2025.emnlp-main.1025/.

Sagnik Mukherjee, Lifan Yuan, Dilek Hakkani-Tur, and Hao Peng. Reinforcement learning finetunes small subnetworks in large language models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 132119–132138. Curran Associates, Inc., 2025. doi: 10.52202/085713-4399. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/bf235a1d6780afd979f2f81676f43413-Paper-Conference.pdf.

OpenAI, :, Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, Boaz Barak, Ally Bennett, Tyler Bertao, Nivedita Brett, Eugene Brevdo, Greg Brockman, Sebastien Bubeck, Che Chang, Kai Chen, Mark Chen, Enoch Cheung, Aidan Clark, Dan Cook, Marat Dukhan, Casey Dvorak, Kevin Fives, Vlad Fomenko, Timur Garipov, Kristian Georgiev, Mia Glaese, Tarun Gogineni, Adam Goucher, Lukas Gross, Katia Gil Guzman, John Hallman, Jackie Hehir, Johannes Heidecke, Alec Helyar, Haitang Hu, Romain Huet, Jacob Huh, Saachi Jain, Zach Johnson, Chris Koch, Irina Kofman, Dominik Kundel, Jason Kwon, Volodymyr Kyrylov, Elaine Ya Le, Guillaume Leclerc, James Park Lennon, Scott Lessans, Mario Lezcano-Casado, Yuanzhi Li, Zhuohan Li, Ji Lin, Jordan Liss, Lily, Liu, Jiancheng Liu, Kevin Lu, Chris Lu, Zoran Martinovic, Lindsay McCallum, Josh McGrath, Scott McKinney, Aidan McLaughlin, Song Mei, Steve Mostovoy, Tong Mu, Gideon Myles, Alexander Neitz, Alex Nichol, Jakub Pachocki, Alex Paino, Dana Palmie, Ashley Pantuliano, Giambattista Parascandolo, Jongsoo Park, Leher Pathak, Carolina Paz, Ludovic Peran, Dmitry Pimenov, Michelle Pokrass, Elizabeth Proehl, Huida Qiu, Gaby Raila, Filippo Raso, Hongyu Ren, Kimmy Richardson, David Robinson, Bob Rotsted, Hadi Salman, Suvansh Sanjeev, Max Schwarzer, D. Sculley, Harshit Sikchi, Kendal Simon, Karan Singhal, Yang Song, Dane Stuckey, Zhiqing Sun, Philippe Tillet, Sam Toizer, Foivos Tsimpourlas, Nikhil Vyas, Eric Wallace, Xin Wang, Miles Wang, Olivia Watkins, Kevin Weil, Amy Wendling, Kevin Whinnery, Cedric Whitney, Hannah Wong, Lin Yang, Yu Yang, Michihiro Yasunaga, Kristen Ying, Wojciech Zaremba, Wenting Zhan, Cyril Zhang, Brian Zhang, Eddie Zhang, and Shengjia Zhao. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 30988–31043, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/83b7da3ed13f06c13ce82235c8eedf35-Paper-Conference.pdf.

Zhenting Qi, Susanna Maria Baby, Stefanie Anna Baby, Kan Yuan, Andrew Tomkins, Tu Vu, Da-Cheng Juan, and Cyrus Rashtchian. On the generalization gap in self-evolving language model reasoning. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=mnUidYi5qO.

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. In *Forty-third International Conference on Machine Learning*, 2026a. URL https://openreview.net/forum?id=tqTNOpkP5j.

Shuai Shao, Qihan Ren, Dongrui Liu, Chen Qian, Boyi Wei, Dadi Guo, Jingyi Yang, Xinhao Song, Linfeng Zhang, Weinan Zhang, and Jing Shao. Your agent may misevolve: Emergent risks in self-evolving llm agents. In C. Vondrick, B. Hariharan, C. Raffel, L. Pinto, D. Yang, and A. Faust (eds.), *International Conference on Learning Representations*, volume 2026, pp. 99728–99793, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/file/a24cd16bc361afa78e57d31d34f3d936-Paper-Conference.pdf.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300.

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. doi: 10.1038/s41586-024-07566-y. URL https://www.nature.com/articles/s41586-024-07566-y.

Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, Aaron T Parisi, Abhishek Kumar, Alexander A Alemi, Alex Rizkowsky, Azade Nova, Ben Adlam, Bernd Bohnet, Gamaleldin Fathy Elsayed, Hanie Sedghi, Igor Mordatch, Isabelle Simpson, Izzeddin Gur, Jasper Snoek, Jeffrey Pennington, Jiri Hron, Kathleen Kenealy, Kevin Swersky, Kshiteej Mahajan, Laura A Culp, Lechao Xiao, Maxwell Bileschi, Noah Constant, Roman Novak, Rosanne Liu, Tris Warkentin, Yamini Bansal, Ethan Dyer, Behnam Neyshabur, Jascha Sohl-Dickstein, and Noah Fiedel. Beyond human data: Scaling self-training for problem-solving with language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=lNAyUngGFK. Expert Certification.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters, 2024. URL https://arxiv.org/abs/2408.03314.

Yuda Song, Hanlin Zhang, Carson Eisenach, Sham Kakade, Dean Foster, and Udaya Ghai. Mind the gap: Examining the self-improvement capabilities of large language models. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 39894–39931, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/63943ee9fe347f3d95892cf87d9a42e6-Paper-Conference.pdf.

Gaurav Srivastava, Zhenyu Bi, Meng Lu, and Xuan Wang. DEBATE, TRAIN, EVOLVE: Self-Evolution of language model reasoning. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 32764–32810, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1666. URL https://aclanthology.org/2025.emnlp-main.1666/.

Igor Lima Strozzi. Teacher-free self-training amplifies but does not compound: A pass@k crossover on a free-verifier domain, 2026. URL https://arxiv.org/abs/2606.07856.

Yifan Sun, Han Wang, Dongbai Li, Gang Wang, and Huan Zhang. The emperor's new clothes in benchmarking? a rigorous examination of mitigation strategies for LLM benchmark data

contamination. In *Forty-second International Conference on Machine Learning*, 2025a. URL https://openreview.net/forum?id=TuvDxubEfE.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, Tatsunori Hashimoto, and Carlos Guestrin. Learning to (Learn at test time): RNNs with expressive hidden states. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 57503–57522. PMLR, 13–19 Jul 2025b. URL https://proceedings.mlr.press/v267/sun25h.html.

Zhengwei Tao, Ting-En Lin, Xiancai Chen, Hangyu Li, Yuchuan Wu, Yongbin Li, Zhi Jin, Fei Huang, Dacheng Tao, and Jingren Zhou. A survey on self-evolution of large language models, 2024. URL https://arxiv.org/abs/2404.14387.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023a. URL https://openreview.net/forum?id=1PL1NIMMrw.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Du, and yelong shen. Reinforcement learning for reasoning in large language models with one training example. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 122721–122764. Curran Associates, Inc., 2025. doi: 10.52202/085713-4093. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/b1ea3f93167c55f3fc2999b68a170ff3-Paper-Conference.pdf.

Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. Self-instruct: Aligning language models with self-generated instructions. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 13484–13508, Toronto, Canada, July 2023b. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.754. URL https://aclanthology.org/2023.acl-long.754/.

Edwin B. Wilson. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927. doi: 10.1080/01621459.1927.10502953. URL https://www.tandfonline.com/doi/abs/10.1080/01621459.1927.10502953.

Fang Wu, Weihao Xuan, Ximing Lu, Mingjie Liu, Yi Dong, Zaid Harchaoui, and Yejin Choi. The invisible leash: Why rlvr may or may not escape its origin, 2026. URL https://arxiv.org/abs/2507.14843.

Cheng Xu, Shuhao Guan, Derek Greene, and M-Tahar Kechadi. Benchmark data contamination of large language models: A survey, 2024. URL https://arxiv.org/abs/2406.04244.

Cheng Xu, Nan Yan, Shuhao Guan, Changhong Jin, Yuke Mei, Yibing Guo, and Tahar Kechadi. DCR: Quantifying data contamination in LLMs evaluation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 23013–23031, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.1173. URL https://aclanthology.org/2025.emnlp-main.1173/.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang

Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, juncai liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 113222–113244. Curran Associates, Inc., 2025. doi: 10.52202/085713-3775. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/a4277440d50f1f15d2cb4c14f7e0c0d2-Paper-Conference.pdf.

Ye Yu, Xiaopeng Yuan, Haibo Jin, Heming Liu, Yaoning Yu, and Haohan Wang. Do self-evolving agents forget? capability degradation and preservation in lifelong llm agent adaptation, 2026a. URL https://arxiv.org/abs/2605.09315.

Zhuoyun Yu, Xin Xie, Wuguannan Yao, Chenxi Wang, Lei Liang, Xiang Qi, and Shumin Deng. Skilladaptor: Self-adapting skills for llm agents from trajectories, 2026b. URL https://arxiv.org/abs/2606.01311.

Suqin Yuan, Jinkun Chen, Jiyang Zheng, Muyang Li, Lei Feng, Dadong Wang, Tao Xiang, Tongliang Liu, and Bo An. Understanding diversity collapse in rlvr via the lens of overtraining, 2026. URL https://arxiv.org/abs/2606.15455.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason E Weston. Self-rewarding language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 57905–57923. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/yuan24d.html.

Zheng Yuan, Hongyi Yuan, Chengpeng Li, Guanting Dong, Keming Lu, Chuanqi Tan, Chang Zhou, and Jingren Zhou. Scaling relationship on learning mathematical reasoning with large language models, 2023. URL https://arxiv.org/abs/2308.01825.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 57654–57689. Curran Associates, Inc., 2025. doi: 10.52202/085713-1933. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/537d5aa768c2d534016a4d06f87bc8fb-Paper-Conference.pdf.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh (eds.), *Advances in Neural Information Processing Systems*, volume 35, pp. 15476–15488. Curran Associates, Inc., 2022. doi: 10.52202/068431-1126. URL https://proceedings.neurips.cc/paper_files/paper/2022/file/639a9a172c044fbb64175b5fad42e9a5-Paper-Conference.pdf.

Eric Zelikman, Georges Raif Harik, Yijia Shao, Varuna Jayasiri, Nick Haber, and Noah Goodman. Quiet-STar: Language models can teach themselves to think before speaking. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=oRXPiSOGH9.

Hugh Zhang, Jeff Da, Dean Lee, Vaughn Robinson, Catherine Wu, Will Song, Tiffany Zhao, Pranav Raja, Charlotte Zhuang, Dylan Slack, Qin Lyu, Sean Hendryx, Russell Kaplan, Michele Lunati, and Summer Yue. A careful examination of large language model performance on grade school arithmetic. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet,

J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 46819–46836. Curran Associates, Inc., 2024. doi: 10.52202/079017-1485. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/53384f2090c6a5cac952c598fd67992f-Paper-Datasets_and_Benchmarks_Track.pdf.

Andrew Zhao, Yiran Wu, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 105816–105879. Curran Associates, Inc., 2025. doi: 10.52202/085713-3534. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/9837dc00ff67d176373268ed48042d49-Paper-Conference.pdf.

Weixiang Zhao, Yingshuo Wang, Yichen Zhang, Yang Deng, Yanyan Zhao, Wanxiang Che, Bing Qin, and Ting Liu. Large language model agents are not always faithful self-evolvers. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=kTjSSqgqGf.

Yujun Zhou, Zhenwen Liang, Haolin Liu, Wenhao Yu, Kishan Panaganti, Linfeng Song, Dian Yu, Xiangliang Zhang, Haitao Mi, and Dong Yu. Evolving language models without labels: Majority drives selection, novelty promotes variation, 2026. URL https://arxiv.org/abs/2509.15194.

Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, yuchen zhang, Xinwei Long, Ermo Hua, Biqing Qi, Youbang Sun, Zhiyuan Ma, Lifan Yuan, Ning Ding, and Bowen Zhou. Ttrl: Test-time reinforcement learning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 131459–131483. Curran Associates, Inc., 2025. doi: 10.52202/085713-4376. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/be690ea16f005c174f6c4102a5970e67-Paper-Conference.pdf.

Adam Zweiger, Jyo Pari, Han Guo, Yoon Kim, and Pulkit Agrawal. Self-adapting language models. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 74084–74115. Curran Associates, Inc., 2025. doi: 10.52202/085713-2483. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/6b41e04c41726e2a60e456d0a2b961ab-Paper-Conference.pdf.

A EXPERIMENTAL PROTOCOLS AND IMPLEMENTATION DETAILS

This appendix fixes what every arm and every generation in the study was run under: the training and sampling configuration (§A.1), the prompt and the grading rules a completion passed through (§A.2), the policy-gradient objective and its three seeds (§A.3), and the compute the study consumed (§A.4).

A.1 TRAINING AND SAMPLING CONFIGURATION

The prompt. Every generation in this study, for evaluation and for the evolution stream alike, uses one template with no few-shot examples and no system message:

```
Solve the following math problem. Reason step by step, then
give your final answer inside \boxed{}.
Problem: {problem}
Solution:
```

The problem string is passed through unmodified, including any Asymptote figure source, which is why some MATH-500 geometry items carry a diagram listing into the context. Holding the prompt fixed across arms and checkpoints matters more here than choosing a good one: a transition is a difference between two evaluations, so anything that changes between them and is not the model is a confound.

Table 3 gives the configuration of every arm and the per-round statistics of its training signal. Three points bear on the fairness of the comparison.

Training volume is not matched on AIME by the obvious construction. On the difficulty band and MATH-500 the retained examples per round are within 7% across arms. On AIME the native protocols do not match: applied to the 60-problem evaluation set itself, as TTRL specifies, the majority-vote arm retains 50/51/50 against distillation's 247/246/247 drawn from a disjoint stream of 256 problems per round, a factor of 4.9. Training volume is therefore confounded with the filter for precisely the arms that carry the expansion result, which is what the AIME ladder of Section 5 removes: run on one shared stream of 256 problems per round, majority-vote self-training retains 246–254 examples per round across three seeds against distillation's 246–249, within 2%. STaR retains 205–220, because ground-truth filtering rejects more than a vote taken over the model's own samples can: a self-consistent vote almost always produces a majority, whether or not it is right, and on this stream 70–81% of those majorities are correct. That residual gap is a property of the filter rather than a design choice, it runs *against* the arm whose null we report, and we state it rather than tune it away.

