SMALL STEPS ON THE GEODESIC: ABILITY-ANCHORED PROGRESSIVE CURRICULA FOR RL ON REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Curriculum design for RL with verifiable rewards (RLVR) is usually framed as a per-step problem: keep each batch maximally informative by admitting only prompts of intermediate pass rate, where the Bernoulli variance $p(1-p)$ peaks. We argue that this objective is incomplete, and give a stylised model of why. If training is idealised as *transport* of the policy's competence distribution along a one-dimensional difficulty axis, the direction of travel must be estimated from the model's own pass rates: a step of length ℓ then leaves the Wasserstein geodesic by an amount *linear in* ℓ, and when successive estimation errors are unbiased and uncorrelated, covering the same displacement in m re-anchored steps reduces the deviation to $O(\ell/\sqrt{m})$. The model motivates bounded, repeatedly re-anchored increments — Krashen's "$i{+}1$" taken in *small* steps — over greedy jumps to the currently most informative difficulty. The premise of the per-step view is nonetheless sound: under group-relative advantages, a prompt whose rollouts are all correct or all wrong contributes exactly zero gradient, and 61–62% of Hendrycks-MATH lies at one of these two extremes. We instantiate the idea as Ri1A and compare it with ODF, AdaCuRL, DAPO, and DAPO with static difficulty ordering (DAPO+static). At *equal wall-clock* (five hours on Qwen3-4B with identical data, batch size and response limit), Ri1A outperforms ODF and AdaCuRL by 2.9 and 2.0 pp on 502 pooled competition problems ($p < 10^{-3}$), at 2.3× and 1.7× lower cost per optimizer step. Against DAPO trained on the full dataset for its complete schedule, Ri1A reaches comparable accuracy at 1.7B and 4B in 43–46% of the wall-clock time, and is ahead at 8B (+1.44 pp on pooled problems, $p{=}.015$). With optimizer steps held equal, the smaller increment wins on all eight benchmarks at 1.7B. Code is available at `https://anonymous.4open.science/r/Ri1A-F62D`.

1 INTRODUCTION

Under GRPO (Shao et al., 2024) and its variants (Yu et al., 2025), now the main lever for mathematical reasoning (Guo et al., 2025; Kimi Team et al., 2025), each rollout's advantage is measured against the mean reward of its group. If all n rollouts for a prompt succeed, or all fail, every advantage vanishes and the prompt contributes nothing to the gradient, however much compute it consumed. This gives Krashen's *input hypothesis* (Krashen, 1982) a measurable form. Its standing objection is that neither the learner's state i nor the input $i{+}1$ was ever defined in testable terms (Gregg, 1984); in RLVR both are observable. Competence i is the pass rate, comprehensible input is the set of prompts with a non-degenerate pass rate, and $i{+}1$ is a bounded increment of the difficulty threshold. Figure 1 measures all three on Hendrycks-MATH (Hendrycks et al., 2021) with Qwen3-1.7B (Yang et al., 2025): the pass-rate histogram is strongly bimodal, 61–62% of the pool sits at a zero-gradient end, and training on the mastered end cuts the fraction of gradient-carrying groups to 0.19, against 0.72 for the middle band.

That the middle band is the right place to train is not new. ODF (Bae et al., 2026) filters each batch down to prompts of intermediate pass rate and reports large gains. Implicit in this recipe is an objective: *make every step carry as much gradient as possible*. Since the Bernoulli variance $p(1-p)$ peaks at $p = 1/2$, admitting the intermediate band is its greedy solution, and the same objective is implicit whenever a curriculum is judged by how informative its batches are.

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.