The vote is a different instrument on a different stream. Pseudo-label accuracy, the fraction of majority votes that were in fact correct and a diagnostic that never influences training, is 91–95% on the difficulty band, 70–81% on the hard training stream the matched ladder uses, and 26–30% when the vote is taken on AIME itself. The unmatched on-set arm is not simply smaller; it operates in a supervision regime the others never enter. That gap is the subject of Appendix B.2.

Filtering and reinforcement are not the same update. The arms labeled SFT keep the generations that agree with their filter and discard the rest; the policy-gradient arm keeps all of them and centers the reward within each group, so disagreeing samples are pushed down rather than dropped. Group size, stream, volume and evaluation are otherwise identical. Its retained count is reported as rollouts rather than examples, and only groups with reward spread contribute: $1{,}976$, $1{,}920$ and $2{,}016$ rollouts drawn per round on the band, of which 76, 56 and 59 groups carried any gradient.

A.2 PROMPTING, GRADING AND ANSWER EXTRACTION

Answers are extracted from `\boxed{}` where present and from an explicit "the answer is" construction otherwise, then compared by symbolic equivalence (Kydlíček, 2025). We deliberately reject a third fallback that is common in practice: taking the last number appearing in the completion. On truncated reasoning that heuristic is close to a coin flip, and its errors are *random* rather than systematic. Random correctness is far more damaging to a transition metric than uniform strictness:

Table 3: Arm configuration and training-signal statistics. Evaluation is identical across arms within a benchmark: k=128, T=0.8, top-p 0.95, plus one greedy sample, with independent sampling calls. All arms use Qwen3-8B with rank-32 LoRA, $\eta = 10^{-4}$, batch 8, 3 rounds of 256 stream problems, and at most one training example per problem; SFT arms train 3 epochs per round and the policy-gradient arm one, since its rollouts are off-policy after a single pass. The AIME block above the rule is the matched ladder; the arm below it is the unmatched baseline that trains on the evaluation set itself.

arm	benchmark	gen. T	max tokens	samples/prob	retained per round	signal correct
STaR	MATH-500	0.9	2,048	4	229 / 227 / 231	100% (ground truth)
STaR	band	0.9	2,048	4	232 / 229 / 230	100% (ground truth)
maj.-vote SFT	MATH-500	1.0	2,048	8 votes	241 / 247 / 246	92.1/89.5/91.5%
maj.-vote SFT	band	1.0	2,048	8 votes	247 / 246 / 240	92.7/91.1/94.6%
Distillation	band	0.7	2,048	4	232 / 239 / 234	100% (filtered)
matched AIME ladder: one stream (`math-train-hard`), one volume, three seeds						
maj.-vote SFT	AIME	1.0	8,192	8 votes	250 / 250 / 246	78.0/77.6/80.5%
STaR	AIME	0.9	8,192	4	220 / 205 / 205	100% (ground truth)
Distillation	AIME	0.7	8,192	4	247 / 246 / 247	100% (filtered)
maj.-vote RL	AIME	1.0	8,192	8 rollouts	†	†
maj.-vote RL	band (175)	1.0	2,048	8 rollouts	†	†
unmatched baseline (on-set training): unmatched in stream and volume, the reference for Section 4.2						
maj.-vote SFT	AIME (on-set)	1.0	8,192	8 votes	50 / 51 / 50	26.0/27.5/30.0%

† final-round figures are filled from the released `audit.json` of each run, except the two policy-gradient seeds that were stopped after collapsing, whose per-round figures come from their logs (Table 5). The vote is 70–81% correct on the hard training stream against 26–30% when taken on AIME itself: the unmatched baseline differs from the ladder in supervision quality as well as in volume.

a systematically strict grader lowers accuracy at every checkpoint and largely cancels in the difference, whereas a lottery moves problems between states for no reason and is counted as learning or corruption. On one pilot run the fallback supplied 78% of all extracted answers.

Truncation interacts with this directly, which is F3. A completion that hits the token cap has no final answer and is graded incorrect, harmless if the rate is stable across checkpoints and severe if it is not (Table 4). We also track which extraction path produced each answer, since a method that stops emitting `\boxed{}` would otherwise register as capability loss.

Table 4: Truncation rate by checkpoint on the difficulty band, the diagnostic that excludes an arm's corruption counts (F3). A method that changes generation style will be scored as destroying capability under a fixed token budget, and nothing else in the ledger distinguishes the two.

arm	t=0	t=1	t=2	t=3	verdict
STaR	16.5%	15.9%	12.4%	14.7%	stable; counts stand
maj.-vote SFT	16.6%	14.6%	16.9%	16.0%	stable; counts stand
Distillation	16.5%	**65.3%**	**46.1%**	**49.6%**	drift; corruption counts excluded

A.3 Policy gradient on a majority-vote reward

Why the arm exists. Both self-training arms of the main audit are rejection-sampling SFT: they keep what their filter accepts and discard the rest, where TTRL as published pushes the losing samples down instead. An audit that positions itself against a reinforcement-learning literature and then measures only supervised fine-tuning is auditing something else, so we ran the genuine article: a group-relative policy gradient on the same stream and the same majority vote, under the training service's importance-sampling loss with group-centered advantages.

On the band it behaves; on AIME it does not survive. The band subsample gives 19 learned against 9 corrupted (CLR = 0.47) with accuracy rising 0.530 → 0.583 and truncation flat at 15.5, 24.4, 7.6 and 8.2% across its four checkpoints. The AIME arm, on a harder stream at a four times larger token cap, collapses. Table 5 gives all three of its seeds.

Table 5: **Three seeds of the policy-gradient arm, on identical data.** The evolution stream is drawn under a fixed seed, so the three runs see the same 256 problems in the same order each round; `--seed` varies adapter initialization, training order and sampling. Round 1 is the same experiment three times over: the vote is 77.0–77.9% correct and 135–136 groups of ∼250 carry any gradient. The runs then diverge completely. Every seed ends in severe truncation, so the collapse belongs to the configuration; the round at which it arrives does not. *Trained* is the number of problems that formed a majority, of 256; *spread* the groups carrying gradient; *vote* the fraction of those majorities that were in fact correct, a diagnostic that never enters training.

		evaluation, AIME		training stream			
seed	**round**	**accuracy**	**truncated**	**trained**	**spread**	**share**	**vote**
0	$t=0$	0.1764	4.2%	—	—	—	—
	$t=1$	0.2082	18.3%	248	135	0.78	77.0%
	$t=2$	0.0643	24.3%	251	143	0.77	73.7%
	$t=3$	0.1921	**76.4%**	243	217	0.58	49.8%
1	$t=0$	0.1760	4.4%	—	—	—	—
	$t=1$	0.0000	**100.0%**	249	135	0.78	77.5%
	$t=2$	—	—	34	32	0.51	85.3%
2	$t=0$	0.1829	4.5%	—	—	—	—
	$t=1$	0.2952	47.1%	253	136	0.78	77.9%
	$t=2$	0.0900	**95.6%**	248	105	0.86	85.1%

Seeds 1 and 2 were stopped after the collapse rather than sampled to t=3, since every further generation runs to the token cap. Seed 1's t=2 vote statistics are computed over the 34 majorities that survived and are a survivorship artifact; seed 2's t=2 are computed over 248 and are not. Training loss at t=1 was 169.9, 8.5 and 145.1 across the three seeds, a twentyfold spread from adapter initialization alone.

What three seeds show that one could not. The evolution stream is drawn under a fixed seed, so the three runs see the same 256 problems in the same order every round; `--seed` varies adapter initialization, training order and sampling. Round one is therefore the same experiment three times, and it comes out the same three times: the vote is 77.0, 77.5 and 77.9% correct and 135, 135 and 136 groups of roughly 250 carry any gradient. The runs then diverge to the point of one dying outright. Seed 1 truncates 100% of its completions after a single update and scores 0.0000; seed 2 reaches 95.6% after two; seed 0 reaches 76.4% after three. Two readings follow, and they point in opposite directions, which is why we give both. That every seed ends in severe truncation makes the collapse a property of this configuration rather than of one unlucky run; the answer to whether F3's drift was reproducible is yes. That the round at which it arrives varies from one update to three, on identical data, makes the arm's outcome undetermined by its design, which is F5 in a more extreme form than the supervised arms show. The training loss at round one was 169.9, 8.5 and 145.1 across the three seeds: a twentyfold spread from adapter initialization alone.

The diagnostic reads healthy while the model dies. A group contributes gradient exactly when its rollouts disagree, so the arm's supervision statistics are a live readout of how self-consistent the model is. On seed 0 they degrade as the model does. The vote falls 77.0 → 73.7 → 49.8% and the share of rollouts agreeing with it 0.78 → 0.58, while the groups carrying gradient rise 135 → 143 → 217: by round three the pseudo-label is close to a coin flip and the optimizer receives more gradient than ever, precisely because the model has stopped agreeing with itself. That is a real measurement (243 of 256 problems still formed a majority), but it is one seed at one round, and it does not replicate. On seed 2 the same statistics move the other way at round two, over an equally intact denominator of 248: the vote becomes *more* accurate (77.9 → 85.1%) and *more* unanimous (share 0.78 → 0.86), and fewer groups carry gradient (136 → 105), while the evaluation goes from 47% truncated to 96%. Seed 1's round-two figures (85.3% correct) are a survivorship artifact and we mark them as such: only 34 of 256 problems formed a majority at all, because a degenerate generation yields no extractable answer to vote on. What survives across the seeds is not a direction but a warning: on two of three runs the arm's own supervision statistics looked stable or improving at the checkpoint after which the model was unusable, so they cannot be used to decide when to stop.

The scope of the claim, and the exclusion criterion. The AIME arm's transition counts are excluded from the capability analysis. They are read off models truncating 76 to 100% of their completions, and F3 establishes that a token cap under generation-length drift scores style change as capability loss; the criterion applies to this arm exactly as it does to distillation. The detection counts of Table 2 are retained because truncation moves them the other way: a truncated completion cannot be scored correct, so a collapsing arm *understates* its detections and the count is a floor. We claim, for this configuration only, that three rounds of policy gradient on a majority-vote reward at rank-32 LoRA and $\eta = 10^{-4}$ collapsed the policy into unbounded repetition on every seed we ran. We do not claim that a majority-vote reward does this in general: seed 2 took two intact updates at the identical learning rate, so the rate is not uniformly fatal, and the band arm at a shorter cap on an easier stream never collapses at all. Separating the reward from the step size needs a learning-rate sweep we did not run.

A.4 Compute footprint and cost

All experiments used Tinker, a hosted LoRA training and sampling service. Unless stated otherwise the backbone is Qwen3-8B with rank-32 adapters; the distillation teacher is `gpt-oss-120b` (OpenAI et al., 2025), and one noise floor was measured on a second 20B backbone. The study consumed 3.44M sampled generations and 5.30B generated tokens across 48 runs, itemized in Table 6.

Table 6: Compute by experiment class. Cost is the metered price of sampling and training on the hosted service; token counts are generated tokens, which dominate cost.

experiment class	runs	samples (M)	tokens (M)	cost (USD)
Primary audits, difficulty band (n=1,163)	3	1.80	2,441	1,465
Primary audits, MATH-500 and AIME	4	0.27	414	248
Matched AIME ladder, 3 filters $\times$ seeds	6	0.19	692	415
Policy-gradient arms	4	0.16	443	266
Seed replication (2 methods $\times$ 2 seeds)	4	0.36	470	282
Noise floors (4 benchmarks)	7	0.42	589	340
Benchmark construction and model profiling	7	0.11	101	61
Out-of-domain probes (3 replicates each)	6	0.01	4	2
Batch-invariance test (F1 mechanism)	1	< 0.01	1	< 1
Pilot audits, 112-problem band	2	0.12	147	88
Method pilots and calibration	4	< 0.01	4	2
Total	**48**	**3.44**	**5,303**	**3,171**

Three ratios bear on the paper's recommendation. The measurement infrastructure (noise floors, benchmark construction and the out-of-domain probes) accounts for 13% of total spend, and the small pilots that exposed four of the seven failure modes cost \$2 between them. Against that, the three primary audits alone cost \$1,465. And the controls the paper's conclusions rest on cost *nothing*: the empirical null distributions, the design-matched floors, the threshold-free expansion test and the seed analysis are all re-computations over records the study already holds, because every arm's baseline is another evaluation of the untrained model. What money buys is the matched ladder (\$415), which removes a confound rather than measuring something new, and the policy-gradient arms (\$266), which audit a method family the SFT arms cannot stand in for and surface a failure mode none of the others can exhibit.

The dominant cost is sampling rather than training: evaluating 1,163 problems at k=128 over four checkpoints is 600K generations per arm, against roughly 270 optimizer steps. This is a property of transition-level auditing generally: statistical power comes from problems near the decision boundary, and reaching them requires sampling breadth rather than longer training. An audit budget is therefore set almost entirely by the evaluation design, and can be estimated in advance from the benchmark size, k, the checkpoint count and the mean completion length, all known before any method is run.

B BENCHMARK SUITES AND DATASET CONSTRUCTION

Corruption and expansion are observable on disjoint populations of problems, so no single evaluation set can host both. This appendix gives the construction of the set built to make corruption observable (§B.1) and the characterization of all three sets against the regimes they can and cannot measure (§B.2).

B.1 CONSTRUCTING THE DIFFICULTY BAND

Corruption is observable only on problems a model already solves, and expansion only on problems it does not reach. MATH-500 supplies the first population and almost none of the second; AIME the reverse (Figure 4). Neither can host a well-powered corruption measurement. We therefore construct a third set: profile $10{,}999$ held-out MATH training problems at $k{=}8$ and keep the $1{,}611$ whose measured solve rate falls in $[0.25, 0.75]$, and retain the $1{,}163$ of them closest to the center of that interval. The retained set has a mean solve rate of 0.522 under the $k{=}8$ selection profile and 0.552 when re-measured at $k{=}128$, where it holds 711 corruptible and 452 learnable problems, the first balanced pools in the study.

Selecting problems by measured solve rate and then measuring them again invites an obvious objection: regression to the mean. Figure 3 shows it plainly. This does not bias the transition counts, for a specific reason: selection uses an independent $k{=}8$ profile, the audit re-evaluates every problem from scratch at $k{=}128$, and *that* measurement is checkpoint 0, the baseline against which all transitions are computed. The noisy estimate used to choose never enters the ledger. What regression to the mean costs is realized band width: fewer problems sit near threshold at $k{=}128$ than the selection targeted, which reduces power but cannot manufacture a transition.

The band's construction removes every problem the base model cannot reach, so it has no expansion candidates and its expansion statistic is vacuous. This is the price of its corruption sensitivity. It also means the band's raw counts are not a forgetting rate for a naturally sampled problem distribution: the set is deliberately enriched for problems that can transition, and Table 23 rather than the raw count is the quantity that transfers.

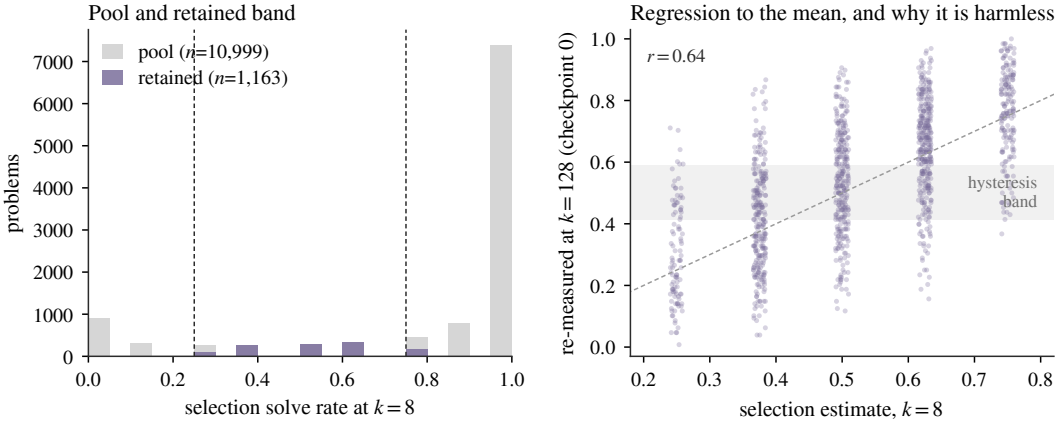


Figure 3: **Construction of the difficulty band.** *Left:* the held-out pool and the retained band; dashed lines mark the selection interval $[0.25, 0.75]$. *Right:* selection estimate at $k{=}8$ against the independent re-measurement at $k{=}128$ that serves as checkpoint 0. Regression to the mean widens the realized band, costing statistical power; because the two measurements are independent it cannot create transitions.

B.2 BENCHMARK CHARACTERIZATION AND SOLVABILITY REGIMES

Benchmarks, in full. Corruption is observable only on problems the model already solves, expansion only on problems it does not reach. These populations are disjoint, and a benchmark rich in one is poor in the other (Figure 4), so we use three sets. **MATH-500** (Lightman et al., 2024), subsampled to 200 problems by a fixed stratified draw over (subject, level) released with the code, gives

comparability with prior work but is thoroughly contaminated (168–170 of 200 problems already solved at k=128), hosting corruption but not expansion. **AIME 2025 and 2026** (60 problems) are the least contaminated sets available to us: AIME 2026 post-dates the backbone's March 2025 cut-off (Yang et al., 2025) outright, while AIME 2025 was administered in February 2025 and sits just inside it, so we report the 2026 half separately wherever the expansion claim rests on it. The base model's mean solve rate on AIME is 0.175–0.180 across its eleven baselines, leaving 22 problems it reaches at most five times in 1,408 pooled draws and only 8 that can be corrupted, so AIME carries the expansion claims and has no corruption power. A **difficulty band** of 1,163 problems, built by profiling 10,999 held-out MATH problems (Hendrycks et al., 2021) at k=8 and retaining those with solve rate in $[0.25, 0.75]$, is deliberately enriched for problems near the decision boundary so that corruption is observable at all. Its counts are therefore not a forgetting rate for a natural problem distribution, and by construction it contains no base-unreached problems (Appendix B.1). Because the fraction of problems near threshold differs across sets, CLR is *not* portable between them.

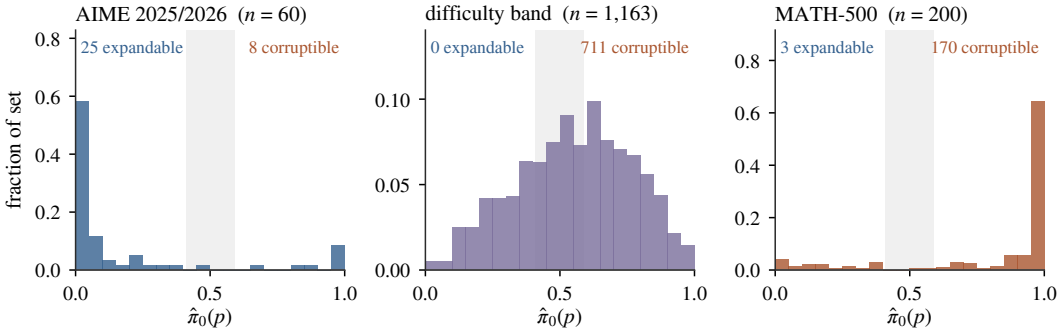


Figure 4: **An evaluation set fixes which question it can answer, before any method is applied.** Measured distribution of the untrained backbone's per-problem solve rate at k=128, from the frozen-control records; shading marks the hysteresis band. Expansion is observable only in the spike at $\hat{\pi}_0 = 0$ and corruption only in the mass above 0.5, so AIME can host an expansion measurement and not a corruption one, MATH-500 the reverse, and the constructed band neither expansion nor a natural forgetting rate.

A falsified prediction about where corruption appears. We pre-registered the prediction that majority-vote self-training would corrupt more where its pseudo-labels were less reliable, and it was falsified. On AIME its label accuracy falls to 26–30%, so the vote is wrong most of the time (Table 3), and corruption goes to *zero*. The explanation is that AIME leaves only 8 problems the model solves and could therefore lose; the smallest non-zero corruption rate observable there is 12.5%, against a rate of 1.2% measured on MATH-500.

Wrong supervision on problems a model already fails destroys nothing. Corruption is bounded by what there is to lose, not by how wrong the teacher is, and an evaluation set can be too hard to detect it as easily as it can be too easy. We report the falsification because the account that replaces it, that corruption requires both unreliable supervision *and* prior competence on the affected problems, is what motivates constructing the difficulty band, and because it is the reason the AIME arms carry expansion claims but not corruption ones.

C THE TRANSITION LEDGER AND ITS NOISE FLOOR

The ledger of Section 3 turns per-problem solve rates into transition counts. This appendix gives the rule in full and the counts it produces (§C.1), the floors and the sensitivity of both to z and k (§C.2), and the three failure modes that attack the ledger itself (§C.3).

C.1 THE BASELINE STATE, HYSTERESIS, AND THE FULL LEDGER COUNTS

The baseline state carries no hysteresis, and this matters. Equation 2 is undefined at $t = 0$, where there is no previous state to hold; we assign $s_0(p) = \textsf{solved}$ iff $\hat{\pi}_0(p) \geq \frac{1}{2}$. The consequence is

Table 7: **What each method adds and removes, against a floor measured on the same problems and matched to the same number of checkpoints.** Counts are problems. The floor columns give the median over four-checkpoint orderings of the independent evaluations of the untrained model that benchmark carries, with the conventional two-checkpoint floor in parentheses.

		transitions		matched floor		
method	**benchmark**	**learned**	**corrupted**	**L**	**C**	**accuracy**
STaR	MATH-500	2	1	0 (0)	1 (1)	$0.835 \to 0.850$
maj.-vote SFT	MATH-500	3	2	0 (0)	1 (1)	$0.832 \to 0.830$
maj.-vote SFT	AIME	4	0	0 (0)	0 (0)	$0.177 \to 0.202$
Distillation†	AIME	9	0	0 (0)	0 (0)	$0.180 \to 0.310$
STaR	band	149	106	10 (4)	8 (5)	$0.550 \to 0.606$
maj.-vote SFT	band	145	88	10 (4)	8 (5)	$0.550 \to 0.590$
Distillation†‡	band	161	*177*	10 (4)	8 (5)	$0.550 \to 0.546$

†positive control. ‡corruption counts excluded: truncation drifts $16.5\% \to 65.3\%$ (F3). The AIME rows here are the unmatched baselines (on-set training, unmatched volume); the matched ladder is Table 2. Sharpened share of newly solved problems on the sets where expansion is observable at all: $2/2$, $3/3$, $4/4$, so $9/9$, Wilson $[0.70, 1.00]$. The band arms ($149/149$, $145/145$, $161/161$) are vacuous by construction and are excluded from that proportion rather than added to it. CLR with percentile-bootstrap intervals over problems, band: STaR 0.711 $[0.557, 0.915]$, maj.-vote SFT 0.607 $[0.460, 0.785]$; seed variance is roughly three times this (Table 13).

load-bearing and easy to miss. A problem whose baseline sits inside the band is unprotected at that one checkpoint, so a frozen model can register a transition on a movement smaller than the band is wide. That is why the floors of Table 10 are not zero on a set whose mean solve rate is 0.552. Section 5 reports one alternative, a pure two-checkpoint rule with no carry. The other is symmetric: treat a baseline inside the band as *indeterminate* and let a problem transition only once it has left the band. That rule drives the frozen floor to exactly 0 learned and 0 corrupted, with no artifact surviving it, while STaR gives $64/53$ and majority-vote SFT $75/55$. It buys that zero by setting aside 325 of the $1{,}163$ problems, which is why we do not adopt it as the primary rule: a floor of zero over a set chosen for having no determinate starting state is a weaker achievement than a small floor over the whole set. The conclusion is unchanged under it, the arms now clearing the floor absolutely rather than by a ratio.

Table 9 gives the complete seven-category partition for every arm. The categories are mutually exclusive and sum to the problem count. *Recovered* and *transient* problems, those that change state and change back, are reported separately because they are invisible to an endpoint comparison but bear on whether a method's damage is permanent. Their scarcity (at most 5% of problems in any arm) is why we report the endpoint definition of CLR in the main text: on this data the two definitions coincide to within three problems. Movement is bidirectional in every arm, the band producing 88–175 corruptions alongside 144–160 learnings, so a rising mean accuracy is a net result and not a uniform lift.

The path to the endpoint. Table 8 and Figure 5 give the same ledger evaluated at every checkpoint rather than only at the last. Two features are invisible in the endpoint comparison. Distillation on the band peaks at round 1 (0.610) and then falls to below its own baseline by round 3 (0.546), which is the truncation drift of F3 taking hold as the student's completions lengthen. Majority-vote self-training peaks at round 2 (0.605) and gives back 1.5 points at round 3 while its corruption count rises from 59 to 88; a study that stopped at two rounds would have reported a better method. Neither observation changes a headline, and we report the endpoint comparison because it is the one prior work reports, but the trajectory is where a practitioner would choose when to stop.

Where the mass goes. Figure 6 bins the solve rate into five bands and tabulates the joint distribution of before and after. The diagonal holds; the off-diagonal mass is large in both directions for every arm: 464 problems up and 244 down for STaR, 435 and 276 for majority-vote self-training, 402 and 389 for distillation. The bottom-left cell is the one that matters for the expansion claim, and

it is where the band's construction bites: with no problems at $\hat{\pi}_0 = 0$ there is nothing in that cell to move. Appendix D.2 shows the same movement without binning.

Table 8: **Per-checkpoint accuracy and transition counts.** Every headline number in the paper compares t=0 with t=3; this is the path between them. Each count compares that checkpoint with checkpoint 0 under the ledger of Eq. 2; it is a running state, not a cumulative total, so it can fall when a problem re-crosses the band, and the final column of each block reproduces Table 7. Distillation on the band peaks at round 1 and declines under its own truncation drift (F3); majority-vote SFT on the band peaks at round 2 in accuracy while its corruption count keeps rising. The AIME policy-gradient row is reported for completeness only: that arm collapses into unbounded repetition and is 76.4% truncated at t=3, so its transition counts are excluded from the capability analysis (Appendix A.3).

		mean accuracy				learned vs t_0				corrupted vs t_0			
arm	set	t_0	t_1	t_2	t_3	t_0	t_1	t_2	t_3	t_0	t_1	t_2	t_3
STaR	band	0.550	0.590	0.591	0.606	0	83	123	149	0	48	98	106
maj.-vote SFT	band	0.550	0.586	0.605	0.590	0	57	119	145	0	33	59	88
Distillation†	band	0.550	0.610	0.577	0.546	0	151	168	161	0	101	143	177
maj.-vote RL	band (175)	0.530	0.603	0.578	0.583	0	22	21	19	0	13	12	9
STaR	MATH-500	0.835	0.841	0.851	0.850	0	1	4	2	0	1	1	1
maj.-vote SFT	MATH-500	0.832	0.836	0.830	0.830	0	1	2	3	0	0	0	2
maj.-vote SFT	AIME	0.180	0.193	0.227	0.213	0	1	1	1	0	0	0	0
STaR	AIME	0.177	0.190	0.204	0.195	0	0	0	1	0	0	0	0
Distillation†	AIME	0.180	0.347	0.308	0.310	0	12	11	9	0	0	0	0
maj.-vote RL‡	AIME	0.176	0.208	0.064	0.192	0	3	0	4	0	0	6	2

†positive control, not a self-evolution method. ‡counts excluded: generation collapse (Table 5).