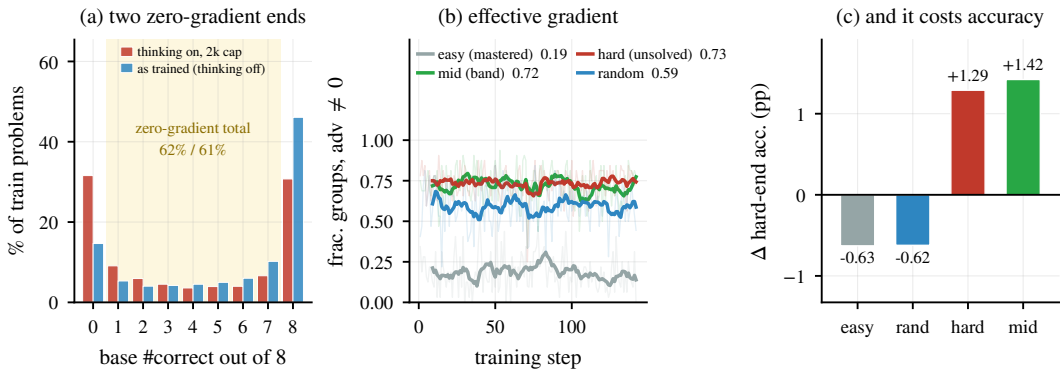


Figure 1: Zero-gradient waste is large, measurable and costly (Qwen3-1.7B, Hendrycks-MATH). **(a)** Pass-rate histogram under two scoring protocols, which agree on the *total* zero-gradient mass (62.3% vs 60.7%) but not on its split between the ends; we claim only the total. **(b)** Fraction of groups with non-zero advantage (Appendix C.4) for four runs ("hard" denotes the lowest-pass-rate band, not unsolvable problems) differing *only* in the band trained on: the mastered end collapses it to 0.19, the lowest-pass-rate band sustains 0.73. **(c)** Hard-benchmark accuracy relative to base at equal data and steps. The bands are defined and the arms detailed in Section 5.

Our thesis is that this objective is incomplete, and that optimising it can backfire. Maximising the information in each step is not the same as reaching the target. If training is viewed as *transport*, a curriculum chooses a path, and deviation accumulates along it unseen by any per-step criterion. Section 3 develops this in a stylised model. The direction of travel is not observable: it must be estimated from the model's own pass rates, with an angular error ε_t, so a step of length ℓ leaves the Wasserstein geodesic by $d_\perp = \ell \sin \varepsilon_t$, *linear in the step length*. Covering the same ground in m re-anchored steps refreshes the estimate each time, and when the refreshed errors are unbiased and uncorrelated the deviations partially cancel, accumulating only as $O(\ell \sin \varepsilon / \sqrt{m})$ (Corollary 3); a persistent bias, by contrast, is not reduced by subdivision. We test the prediction where it is sharpest: with optimizer steps held equal, the smaller increment should win, and at 1.7B it does (Table 19). In this reading Krashen's "+1" is a hedge against estimation error, not only a pedagogical intuition.

Ri1A (Section 4) implements this design. Problems are ranked once by base-policy pass rate; each stage admits the easiest $c\ell_k$ fraction of the remaining pool; the stage-$i{+}1$ threshold is re-anchored on the pass rate the *current* policy achieves on what stage i trained on; and problems that remain too hard are *deferred*, not discarded. Deferral matters because the lowest-pass-rate band still sustains a 0.73 non-zero-advantage rate (Figure 1b): a single $n{=}8$ scan cannot tell which problems a policy will never learn from (Appendix C.3), so a hard filter throws away live gradient.

Contributions. *(i)* A stylised transport model in which per-step informativeness and path deviation come apart, so that a criterion maximising the former need not control the latter (Section 3). Its sharpest prediction holds in the one controlled test available: with equal optimizer steps and the increment as the only difference, the *less* informative small increment wins on all eight benchmarks at 1.7B (Table 19). *(ii)* The zero-gradient ends are harmful, not merely wasteful (Section 5), and a filter that discards them outright loses on both hard and easy benchmarks (Section 8). *(iii)* With identical data and batch size and equal wall-clock, Ri1A beats both published curricula on competition mathematics at 4B; against plain DAPO it reaches comparable accuracy in under half the time, with a resolved gain at 8B (Tables 1 and 12). Table 7 lists each claim's evidence and gaps.

2 Related Work

Difficulty filtering in RLVR. ODF (Bae et al., 2026) admits only prompts of intermediate pass rate, refilling each batch asynchronously, and reports gains of up to +12% over GRPO with fewer than half the updates. Its Proposition 3.1 justifies the band through the Bernoulli variance $p(1-p)$, which is exactly the per-step objective we argue against (Appendix B.4). Two features distinguish its setting from ours: the difficulty signal is recomputed for every batch and never reused, and re-entry

is governed by visit count rather than progress, so the band never advances. DAPO (Yu et al., 2025), on which we build, drops the KL penalty and oversamples until every group carries a non-zero advantage. This within-batch remedy is orthogonal to ordering, as is reward reshaping: STRIDE (Zhao et al., 2026) sharpens credit assignment *within* a group but ignores degenerate groups.