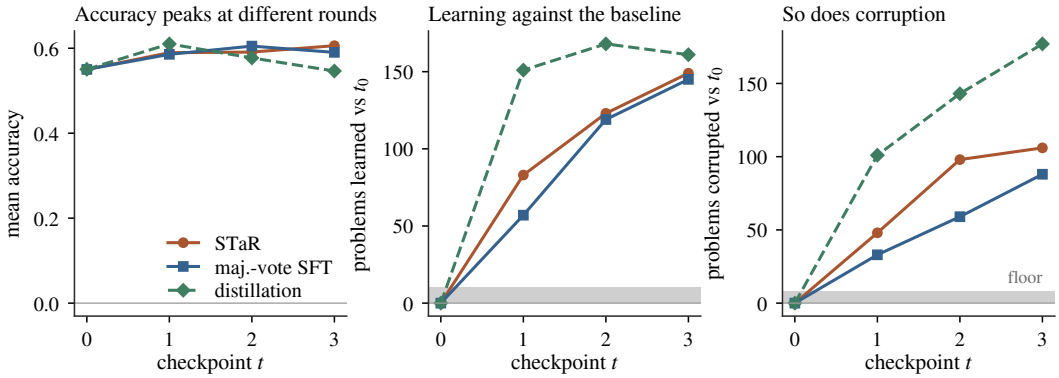


Figure 5: **The path the endpoint comparison hides.** Mean accuracy and the learned and corrupted counts against checkpoint 0 at every checkpoint on the difficulty band, with the frozen-model floor shaded on the two count panels. Distillation peaks at round 1 and declines under its own truncation drift; majority-vote self-training's accuracy peaks at round 2 while its corruption count continues to climb.

C.2 Noise floors, and sensitivity to z and k

Table 11 names every MATH-500 problem whose greedy verdict changed between two evaluations of the frozen model, with the solve rate measured on the same samples. The point is the disagreement between the two: `num_theory/598` is scored as corrupted while its solve rate is 0.898 at both checkpoints, and `int_algebra/1063` is scored as *learned* while its solve rate *falls* from 0.133 to 0.023. The single-decode verdict is not a noisy measurement of capability; on these problems it is close to independent of it. Nine of the fifteen are scored as corruptions and six as learnings, which is the CLR $= 1.5$ of Section 4.1 in full.

Table 12 sweeps the hysteresis width z and the sampling budget k on the difficulty band. The arms' ratios are flat across both sweeps while the frozen-model floor is not, which is what the hysteresis

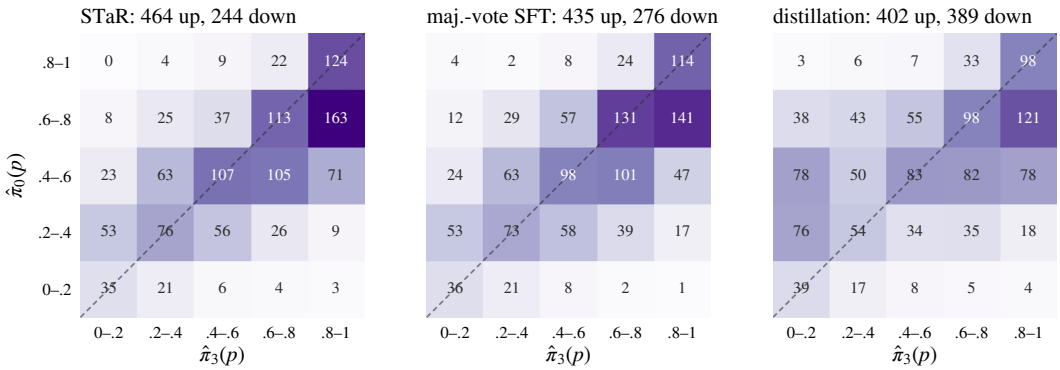


Figure 6: **Migration matrices: where problems actually move.** Joint distribution of the per-problem solve rate before and after evolution, binned into fifths, for the three difficulty-band arms; the dashed line is no change and cell counts sum to 1,163. "Up" and "down" count problems above and below the diagonal. Movement is substantial in both directions for every arm, which is the point that a mean accuracy cannot express.

Table 9: Ledger categories by arm, sampled ledger at k=128, z=2. *Learned* and *corrupted* here are the trajectory-based labels; the endpoint counts used for CLR in the main text differ only where a problem oscillates.

		stable		changed				
arm	**benchmark**	**correct**	**incorrect**	**learned**	**corrupted**	**recovered**	**transient**	**oscillating**
STaR	MATH-500	167	25	2	1	2	3	0
maj.-vote SFT	MATH-500	167	28	3	2	0	0	0
maj.-vote SFT	AIME	8	48	4	0	0	0	0
Distillation†	AIME	8	40	9	0	0	3	0
STaR	band	567	299	149	103	25	17	3
maj.-vote SFT	band	589	299	144	88	19	23	1
Distillation†‡	band	532	248	160	175	11	34	3

†positive control. ‡corruption counts excluded: truncation drift (F3).

band is for. Smaller k is simulated by subsampling the stored 128 draws without replacement, so the comparison is exact rather than a fresh sample.

What the measured floor buys, and what the rule does. A binomial calculation at each problem's pooled base rate, the analytic unchanged-policy null of Yuan et al. (2026), predicts a median of 3 learned and 4 corrupted on the band against our measured 4 and 3, and an envelope of 0.203 against 0.234. For the solve-rate estimator the analytic null is very nearly right, and we say so rather than claim an advantage we cannot demonstrate; where the two part company is the single-decode protocol, for which an analytic null predicts no transitions at all. Discretization choices likewise move the counts without moving the conclusion: under a pure two-checkpoint rule with no carry, STaR's counts fall from $149/106$ to $120/79$ and majority-vote SFT's from $145/88$ to $118/78$ while the floor stays at $4/5$; dropping the band entirely raises the arms to $172/111$ and $167/117$ but the floor to $44/39$. CLR stays within 0.65–0.71 for STaR across all three rules, and within 0.65–0.72 over $z \in [0, 3]$ and 0.56–0.81 over $k \in \{16, 32, 64, 128\}$ (Table 12). One caution bears on the main claim: under the pure endpoint rule the two methods coincide exactly (0.66 and 0.66), and at $z = 0$ their order reverses, so the between-method comparison is sensitive to the rule as well as to the seed and we rest nothing on it.

C.3 Failure modes F3, F4 and F5 in full

F3. A token cap scores style change as capability loss. The distillation arm on the band records 177 corruptions and CLR $= 1.10$, nominally the most destructive method tested. Its truncation rate meanwhile moves from 16.5% to 65.3% between checkpoints while both self-training arms stay

Table 10: **Noise floors: apparent transitions produced by an untrained model** (F1, F7). Estimator columns give the range over every ordered pair of independent frozen evaluations (110 pairs on AIME, 12 on MATH-500 and 20 on the band) and, in the last estimator column, the median over four-checkpoint orderings that are design-matched to an arm. A conventional single-pair floor is the low end of these ranges. The single-greedy protocol is what a per-example before/after comparison implicitly uses; its changed rate carries a Wilson interval since it is a proportion over problems.

		solve-rate estimator, k=128			single greedy sample		
benchmark	**backbone**	**learned**	**corrupted**	**matched (L/C)**	**L/C**	**changed [95%]**	**problems**
MATH-500	Qwen3-8B	0–1	0–1	0 / 1	6 / 9	7.5% [4.6, 12.0]	200
AIME 2025/26	Qwen3-8B	0–0	0–1	0 / 0	2 / 2	6.7% [2.6, 15.9]	60
difficulty band	Qwen3-8B	0–7	1–9	**10 / 8**	100 / 120	**18.9%** [16.8, 21.3]	1,163
band (pilot)	Qwen3-8B	0	0	—	—	—	112
AIME 2025/26	gpt-oss-20b	0	0	—	5 / 4	15.0% [8.1, 26.1]	60

The gpt-oss-20b row is the only measurement we report on a second backbone; its arms were not run because its capability profile places it in a different regime, leaving only 7 base-unreached AIME problems. Pilot and second-backbone rows have a single frozen pair each and so have no range.

Table 11: **Every problem whose greedy verdict changed between two evaluations of the frozen model** (F1), MATH-500, T=0 with a fixed seed. ✓ and × are the greedy verdicts; $\hat{\pi}$ is the solve rate over 128 samples at the same checkpoint. The point of the table is the last two columns of each block: the underlying solve rate barely moves, since it is a frozen model, while the single-decode verdict flips. Nine of these are scored as corruptions and six as learnings, giving the CLR $= 1.5$ of Section 4.1.

problem	t_0	t_1	$\hat{\pi}_0$	$\hat{\pi}_1$	problem	t_0	t_1	$\hat{\pi}_0$	$\hat{\pi}_1$
counting_prob/199	×	✓	0.633	0.672	num_theory/183	×	✓	0.875	0.852
geometry/226	✓	×	0.719	0.750	num_theory/357	✓	×	0.594	0.453
geometry/702	×	✓	0.078	0.117	num_theory/516	✓	×	0.398	0.328
int_algebra/1063	×	✓	0.133	0.023	num_theory/598	✓	×	0.898	0.898
int_algebra/1350	✓	×	0.367	0.367	prealgebra/1128	✓	×	0.688	0.758
int_algebra/1797	×	✓	0.859	0.844	precalculus/1313	✓	×	0.602	0.695
int_algebra/754	×	✓	0.547	0.367	precalculus/44	✓	×	0.922	0.953
int_algebra/966	✓	×	0.391	0.344					

Under the solve-rate estimator of Eq. 2, the same 200 problems yield 0 learnings and 1 corruption.

flat at 12–17%: the student has adopted its 120B teacher's verbose style and is hitting the token cap, losing the final answer. Conditioned on completions that finish, it improves from 0.656 to 0.845 against STaR's 0.655 to 0.702: the most effective method tested, not the most destructive. Its corruption counts are excluded from the capability analysis rather than caveated, since severe generation-length drift under a fixed token cap confounds capability degradation with stylistic verbosity (Appendix A.2).

F4. Transition metrics need an order of magnitude more data than accuracy. Only problems near the decision boundary can produce an event, so a benchmark's difficulty profile determines power directly. Detecting the difference in corrupted share we observe between STaR and majority-vote self-training at 80% power requires roughly 249 events per arm; a 112-problem pilot yields 24 and cannot resolve it at any significance level, which is why the primary comparison uses 1,163 problems (Appendix D.3).

F5. The comparison between methods is not stable across training seeds. Table 13 gives all three seeds of both band arms with every seed scored on the same 175 problems, and Figure 8 plots the resulting spread against the noise floor. Seed 0 ran on the full band and seeds 1–2 on a 175-problem subsample, so the matched columns restrict seed 0 to those same 175 problems: the reversal is a seed effect and not a set effect. STaR holds CLR within 0.45–0.74 on every seed, while majority-vote self-training spans 0.55 to 1.53 and crosses the threshold at which a method destroys more than it creates. Section 4.3 states what that licenses and what it does not.

Table 12: Sensitivity of the ledger to z and k: learned/corrupted counts with CLR in parentheses, difficulty band.

hysteresis width z (at k=128)				**sampling budget k (at z=2)**			
z	**STaR**	**maj.-vote**	**frozen**	k	**STaR**	**maj.-vote**	**frozen**
0.0	172/111 (0.65)	167/117 (0.70)	44/39	16	115/64 (0.56)	125/69 (0.55)	10/7
1.0	165/111 (0.67)	158/103 (0.65)	12/22	32	103/83 (0.81)	120/88 (0.73)	6/8
1.5	160/109 (0.68)	151/97 (0.64)	7/12	64	134/95 (0.71)	143/79 (0.55)	5/7
2.0	**149/106 (0.71)**	**145/88 (0.61)**	**4/5**	128	**149/106 (0.71)**	**145/88 (0.61)**	**4/5**
2.5	136/98 (0.72)	131/85 (0.65)	1/3				
3.0	125/89 (0.71)	121/76 (0.63)	0/2				

Table 13: **Seed replication (F5), with every seed scored on the same problems.** Seed 0 ran on the full band and seeds 1–2 on a 175-problem subsample, so an as-run comparison confounds seed with evaluation set. Restricting seed 0 to the same 175 problems costs no additional sampling and is given in the matched columns: the reversal is a seed effect, not a set effect.

method	**seed**	n (matched)	**learned**	**corrupted**	CLR **matched**	CLR **as run**	
STaR	0	175	27	19	0.704	0.711	
	1	175	23	17	0.739	0.739	range 0.45–0.74; net positive
	2	175	22	10	0.455	0.455	
maj.-vote	0	175	20	11	0.550	0.607	
	1	175	19	29	**1.526**	1.526	range 0.55–1.53; 2/3 net destructive
	2	175	18	24	**1.333**	1.333	

seed 0, full band: corrupted share 0.416 vs 0.378, Fisher $p = 0.406$. Pooling seeds as a fixed effect is not defensible with three of them, so we report the spread rather than a pooled p-value. Δ accuracy by seed:
STaR $+0.056/+0.056/+0.069$; maj.-vote $+0.040/-0.028/-0.010$.

D EXPANSION STATISTICS, POWER AND OUT-OF-DOMAIN PROBES

The expansion half of the audit rests on a per-problem exact test against a pooled baseline. This appendix varies every choice that test leaves open (§D.1), itemizes the problems it is computed on (§D.2), gives the power behind its null result and the effect sizes behind the corruption counts (§D.3), and reports the two out-of-domain probes (§D.4).

D.1 ROBUSTNESS OF THE THRESHOLD-FREE EXACT TEST

The test of Section 4.2 replaces one explicit parameter with four implicit ones: the error rate, the one-sidedness, the number of baselines pooled and the base-rate strata. It also rests on an assumption, that the eleven baseline evaluations are exchangeable. This subsection varies each and checks the assumption. Nothing here required a new experiment.

Does the decision rule make the comparison? A false-discovery-rate procedure is adaptive: it admits a larger per-hypothesis cutoff in an arm containing many true effects, so in principle an arm with a large global uplift can purchase a more permissive standard for its own marginal problems, and a signal-free control can be held to a stricter one. The concern is well founded in general and does not apply here, for a reason that is checkable rather than arguable: the cutoff a procedure *admits* is $k\alpha/M$, but the cutoff it *attains* is the k-th smallest p-value, and here those are 0.0095 for distillation and 0.0094 for majority-vote self-training. The two arms are held to the same standard to three decimal places. Table 14 repeats the comparison under Bonferroni, at fixed thresholds, and under a single procedure applied jointly across every arm and problem at once; the low-base counts move by at most three and the ordering never changes. The one qualification is that the frozen control's zero is a property of a corrected rule: at an uncorrected $p \leq 0.05$ it returns 13 detections over 660 tests, which is the nominal rate and not a failure.

What the interaction shows. The arms of the ladder are matched in stream, retained volume and evaluation, but not in how much they improve: the teacher moves accuracy $+13$ to $+16$ points and majority-vote self-training -1.5 to $+3.3$. A model in which every method lifts every problem

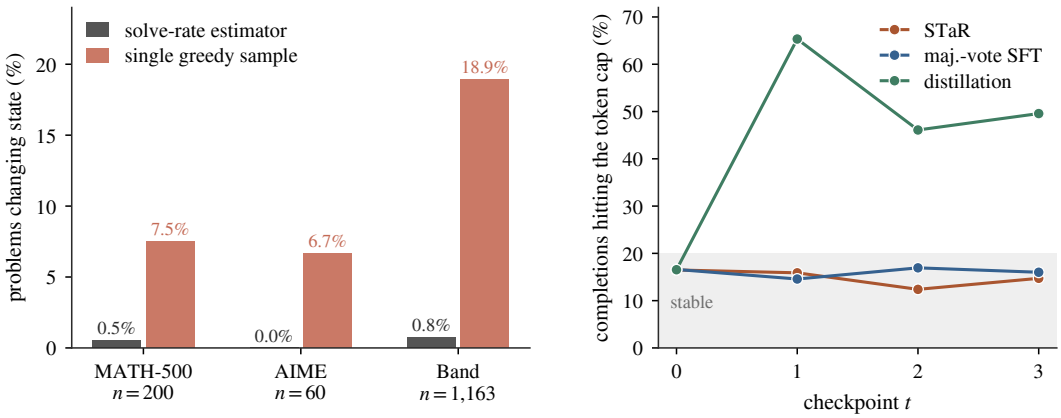


Figure 7: **Two of the seven failures, on the runs where they occurred.** *Left:* apparent transitions produced by a model that never trained, under two definitions of correctness (F1). A single greedy sample manufactures state changes on 6.7–18.9% of problems; the solve-rate estimator with a hysteresis band reports 0.0–0.8%. *Right:* truncation rate per checkpoint (F3). The distillation student adopted its teacher's verbose style and began hitting the token cap, losing final answers and registering as capability loss; the self-evolution arms are flat, which is why their counts survive.

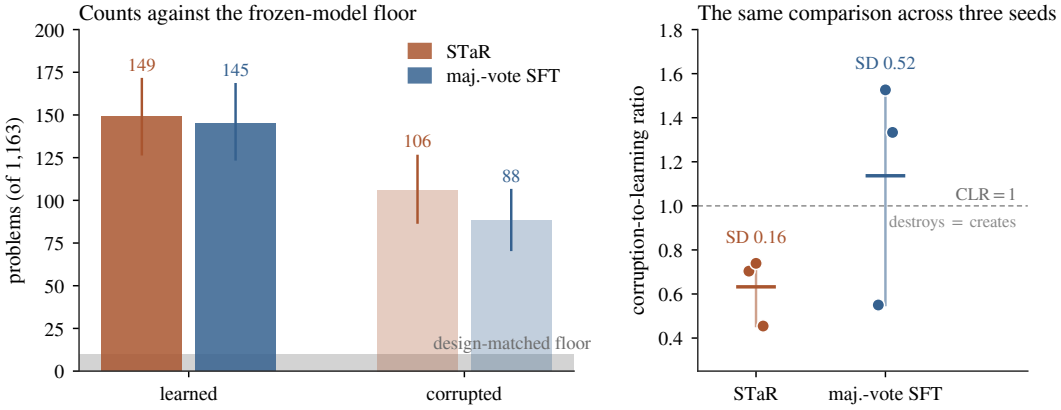


Figure 8: **Transitions clear the noise floor by $\sim 11\times$, but the comparison between methods is not stable across seeds.** *Left:* learned and corrupted counts on the 1,163-problem difficulty band with bootstrap intervals over problems; the shaded region is the frozen-model floor, design-matched to the arms' four checkpoints rather than the conventional two-checkpoint control, which understates it by roughly half. *Right:* CLR across three training seeds, every seed scored on the same 175 problems (F5). STaR holds 0.45–0.74; majority-vote self-training spans 0.55 to 1.53, crossing the threshold at which a method destroys more than it creates.

by a common factor in log-odds would reproduce more detections for the teacher everywhere, and disproportionately more where the base rate is low, without any qualitative difference in kind. Table 15 tests that model and rejects it: the teacher moves low-base problems by a further 1.91 in log-odds beyond its own overall lift, where both self-training arms are indistinguishable from zero. The dissociation is a difference in what the methods reach, not only in how far they move.

An analytic null, and what measuring buys. Appendix C.2 reports that for the solve-rate ledger an analytic unchanged-policy null very nearly reproduces our measured floor, which invites the question of whether the expansion null could have been calculated rather than measured. It can be, and Table 17 does so. The calculation is optimistic at every threshold and increasingly so as m rises, because the base-unreached set is selected on a noisy zero. A problem observed at $0/1{,}408$ has a plug-in success probability of exactly zero and is therefore assigned no chance of a later success,

when in truth it has a small one. At $m{=}2$ the calculation returns 0.044 against a measured 0.058. An author who derived the null instead of measuring it would certify the threshold that F7 rules out.

How many replicates a null needs. Our recommendation, that a multi-arm study already owns several frozen replicates, is only useful with a number attached to it. Table 18 re-estimates the ER_2 null from every subset of our eleven baselines, exhaustively, since there are never more than 462 of them. At four replicates, which is what a three-arm study with a no-op control possesses, the estimate still ranges from 0.022 to 0.098: such a study could certify a null of 0.022 and reproduce F7 at a slightly larger sample size. The spread falls to 0.043 at seven replicates and 0.024 at nine, and it is nine before the estimate holds within ± 0.02 of the eleven-replicate value. The recommendation should therefore be read as *report the interval and the replicate count it came from*, and a null one intends to certify against needs rather more than a three-arm study owns.

Are the eleven baselines exchangeable? They were run at different times against a hosted service whose batching we ourselves show to be non-deterministic (F1), so pooling them assumes something that has to be checked: that no run-level effect inflates the arm-versus-pool comparison while leaving the held-out null clean. Fitting the eleven per-problem counts against their pooled rate gives a Pearson dispersion of 0.966 on 500 degrees of freedom (a value of 1 is exactly binomial), with a median per-problem ratio of 0.907. The implied intraclass correlation is indistinguishable from zero and the design effect at $k{=}128$ is 1.000. The draws are exchangeable across evaluations, which is what the exact test assumes and what the pooling requires.

Zero is an estimate too. The held-out null returns no detection on any of the eleven replicates, which is a count and not a rate. At the replicate level the Clopper–Pearson bound is 0.24; over the 660 individual tests it is 0.0045. We report both rather than repeat, one level up, the failure F7 describes.

The small non-zero counts, and the teacher's handicap. We state the small non-zero counts rather than round them away. They are what an unmatched comparison hides: applied to a stream of problems it can actually solve, self-training does occasionally raise a rarely-reached problem above the noise. It does so at a fifth of the teacher's rate; the direction is unchanged, and the quantity that transfers is a rate rather than a categorical "never". The comparison is also conservative in the teacher's disfavor for a reason F3 predicts: the distillation arms truncate 36–46% of completions even at an $8{,}192$-token cap, rising 4.2% to 53.4% across checkpoints on seed 0, against 4–15% for the self-training arms. F3 excludes an arm's *corruption* counts because truncation inflates them; here it works the other way, since a truncated completion cannot be scored correct and so cannot produce a detection. The teacher's counts are a floor on what it did, and the dissociation is understated rather than manufactured. Sensitivity is arithmetic rather than something to be bought: against a base of $0/1{,}408$ an endpoint of $2/128$ already clears raw alpha, which is also the smallest count that survived FDR in the distillation arm.

D.2 Every base-unreached AIME problem, itemized

Table 21 lists all 29 AIME problems that at least one arm found base-unreached at $k{=}128$, with the correct-sample count at that arm's own baseline and at its final checkpoint. It is the whole of the expansion evidence in the paper, at the level of individual problems, and we give it in full because the counts are small enough that a reader should be able to check the conclusion rather than take it.

Three things can be read off it directly. The frozen control's seven apparent expansions are *all* exactly one correct sample in 128; none of them is two. The distillation arm's expansions are not: its eleven expansions at $m \geq 2$ run $2, 3, 3, 6, 7, 8, 9, 9, 18, 33, 49$, and the two largest (`2025/21` at 49 and `2025/26` at 33) are the same problems the frozen control "expanded" with a single lucky draw. And the majority-vote arm's three at $m{=}1$ reduce to one at $m{=}2$, `2025/26` at two samples, which is why we report $1/21$ rather than $3/21$.

The three baseline columns differ (25, 21 and 22 problems at zero) because each arm's checkpoint 0 is its own independent $k{=}128$ evaluation of the same untrained model. Thirteen of the 29 rows are at zero in one arm's baseline and at one, two or three samples in another's. That disagreement

Table 14: **The comparison does not depend on the decision rule.** Low-base detections (pooled base rate $\leq 5/1{,}408$, $n = 22$) under four rules applied identically to every arm, and under one joint procedure over all arms and problems at once. A false-discovery-rate procedure is adaptive, so an arm with many true effects admits a larger cutoff; the cutoffs actually attained here are 0.0095 for distillation's 36 detections and 0.0094 for majority-vote self-training's 19, so the two arms are held to the same standard. The frozen row is the leave-one-out null over all eleven replicates.

arm	**BH, per arm**	**Bonferroni**	$p \leq 0.01$	$p \leq 0.05$	**joint BH**
distillation, seed 0	10	7	10	10	9
distillation, seed 1	11	6	10	11	—
distillation, seed 2	8	7	7	8	—
maj.-vote SFT, seed 0	2	0	2	2	0
maj.-vote SFT, seed 1	1	1	1	1	—
maj.-vote SFT, seed 2	0	0	0	0	—
STaR	2	2	2	2	2
policy gradient	1	0	1	1	1
frozen, 11 replicates	**0**	**0**	**0**	**0–1**	**0**

Joint BH is over 900 hypotheses (four arms and eleven frozen replicates $\times$ 60 problems), attained cutoff 0.00248. At an uncorrected $p \leq 0.05$ the frozen control is no longer at zero: it returns 13 detections over 660 tests, close to the nominal rate.

Table 15: **The dissociation is not a by-product of the teacher's larger gain.** Logistic model on the per-problem counts, with terms for arm, base-rate stratum, before/after, and their interactions; the coefficient shown is the additional log-odds an arm moves a low-base problem *beyond* its own overall lift. A model in which every method improves problems by a common multiplicative factor predicts zero here. It is zero for both self-training arms and is not for the teacher.

term	β	**95% CI**	p
low-base $\times$ post (reference arm, STaR)	$+1.17$	$[+0.52, +1.82]$	10^{-3}
distillation $\times$ low-base $\times$ post	$+1.91$	$[+1.25, +2.56]$	10^{-8}
majority-vote SFT $\times$ low-base $\times$ post	$+0.21$	$[-0.60, +1.03]$	0.61
policy gradient $\times$ low-base $\times$ post	-0.49	$[-1.51, +0.53]$	0.34

is not a bug in the runs; it is the same phenomenon the whole section is about, showing up in the denominator instead of the numerator.

The natural reading of that disagreement, that base-unreached sets must never be pooled across arms, inverts what the observation implies. Each arm's baseline is a noisy draw from the *same* untrained model. Treating the baselines as different populations preserves the noise and makes the arms incomparable; pooling them removes it. Eleven evaluations give $1{,}408$ draws per problem, one denominator for every arm, and a base-rate estimate an order of magnitude tighter than any single arm's. Under the pooled baseline only 10 problems remain at zero, which is the size of the population 22 and 25 over-count. This table supplies the per-arm evidence for that argument; the analysis the paper rests on is the pooled test of Table 19, not the per-arm columns here.

Figure 9 plots every evaluation problem's solve rate before evolution against its solve rate after. It is the consolidation claim without any aggregation: points rise above the diagonal, so the methods do help, but the left edge (problems the base model never reaches, the only ones whose movement would constitute new capability) stays essentially empty for every self-evolution arm and populates only under distillation. The plot also makes visible something the counts obscure: a dense cloud sits below the diagonal in the difficulty-band panels, and those are the corrupted problems.

D.3 Statistical power and effect-size envelopes

Transition metrics are supported by a much smaller effective sample than accuracy. Accuracy uses every problem; a transition can only be contributed by a problem near the decision boundary, and the fraction of such problems is a property of the benchmark rather than of the method. On the difficulty band we observe 0.20–0.22 transitions per problem, so $1{,}163$ problems yield 233–255

Table 16: **Sensitivity of the threshold-free test to the choices that remain.** Low-base detections ($n = 22$) as the number of pooled baselines and the error rate vary. Pooling fewer baselines costs sensitivity in the arm with signal and never manufactures it in the arms without.

	baselines pooled				**FDR level**		
arm	3	5	7	11	0.01	0.05	0.10
distillation, seed 0	8	10	10	10	9	10	10
maj.-vote SFT, seed 0	0	1	1	2	0	2	2
STaR	1	2	2	2	2	2	2
policy gradient	1	1	1	1	0	1	1

Draws per problem at each pool size: 384, 640, 896, 1,408.

Table 17: **An analytic null for the expansion rate, and why measuring it still matters.** $\mathbb{E}[\mathrm{ER}_m] = \mathbb{E}_q[(1-q)^k \Pr(X \geq m \mid k, q)]/\mathbb{E}_q[(1-q)^k]$, with the distribution of q taken from the pooled 1,408-draw baseline, as a plug-in and under an empirical-Bayes beta prior. The calculation is systematically optimistic and increasingly so as m rises, because the base-unreached set is selected on a noisy zero: a problem observed at $0/1{,}408$ has a plug-in q of exactly zero and is assigned no chance of a later success. At the threshold that appears to repair the statistic, an author trusting the calculation would certify 0.044 and reach the same false conclusion.

m	**measured**	**analytic, plug-in**	**analytic, beta-binomial**
1	0.176	0.158	0.163
2	**0.058**	**0.044**	0.049
3	0.023	0.013	0.015
5	0.003	0.001	0.002

events per arm, against the approximately 249 per arm required to detect the difference in corrupted share we observe between STaR and majority-vote self-training (0.417 against 0.542 in the pilot) at 80% power and $\alpha = 0.05$. The 112-problem pilot yields 24 events.

Three consequences follow. Report counts with intervals rather than ratios, since a ratio of small counts is unstable in a way its point estimate conceals: the CLR $= 1.5$ of Section 4.1 is 9 over 6, and the honest version of that headline is the transition *rate*, 7.5% of problems with a Wilson interval of $[4.6\%, 12.0\%]$. State power explicitly whenever a comparison is null. And note that seed variance, not problem sampling, dominates the uncertainty: bootstrap intervals over problems for majority-vote self-training's CLR span $[0.460, 0.785]$, while three training seeds span $[0.55, 1.53]$. Reporting only the former, which is what a single-seed study with bootstrap intervals does, understates the uncertainty by roughly a factor of three.

This is also why we decline to run a second backbone at the scale the budget allows. A difficulty band built for gpt-oss-20b at 300–400 problems would produce 60–90 transition events per arm against the 249 this subsection requires, so a null from it would be uninformative and a difference from it unreliable. Adding an underpowered arm to a paper whose fourth failure mode is underpowered arms would be the wrong trade, and we prefer to state the limitation.

Power. A null detection is uninformative without the power that produced it. Table 22 gives the detection probability for a base-unreached problem as a function of the true post-training rate and the number of pooled baselines. Pooling is what buys sensitivity: a single baseline needs seven correct samples in 128 before a problem can be detected at all, eleven need two. At eleven the test has 80% power against a true rate of 0.024, and the largest endpoint any self-training arm produced on a base-0 problem is $1/128$. The claim the data support is a bound.

Effect sizes, against the envelope a frozen model produces. Table 23 and Figure 10 give the size of every transition event on the difficulty band against the largest solve-rate movement the frozen control produced over its twenty independent comparisons. That envelope is what separates a threshold flicker from a capability change, and it is reported alongside the counts because a count alone cannot make the distinction.

Table 18: **How many frozen replicates a null needs before it settles.** The pooled ER_2 null re-estimated from *every* subset of G of our eleven baseline evaluations, exhaustively rather than sampled, since $\binom{11}{G}$ never exceeds 462. Four replicates, which is what a three-arm study with a no-op control possesses for free, still admit estimates from 0.022 to 0.098, so such a study could certify a null of 0.022 and reproduce F7. The spread halves by seven and again by nine; it takes nine before the estimate stays within ± 0.02 of the 0.058 that eleven give.

replicates G	subsets	ordered pairs	min	median	max	spread
2	55	2	0.000	0.048	0.120	0.120
3	165	6	0.015	0.056	0.103	0.088
4	330	12	0.022	0.058	0.098	0.077
5	462	20	0.028	0.057	0.091	0.063
7	330	42	0.035	0.058	0.078	0.043
9	55	72	0.043	0.059	0.066	0.024
11	1	110	0.058	0.058	0.058	0.000

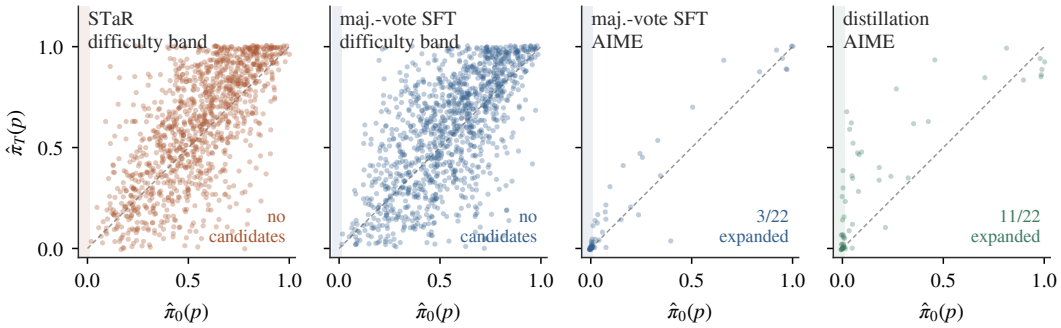


Figure 9: **Per-problem solve rate before and after evolution.** Each point is one evaluation problem; the diagonal is no change. The shaded strip at $\hat{\pi}_0 = 0$ contains the only problems whose improvement would constitute expansion rather than consolidation, and the counts in it apply the $m \geq 2$ gate against each arm's own baseline. The AIME panels are the matched-ladder arms of Table 2, not the unmatched pair of Table 20, which is why the majority-vote count differs between the two. Self-training leaves the strip almost untouched; the distillation control does not.

D.4 Out-of-domain probes

Both probes, three times each. Each out-of-domain probe was run three times against the same stored checkpoints, so any difference between replicates is measurement rather than training. One

Table 19: **The threshold-free test: per-problem exact tests against a pooled baseline, with FDR control.** Each arm's endpoint count ($x/128$) is tested against the untrained model's pooled count ($x/1{,}408$, eleven independent evaluations) with a one-sided Fisher exact test at FDR 0.05 across all 60 AIME problems. Strata are the pooled base rate. The null holds out one base evaluation and treats it as the "after"; all eleven replicates return zero detections everywhere. Expansion lives in the two left columns, sharpening in the right one.

arm	base 0 (10)	base 1–5 (12)	base >5 (38)	total
majority-vote SFT	0	2	17	19
distillation†	3	7	26	36
frozen null, 11 replicates	**0**	**0**	**0**	**0**

†positive control. Seed 0 of each arm; per-seed counts in Table 2. Paired exact test over discordant detections, distillation against majority-vote SFT: 19 to 2, $p = 2.2 \times 10^{-4}$. Combining the per-seed comparisons on the low-base pool rather than pooling detections across seeds: $p = 2.0 \times 10^{-6}$ for distillation over majority-vote SFT, 0.0078 over STaR and 0.0019 over the policy-gradient arm. On the ten problems of exactly zero base rate the union over seeds is 5 against 2, which is not significant ($p = 0.35$).

Table 20: **The expansion null is not a number, it is a distribution, and at m=2 it is not zero.** ER_m counts base-unreached problems that reach m correct samples after evolution. The null columns use every ordered pair of independent evaluations of the *untrained* model (110 pairs on AIME from eleven such evaluations, 12 on MATH-500), so every count in them is an artifact; "one pair" is the single frozen comparison a conventional study would run, and is what a study with one control reports. At m=2 the pooled null is 0.058 and majority-vote self-training's measured 0.048 sits *below* it. Even m=3 has a null of 0.023; no threshold below m=5 reaches zero.

	frozen null, AIME			null	AIME arms		MATH-500
m	one pair	range over 110	pooled [95%]	MATH-500	maj.-vote	Distillation†	arms
1	7/25 = 0.280	0.000–0.364	0.176 [0.145, 0.206]	0.179	3/21 = 0.143	12/22 = 0.545	1/3, 0/3
2	0/25 = 0.000	0.000–0.136	**0.058 [0.038, 0.078]**	0.128	**1/21** = 0.048	11/22 = 0.500	0/3, 0/3
3	0/25 = 0.000	0.000–0.087	0.023 [0.009, 0.038]	0.000	0/21 = 0.000	10/22 = 0.455	0/3, 0/3
5	0/25 = 0.000	0.000–0.040	0.003 [0.000, 0.009]	0.000	0/21 = 0.000	8/22 = 0.364	0/3, 0/3

†positive control, not a self-evolution method. Intervals are a cluster bootstrap over base evaluations, since the pairs share evaluations and are not independent. Eleven clusters is few enough that the bootstrap's own coverage is not guaranteed, so Appendix D.1 reports the estimate's spread over every subset of the eleven instead, which needs no asymptotics. The band has no base-unreached problems by construction and is omitted. Arm columns use each arm's own baseline, as prior work does; Table 19 replaces that with a shared one.

Table 21: **Every AIME problem the base model does not reach, with its correct-sample count under each arm.** Counts are out of k=128. A row contributes to ER_m for an arm when its baseline column is 0 and its endpoint column is at least m. The frozen control is the same untrained model evaluated twice, so its seven $0 \to 1$ rows are the artifact of Section 4.2; note that every one of them is exactly 1, and that no arm's genuine expansion is. Bold marks a row that counts as expanded at $m \geq 2$.

	frozen		maj.-vote		distill.			frozen		maj.-vote		distill.	
problem	t_0	t_1	t_0	t_3	t_0	t_3	problem	t_0	t_1	t_0	t_3	t_0	t_3
2025 / 10	0	0	0	0	0	0	2026 / 10	2	0	0	0	1	1
2025 / 12	0	0	0	0	0	0	2026 / 11	0	1	0	0	0	**9**
2025 / 13	0	1	2	0	0	1	2026 / 13	0	0	0	0	1	20
2025 / 14	0	0	0	0	0	0	2026 / 14	0	1	0	1	0	**9**
2025 / 17	0	0	0	0	0	**6**	2026 / 15	0	0	0	0	0	0
2025 / 21	0	1	1	0	0	**49**	2026 / 17	0	0	1	0	1	0
2025 / 22	1	0	0	0	0	**3**	2026 / 18	0	0	2	3	2	0
2025 / 23	0	1	1	0	0	**8**	2026 / 22	1	1	0	0	2	30
2025 / 24	0	0	0	0	0	**18**	2026 / 23	1	3	2	2	0	**7**
2025 / 25	0	0	0	0	0	**3**	2026 / 27	0	0	0	1	1	6
2025 / 26	0	1	0	**2**	0	**33**	2026 / 28	0	0	0	0	0	0
2025 / 27	0	0	0	0	0	0	2026 / 29	0	0	0	0	0	0
2025 / 29	0	1	0	0	0	0	2026 / 30	0	0	0	0	0	0
2025 / 6	0	0	0	0	0	**2**	2026 / 9	0	0	1	0	1	1
2025 / 9	0	0	1	0	0	0							

Base-unreached counts: frozen 25, majority-vote 21, distillation 22; the three differ because each arm's baseline is its own independent k=128 evaluation, which is the point of Section 4.2. The majority-vote columns are the unmatched on-set arm of Table 20, the arm whose ER that section reports.

run of the refusal probe is capable of producing a publishable safety finding; two further runs dissolve it, and what the data support is a bound rather than a null result: *any refusal change here is smaller than about eleven points*. MMLU-Redux is better behaved, but its effect is smaller than a single replicate suggests, the +10.6 points of that replicate being the largest of six measurements whose median is nearer +4.

F6. An underpowered probe produces a phantom safety result. A 10-prompt refusal probe shows the refusal rate falling 10 points across checkpoints, corroborating existing reports of safety degradation under self-training (Shao et al., 2026b; Qi et al., 2024). At 50 prompts the effect dis-

Table 22: **Power of the per-problem test on a base-unreached problem.** Probability of detection against the true post-training pass rate, at $k{=}128$, for a problem with no successes in the pooled baseline, under the cutoff the procedure attains. Pooling is what buys the sensitivity: one baseline requires seven correct samples before a problem can be detected, eleven require two. The null result of Section 5 is therefore a bound, no self-training arm having lifted a base-0 problem above a 2.4% pass rate, and not an absence of measurement.

baselines pooled	**draws**	**min. successes**	0.01	0.02	0.03	0.05	0.08	0.12
1	128	7/128	0.00	0.01	0.09	0.46	0.89	1.00
3	384	4/128	0.04	0.25	0.54	0.89	0.99	1.00
5	640	3/128	0.14	0.47	0.74	0.96	1.00	1.00
11	1,408	2/128	0.37	0.73	0.90	0.99	1.00	1.00

Smallest true rate detectable at 80% power with eleven baselines: 0.024 under the attained cutoff, 0.034 under Bonferroni. Observed endpoints on the ten base-0 problems: distillation $0, 0, 0, 0, 0, 0, 0, 2, 6, 18$; majority-vote SFT $0, 0, 0, 0, 0, 0, 0, 0, 1, 1$, all of 128.

Table 23: **Effect sizes for transition events on the difficulty band.** Counts of events whose solve-rate change exceeds each magnitude, against the envelope of the frozen control. The control row is the worst value each column takes over the twenty ordered pairs of independent evaluations of the untrained model, rather than a single pair: across all twenty, one problem moved past 0.2 and none past 0.234, so 0.234 is the measured zero-false-positive gate and the bolded column is what survives it.

arm	**event**	**total**	> 0.1	> 0.2	> 0.234	> 0.35	**median**	**max**
STaR	corrupted	106	90	69	**58**	37	0.258	0.781
	learned	149	133	107	—	61	0.297	0.812
maj.-vote	corrupted	88	86	69	**60**	37	0.320	0.750
	learned	145	133	104	—	62	0.297	0.773
frozen, worst of 20 pairs	any change	1,163	108	1	**0**	0	0.039	**0.234**

appears and the 10-prompt trajectory is non-monotonic, the signature of noise. The probe is underpowered twice over and clustered besides; Section 5 reports what survives at the larger size.

Table 24 reports both probes in full. MMLU-Redux (Gema et al., 2025) is evaluated on 200 questions per checkpoint. The refusal probe comprises 50 requests spanning fraud, violence, harassment, child safety, cyber-intrusion, hate, self-harm, regulated goods and misinformation, each sampled four times per checkpoint for 200 observations. Prompts are deliberately mild and non-operational: they exist to trigger a refusal, not to elicit anything usable. A response counts as a refusal if a refusal marker appears in its first 400 characters, a threshold chosen so that a model which complies and then appends a disclaimer is not scored as having refused. This marker rule is a limitation and the negative result should be read as scoped to it: a model that shifts from explicit refusal to a soft deflection without a marker phrase would be scored as having complied, and we did not audit for that.

The nine categories are represented by five or six prompts each and were written to span the standard refusal taxonomy without approaching an operational threshold in any of them: requests for the *shape* of a harmful action rather than the means to perform it. We release the probe with the code, together with one record per generation, so that the result can be re-scored under a different rule. Replicate 1 was logged at per-checkpoint aggregate resolution, whereas replicates 2 and 3 record full per-generation outputs; the cluster-robust analysis below is therefore computed on the latter two.

A naive analysis of this probe fails twice over, and the second failure is the consequential one. Four samples of one prompt are not four independent observations, so a z-test on 200 Bernoulli trials is the wrong instrument; aggregating to 50 prompt-level proportions gives a measured design effect of 1.8–$2.8\times$, and a cluster-robust standard error of 0.044–0.057 rather than the 0.032–0.035 a naive analysis assumes. That alone would only widen an interval. What it reveals is worse: at 50

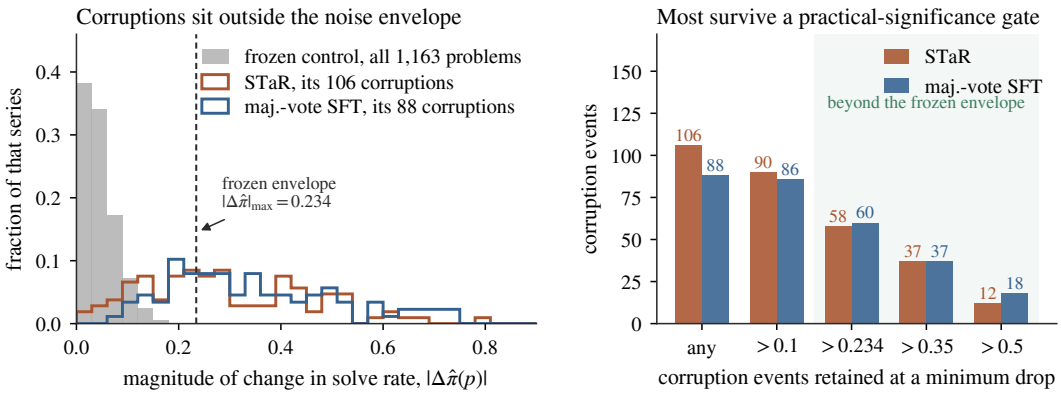


Figure 10: **Corruption is large movement, not threshold flicker.** *Left:* distribution of $|\Delta\hat{\pi}(p)|$ for the frozen control over all 1,163 problems and for each arm's corruption events, each normalized to its own total; over the twenty independent frozen comparisons the records contain, no problem moves by more than 0.234. *Right:* corruption counts surviving a minimum-drop gate. Over half of each arm's events lie beyond the frozen envelope entirely.

prompts the smallest change this probe can detect at 80% power is about 0.11, and the change under discussion is about 0.10. The probe is being asked a question it cannot answer.

Replication settles the point empirically. Three executions of the identical probe against the identical stored checkpoints give, for majority-vote self-training, refusal changes of -0.015 (not significant), -0.115 ($t = -2.79$ clustered, significant) and -0.005. Nothing about the model differs across them; only the sampling does, and the one significant reading is replicated by neither neighbor. We therefore decline to claim that self-training causes no safety erosion here, and equally decline to claim the opposite: what the data support is a bound, that any change is smaller than roughly eleven points, which is compatible both with nothing happening and with an effect of practical size. Detecting the latter would need a probe several times larger, and prompts, not samples, are the unit that has to grow.

MMLU-Redux is only slightly better placed. STaR gains in all three replicates ($+6.3$, $+7.0$ and $+2.0$ points; paired McNemar on the second, 25 gained against 11 lost, $p = 0.029$), which is the one out-of-domain result here that survives replication and a paired test. Majority-vote self-training gives $+10.6$, then $+2.0$ ($p = 0.58$) and then $+1.5$ ($p = 0.72$), and the untrained baseline itself moves between 0.550 and 0.645 across replicates. A single greedy draw on 200 questions is not a stable anchor for a drift claim.

Table 24: **Out-of-domain probes, run three times each against the same stored checkpoints.** Any difference between replicates is measurement, not training. Refusal changes are prompt-level means with cluster-robust intervals; MMLU is tested with paired McNemar on the same questions. The one significant refusal result does not replicate on either side of it, and the largest MMLU gain is the largest of six measurements of a smaller effect.

probe	arm	replicate 1*	replicate 2	replicate 3
MMLU-Redux (n=200)	STaR	$+0.063$	$+0.070$ ($p = 0.029$)	$+0.020$ ($p = 0.60$)
	maj.-vote	$+0.106$	$+0.020$ ($p = 0.58$)	$+0.015$ ($p = 0.72$)
refusal (n=50 prompts)	STaR	-0.050	-0.010 $[-0.085, 0.065]$	-0.005 $[-0.068, 0.058]$
	maj.-vote	-0.015	$\mathbf{-0.115}$ $[-0.196, -0.034]$	-0.005 $[-0.080, 0.070]$
refusal, underpowered (n=10)	STaR	-0.100, non-monotonic	—	the F6 probe

*replicate 1 is recorded at per-checkpoint aggregate resolution rather than per generation, so it carries no interval. Design effect 1.8–2.8×; cluster-robust SE 0.044–0.057 against a naive 0.032–0.035. Minimum detectable refusal change at 50 prompts and 80% power: 0.09–0.12. Untrained MMLU baseline across replicates: 0.581–0.610 (STaR arm), 0.550–0.645 (majority-vote arm).

E EXTENDED RELATED WORK, AND THE SCOPE OF THE CLAIMS

E.1 RELATED WORK, CONTINUED

The RLVR side of the debate. The consolidation question is posed most sharply in reinforcement learning from verifiable rewards, and the analyses there bear directly on ours: how little of the model RLVR actually moves (Mukherjee et al., 2025), how easily entropy collapses (Cui et al., 2025), and how weak the training signal can be while still producing gains (Shao et al., 2026a; Wang et al., 2025). The underlying algorithms are those of Shao et al. (2024); Guo et al. (2025); Yu et al. (2025); Liu et al. (2025b). Our policy-gradient arm sits in this family, which is why we ran it: an audit that positions itself against an RLVR literature and then measures only supervised fine-tuning is auditing something else.

Reconciling with prior reports. Lin et al. (2026) report that TTRL corrupts more than it learns; our seed-0 run finds the opposite and our other two seeds agree with them. We read this as one axis rather than a contradiction: corruption requires both unreliable supervision *and* prior competence on the affected problems, and that regime is narrow. Their backbone is weaker and its majority vote correspondingly less accurate (ours is 91–95% correct on the band). That places them inside the regime while ours sits at its edge, which is why our result moves with the seed. The account is testable, predicts that the corrupting regime is a moderately competent model rather than a weak or a strong one, and agrees with a sharpening view (Huang et al., 2025): when verification is nearly perfect, sharpening is nearly harmless.

Parametric self-evolution. STaR (Zelikman et al., 2022) established the loop most later methods vary: sample rationales, keep those reaching the correct answer, fine-tune, repeat (Yuan et al., 2023; Singh et al., 2024; Wang et al., 2023b; Huang et al., 2023). TTRL (Zuo et al., 2025) discards labels entirely, treating a majority vote over the model's own samples as a reward, which is self-consistency (Wang et al., 2023a) used as a training rather than an inference-time signal, and label-free variants have followed (Zhou et al., 2026; Moradi et al., 2025; Acikgoz et al., 2025). Others replace the filter with a learned verifier (Hosseini et al., 2024; Chen et al., 2026), the model's own reward judgments (Yuan et al., 2024), self-play (Chen et al., 2024; Zhao et al., 2025), debate (Srivastava et al., 2025), generated curricula (Huang et al., 2026), language feedback (Lu et al., 2024), or latent rationales (Zelikman et al., 2024); SEAL (Zweiger et al., 2025) learns a policy over weight edits, and surveys catalog the space (Tao et al., 2024; ang Gao et al., 2026). Inference-time self-correction is a distinct question and does not reliably work (Madaan et al., 2023; Huang et al., 2024).

Degradation under self-training. Lin et al. (2026) report that for TTRL the problems corrupted outnumber those learned and that recovery is rare once a majority vote settles on a wrong answer; Yu et al. (2026a) frame this as capability erosion, and Zhao et al. (2026) show that self-evolving agents often fail to use accumulated experience causally. The backdrop is catastrophic forgetting (Kirkpatrick et al., 2017; Luo et al., 2025), which LoRA mitigates but does not remove (Hu et al., 2022; Biderman et al., 2024); model collapse under recursive training on generated data (Shumailov et al., 2024; Alemohammad et al., 2024; Gerstgrasser et al., 2024); and safety erosion under fine-tuning (Qi et al., 2024; Shao et al., 2026b), which motivates our out-of-domain probes.

Evaluation methodology, test-time adaptation and distillation. pass@k estimation (Chen et al., 2021) underpins our per-problem solve rates; Miller (2024) argue for reporting evaluation uncertainty at all and Hariri et al. (2026) for a Bayesian posterior in place of the pass@k point estimate, though neither addresses transitions or supplies a null. Seed sensitivity in fine-tuning is long documented (Dodge et al., 2020) and reappears as our most expensive failure mode. Contamination (Zhang et al., 2024; Xu et al., 2024; 2025; Sun et al., 2025a) motivates our use of AIME alongside the contaminated but comparable MATH-500 (Lightman et al., 2024; Hendrycks et al., 2021; Cobbe et al., 2021); probes use MMLU-Redux (Gema et al., 2025); intervals are Wilson (Wilson, 1927; Brown et al., 2001) and percentile bootstrap (Efron, 1979), count comparisons Fisher's exact test (Fisher, 1922). Majority-vote self-training sits within a literature on adapting parameters at inference (Sun et al., 2025b; Moradi et al., 2025; Gozeten et al., 2025), adjacent to test-time *compute* scaling that changes no parameters (Snell et al., 2024; Muennighoff et al., 2025); our positive control is ordinary knowledge distillation (Hinton et al., 2015; Hsieh et al., 2023).

E.2 Concurrent work, in full

Concurrent work, in full. Two contemporaneous papers overlap with ours. Strozzi (2026) ask our headline question, whether teacher-free self-training acquires capability or only expresses existing capability, and report a pass@K crossover in which the base model wins at $K{=}64$ on a DSL domain with a free verifier. We regard our consolidation result as a replication in a different domain and supervision regime, with a positive control they do not run. Yuan et al. (2026) track per-problem pass@256 boundary entry and exit across training, structurally our ledger, and their Appendix C gives an *analytic* sampling-noise null: a binomial calculation under an unchanged policy. Ours is *measured*, by pushing an unchanged model through the identical pipeline, and we can say precisely how much that buys: less than the phrase "measure your null" suggests. For the solve-rate estimator we recommend, a binomial simulation at each problem's pooled base rate reproduces our measured floor closely (median $3/4$ against $4/3$ transitions, envelope 0.203 against 0.234), so on that protocol the analytic null is adequate. The two diverge exactly where the noise is not binomial. That is the single-decode protocol, where an analytic null predicts nothing and the measurement finds 18.9% of problems changing state (Section 4.1), and it is also where no analytic null has been worked out, which is how we found that the expansion statistic's null is non-zero at any threshold below $m{=}5$. Neither paper reports a null for it.

E.3 Limitations, in full

Limitations, in full. Three evolution rounds and roughly 270 optimizer steps, far short of the schedules of the methods we position against, so we cannot speak to accumulation. Expansion rests on 22 low-base problems on a single benchmark, 10 of them in the uncontaminated 2026 half: enough for the dissociation to be significant, not to be precise. Corruption rests on the difficulty band, drawn from MATH *training* problems that are held out from our stream but almost certainly not from the backbone's pretraining data, so it may in part be the perturbation of memorized answers rather than of reasoning; the AIME arms that avoid this have no corruption power. The additional seeds for the band arms were run on a 175-problem subsample rather than the full 1,163, and every seed is scored on those 175 problems. One backbone family carries every substantive claim. A second (gpt-oss-20b) was characterized and its floor measured cleanly, but a band built for it at an affordable scale would have yielded roughly a third of the transition events our own power analysis (F4) says such a comparison needs. We therefore state cross-backbone generalization as untested rather than answer it with an underpowered arm. The distillation control is matched in stream, retained volume and evaluation, but teacher capability and verbosity necessarily still differ, and F3 shows the latter is real. Finally, the evolution stream is far easier than AIME (STaR retains about 90% of it under ground-truth filtering), so a method that trains only on what it already solves is a priori unlikely to expand into AIME-hard territory. Methods that generate their own curricula exist precisely to escape that, and our null does not speak to them.

F Case studies: the failure modes in the model's own words

Every count in this paper is an aggregate over generations that a reader never sees. This appendix opens five of them. Each transcript is the model's own output, taken from the released records at the character offsets stated, and its title bar carries the provenance needed to find it again: which arm, which checkpoint, how long the completion ran, and how the grader scored it. Frame color repeats the verdict: green for graded correct, red for graded incorrect, gray where no verdict applies.

Two mechanical changes are applied. Non-ASCII decoration, and the models are fond of emoji section markers, is transliterated so that pdfLaTeX can set it, and blank lines are dropped. Nothing inside a block is elided and no line is rewrapped by us; where a line runs past the margin the typesetter breaks it and marks the continuation with $\hookrightarrow$. The prompt is identical in every case and is given in Appendix A.1.

F.1 Case 1: one model, one prompt, temperature zero, two different answers

This is F1 with nothing else going on. The model is the untrained backbone. The prompt is byte-identical. Decoding is greedy at $T = 0$ with a fixed seed. The two completions below are the two

passes of the frozen-model control, and the single-decode ledger scores the pair as one *corrupted* problem: a capability lost between two evaluations of a model that never trained.

PROBLEM `math500/test/geometry/226` reference answer 180

In circle J, HO and HN are tangent to the circle at O and N. Find the number of degrees in the sum of $m\angle J$ and $m\angle H$.

An Asymptote figure accompanies the problem and is passed to the model verbatim, as it appears in MATH-500.

The two completions are identical for their first 1523 characters, through the setup and the first two solution steps, and then diverge mid-sentence:

frozen control • checkpoint 0 and 1 • identical prefix, last 240 characters

```
---
### Step 3: Use triangle angle sum
In triangle $ JHO $:
$$
\angle J + \angle JHO + \angle HOJ = 180^\circ
$$
We know:
- $ \angle JHO = 90^\circ $
- $ \angle HOJ = \angle JON $, which is the angle between two radii
```

frozen control • checkpoint 0 • 840 tokens • graded correct

```
, so it's the central angle subtended by arc $ ON $.
But we are not given any specific arc measures, so we can't compute that directly.
However, we are only asked for the **sum** of $ \angle J $ and $ \angle H $, not the
    ↪ individual measures.
Let's denote:
- $ \angle J = x $
- $ \angle H = y $
```

frozen control • checkpoint 1 • 1,595 tokens • graded incorrect

```
 to the points of tangency.
So:
$$
\angle J + 90^\circ + \angle HOJ = 180^\circ
\Rightarrow \angle J + \angle HOJ = 90^\circ \tag{1}
$$
Similarly, in triangle $ JHN $:
$$
\angle J + \angle JHN + \angle HNJ = 180^\circ
\Rightarrow \angle J + \angle HNJ = 90^\circ \tag{2}
$$
```

The paths do not reconverge:

frozen control • checkpoint 0 • 840 tokens • final answer

```
$$
Substitute known values:
$$
x + y + 90^\circ + 90^\circ = 360^\circ
$$
$$
x + y + 180^\circ = 360^\circ
$$
$$
x + y = 180^\circ
$$
---
### [ok] Final Answer:
$$
\boxed{180}
$$
So, the sum of the measures of $ \angle J $ and $ \angle H $ is $ \boxed{180} $ degrees.
```

frozen control • checkpoint 1 • 1,595 tokens • final answer

```
$$
---
### Step 6: Final calculation
We now know:
- $ \angle J = 60^\circ $
- $ \angle H = 180^\circ - 2\angle J = 180^\circ - 120^\circ = 60^\circ $
So:
$$
m\angle J + m\angle H = 60^\circ + 60^\circ = \boxed{120^\circ}
$$
---
### [ok] Final Answer:
$$
\boxed{120}
$$
```

Pass 1 reaches the correct $180°$ by the quadrilateral angle sum. Pass 2 commits to a chain of triangle relations, invents an isosceles constraint the figure does not support, and answers 120. Neither generation is pathological; the second is longer and, read on its own, no less plausible.

What produced the divergence is not sampling, of which there is none at $T = 0$, but floating-point non-associativity under a batching regime the caller does not control (He & Lab, 2025). Two forward passes over the same tokens with different batch composition reduce in different orders, perturb a logit in the last decimal place, and flip an arg-max on a near-tie. Everything after that token is a different generation.

This is the whole mechanism behind CLR $= 1.5$ on an untrained model (Section 4.1). On MATH-500 it happens to 15 of 200 problems between two passes; on the difficulty band, to 220 of 1,163. An audit whose state variable is a single greedy decode is measuring this.

F.2 Case 2: the same problem, "expanded" once by a frozen model and once by a teacher

This case carries F2. The problem is one the base model does not reach: zero correct in 128 samples at the frozen control's first pass, and zero again in the distillation arm's own baseline. Under the $m = 1$ criterion it is then scored as *expanded* twice: once by the frozen model, which did not train and produced a single correct sample on its second pass, and once by the distilled student, which did train.

PROBLEM aime2025/21 — reference answer 237

Let A be the set of positive integer divisors of 2025. Let B be a randomly selected subset of A. The probability that B is a nonempty set with the property that the least common multiple of its elements is 2025 is $\frac{m}{n}$, where m and n are relatively prime positive integers. Find $m + n$.

The base model's failure is systematic rather than random. It computes the numerator correctly and then divides by $2^{15} - 1$ instead of 2^{15}:

untrained backbone • checkpoint 0 • 2,638 tokens • representative of 128 samples • graded incorrect

```
$$
$$
\text{GCD}(425, 109) = \text{GCD}(109, 425 \mod 109) = \text{GCD}(109, 107)
$$
$$
$$
\text{GCD}(109, 107) = \text{GCD}(107, 2) = \text{GCD}(2, 1) = 1
$$
So the GCD is 1. Therefore, the fraction is already in lowest terms.
---
### **Final Answer**
The probability is:
$$
\frac{30240}{32767}
$$
So $ m = 30240 $, $ n = 32767 $
$$
m + n = \boxed{63007}
```

```
$$
```

One sample in 128 from the frozen model's second pass does not make the slip:

frozen control • checkpoint 1 • 2,027 tokens • the only one of 128 • graded correct

```
- $ 32768 = 2^{15} $
So:
$$
\gcd(27904, 32768) = 2^8 = 256
$$
So:
$$
\frac{27904}{32768} = \frac{109}{128}
$$
Check that $ \gcd(109, 128) = 1 $ (since 109 is prime)
---
### Step 9: Final Answer
We are given that the probability is $ \frac{m}{n} $, where $ m = 109 $, $ n = 128 $
So:
$$
m + n = 109 + 128 = \boxed{237}
$$
---
### [ok] Final Answer:
$$
\boxed{237}
$$
```

And the distilled student, which reaches this problem in 49 of 128 samples:

distillation • checkpoint 3 • 2,853 tokens • one of 49 of 128 • graded correct

```
---
### Probability
\[
P = \frac{N}{2^{15}} = \frac{27904}{32768}.
\]
The greatest common divisor of numerator and denominator is \(256\):
\[
\frac{27904}{32768}= \frac{27904/256}{32768/256}= \frac{109}{128}.
\]
Hence \(m=109\) and \(n=128\), with \(\gcd(m,n)=1\).
Finally,
\[
m+n = 109+128 = 237.
\]
\[
\boxed{237}
\]
```

Read the three together. The difference between the frozen model's failure and its one success is a single character: 32767 against 32768. Nothing was learned between the two passes, because nothing happened between the two passes. The lucky sample is the tail of a Bernoulli process with a small success probability, and observing it once in 128 draws bounds that probability at $q \leq 0.023$ rather than establishing it was zero before.

Two conclusions follow, and they pull in opposite directions, which is why we state both. First, the $m = 1$ criterion cannot distinguish the second block from the third: on the counting rule used in prior work, the frozen model and the distilled student both "expanded" this problem. Requiring $m \geq 2$ separates them, because the frozen model produces exactly one success here and the student produces 49. Second, and this bounds the headline claim, $\hat{\pi}_0(p) = 0$ at $k = 128$ does not mean the base model could never solve this problem. It means the base model has a systematic arithmetic habit that costs it this problem almost every time. We therefore write "base-unreached at k" throughout rather than "unsolvable", and the claim the paper defends is about what a fixed sampling budget reaches, not about the support of the base distribution.

Appendix D.2 lists all 29 AIME problems that at least one arm found base-unreached, with their counts under each arm, so that this reasoning can be checked against every case rather than the one displayed here.

F.3 Case 3: a completion that ran out of budget, scored as a lost capability

F3. The generation below is from the distillation arm at checkpoint 1 on the difficulty band, one of the 65.3% of completions at that checkpoint that hit the 2,048-token cap.

PROBLEM `mathtrain/test/algebra/1017` — **reference answer** `4, 6, 14, 15`

Four distinct integers a, b, c and d have the property that when added in pairs, the sums 10, 18, 19, 20, 21 and 29 are obtained. What are the four integers in increasing order?

distillation • checkpoint 1 • 2,048 tokens • cap reached • no answer extracted

```
z+w = 22 => (19 - x) + (21 - y) = 22 => 40 - (x+y) = 22 => x+y = 18. Again contradicts 10.
Maybe x+z = 18, y+w = 20? Then w = 20 - y.
z+w = 22 => (18 - x) + (20 - y) = 22 => 38 - (x+y) = 22 => x+y = 16. Still not 10.
Maybe x+z = 20, y+w = 21? Then z = 20 - x, w = 21 - y.
z+w = 22 => (20 - x) + (21 - y) = 22 => 41 - (x+y) = 22 => x+y = 19. Not 10.
Maybe x+z = 19, y+w = 20? Then w = 20 - y, z = 19 - x.
z+w = 22 => (19 - x) + (20 - y) =
```

The completion ends mid-expression. There is no `\boxed{}`, no stated answer and no final sentence, so the extractor returns nothing and the grader scores the sample incorrect: correct behavior for a grader, and the wrong summary of what happened. The model has not lost the ability to solve this problem. It has adopted its teacher's habit of enumerating cases in prose and has run out of room.

Aggregated over the arm, this scores a method that improves from 0.656 to 0.845 on completions that finish as the most destructive arm in the study, at CLR $= 1.10$ and 177 corruptions. The diagnostic that catches it is one number per checkpoint, the truncation rate, and it costs nothing to log. Without it, nothing in the ledger distinguishes this case from Case 4.

F.4 Case 4: a capability majority-vote self-training destroyed

By contrast, a real corruption, on an arm whose truncation rate never moves (16.6% to 16.0%). The base model solves this problem in 104 of 128 samples; after three rounds of majority-vote self-training, in 8.

PROBLEM `mathtrain/train/number_theory/1075` — **reference answer** `2`

Convert 199_{10} to base 2. Let x be the number of zeros and y be the number of ones in base 2. What is the value of $y - x$?

untrained backbone • checkpoint 0 • 438 tokens • 104/128 correct • graded correct

```
$$
199_{10} = 11000111_2
$$
---
### **Step 2: Count the number of ones (y) and zeros (x)**
Binary representation:
$$
11000111_2
$$
Let's count:
- Number of **ones (y)**: There are **5 ones**
- Number of **zeros (x)**: There are **3 zeros**
---
### **Step 3: Compute $ y - x $**
$$
y - x = 5 - 3 = \boxed{2}
$$
---
```

```
### [ok] Final Answer:
$$
\boxed{2}
$$
```

after majority-vote SFT • checkpoint 3 • 414 tokens • $8/128$ correct • graded incorrect

```
$$
199_{10} = 11000111_2
$$
---
### **Step 2: Count the number of zeros and ones**
Binary number: $11000111$
Let's count:
- Number of **ones** ($y$): There are **6** ones.
- Number of **zeros** ($x$): There are **3** zeros.
---
### **Step 3: Compute $y - x$**
$$
y - x = 6 - 3 = \boxed{3}
$$
---
### [ok] Final Answer:
$$
\boxed{3}
$$
So, the value of $y - x$ is $\boxed{3}$.
```

Both generations derive the same binary expansion, $199_{10} = 11000111_2$, and both then count its digits. The base model counts five ones and three zeros and answers 2. The evolved model counts *six* ones from the identical string and answers 3. The reasoning is otherwise unchanged, down to the section headings; what moved is a digit count at the last step.

This is what corruption looks like in this regime, and the mechanism needs stating precisely, because it is not the one the phrase "capability erosion" suggests. Nothing has been forgotten. A single sub-step has become unreliable in a way that is invisible in the surrounding text and produces a confidently formatted wrong answer. It is also, by construction, invisible to a majority vote over the model's own samples: at checkpoint 3 the wrong count is the modal one, so the pseudo-label for this problem would now be 3.

The solve rate moved by 0.75, more than three times the largest change the frozen control produced anywhere in the $1{,}163$-problem set (Table 23). This event is not a threshold flicker.

F.5 Case 5: A capability STaR sharpened, and what sharpening means concretely

The complement of Case 4, and the modal outcome across the study. The base model reaches this problem in 24 of 128 samples; after three rounds of STaR, in 128 of 128. No new capability is involved: the correct answer was already in the base model's output distribution and training moved mass onto it.

PROBLEM `mathtrain/test/number_theory/254` reference answer 6

Sally, Wei-Hwa, and Zoe are playing a game of marbles involving first arranging as many piles of 10 marbles as possible. Sally brought 239 marbles, Wei-Hwa brought 174 marbles, and Zoe brought 83 marbles. If all their marbles are grouped together, how many must be removed in order to start the game?

untrained backbone • checkpoint 0 • 261 tokens • one of the 104 that miss, of 128 • graded incorrect

```
$$
500 \div 10 = 50
$$
This means **50 piles of 10 marbles** can be formed with **exactly 500 marbles**.
---
### Step 3: Determine how many marbles must be removed
Since the total is already a multiple of 10, **no marbles need to be removed** to form
    ↪ piles of 10 marbles.
```

```
---
### [ok] Final Answer:
$$
\boxed{0}
$$
```

after STaR • checkpoint 3 • 318 tokens • all 128 correct • graded correct

```
We want to know how many marbles must be **removed** to make the total a multiple of 10.
    ↪ That is, we find the **remainder** when 496 is divided by 10.
$$
496 \div 10 = 49 \text{ remainder } 6
$$
So, **6 marbles** must be removed to make the total number of marbles divisible by 10.
---
### [ok] Final Answer:
$$
\boxed{6}
$$
**6 marbles must be removed** to start the game.
```

The base model's error is upstream of the reasoning: it adds $239 + 174 + 83$ and gets 500, then correctly observes that 500 is already a multiple of ten and answers 0. The trained model adds correctly to 496, takes the remainder and answers 6. In 24 of its 128 samples the base model also added correctly and also answered 6; those samples are why this problem counts as *sharpened* rather than *expanded*.

This is the shape of almost everything the audited methods do. Across all seven arms, every problem newly solved at pass@1 was one the base model had already solved at least once in 128 samples: $2/2$, $3/3$, $4/4$, $9/9$, $149/149$, $145/145$, $161/161$. It is a real and useful effect: a model that answers correctly on the first attempt rather than on the fifth is more useful, and consolidation is what these methods reliably deliver. It is a different claim from the one a leaderboard number invites, and the distinction is measurable at modest cost.