Adaptive curricula with anti-forgetting machinery. AdaCuRL (Li et al., 2025a) identifies the same two failure modes, *gradient starvation* and *policy degradation*, and addresses them with coarse-to-fine difficulty estimation, adaptive scheduling, revisiting of historical data, and an adaptive reference model with sparse KL. Its remedy is heavier than ours: five generations per problem to bin the pool, then $N{=}100$ on the selected subset, against a single $n{=}8$ scan and no reference model.

Curricula for RLVR reasoning. E2H (Parashar et al., 2026) shifts sampling from easy to hard buckets on a fixed schedule, and AdaRFT (Shi et al., 2026) moves a target difficulty with the batch reward and samples the problems nearest to it; both read difficulty from labels or a reference model scored once, so the per-problem ordering does not follow the policy being trained. Cai et al. (2026) re-locate the model's capability boundary each round with up to 256 samples per problem and add teacher guidance there, and R^3 (Xi et al., 2024) runs a reverse curriculum over the steps of a demonstration. Ri1A re-estimates only a scalar threshold, from problems it has already rolled out, with no teacher and no step-level intervention; guidance-based methods are complementary and could serve its deferred tail.

Curriculum learning and data selection. Classical curricula order examples by a fixed or learned difficulty proxy, and automated variants track learning progress or set goals at the competence frontier. These are close in spirit to a moving band but lack a pass rate from which to read it off; a parallel line within RLVR shrinks the training set instead of reordering it (Appendix A). What makes $i{+}1$ operational here is that difficulty is a property of the current policy and is cheap to measure (Bae et al., 2026): it is a verifier pass rate. The construction is therefore not specific to mathematics: curriculum RLVR already spans multi-turn dialogue (Li et al., 2026), RLVR reaches layout-aware image generation (Liu et al., 2026), and reasoning-style training extends to tactile commonsense (Lyu et al., 2026), each of which supplies or could supply the probe μ_t; we test only text mathematics.

3 Why small steps: a geodesic view

A stylised model. We do not model RL updates directly. We summarise a curriculum's effect by how it moves the policy's competence across difficulty levels, and ask what this idealisation implies for step size. Fix an *exogenous* difficulty axis: each task q has an intrinsic difficulty $d(q) \in \mathbb{R}$ independent of the policy, with $P = \mathrm{law}(d(q))$ over the pool. A *probe* returns a competence signal $\mu_t(q) \in [0,1]$ that increases with mastery (pass rate in RL, inverse perplexity in SFT), and the *competence distribution* is the competence-weighted law of difficulty,

$$\widehat{Q}_t \;=\; \frac{1}{Z_t}\sum_i \mu_t(q_i)\,\delta_{d(q_i)}, \qquad Z_t \;=\; \sum_i \mu_t(q_i). \tag{1}$$

A uniformly competent policy (μ_t constant) has $\widehat{Q}_t = P$, so P is the natural target and $W_2(Q_t, P)$ measures how unevenly competence is spread over difficulty; Q_t denotes the population (large-pool) version of $\widehat{Q}_t$. Because the axis is one-dimensional, $Q \mapsto F_Q^{-1}$ embeds $(\mathcal{P}_2(\mathbb{R}), W_2)$ isometrically as a convex cone in $L^2(0,1)$. The geodesic from Q_0 to P is therefore unique, for atomic and continuous laws alike, and interpolates quantile functions,

$$F^{-1}_{Q_\tau} \;=\; (1-\tau)\,F^{-1}_{Q_0} + \tau\,F^{-1}_{P}, \qquad W_2(Q_\tau, P) \;=\; (1-\tau)\,W_2(Q_0, P). \tag{2}$$

The embedding is flat, so tangent directions at different states live in one Hilbert space and angles between them are well defined. We model a curriculum round as a step of length ℓ in a direction $\hat{v}_t$ and measure its quality by the angle to the geodesic direction $v^\star \propto F_P^{-1} - F_{Q_t}^{-1}$,

$$\varepsilon_t \;=\; \angle(\hat{v}_t,\, v^\star), \qquad \text{per-step progress along } v^\star \;=\; \ell\cos\varepsilon_t. \tag{3}$$

The angle and its amplification. $v^\star$ depends on the current competence profile and must be estimated from probe outcomes. An idealised curriculum would do so by Bayesian updating, with θ parametrising Q_t and y_t the probe outcomes:

$$p(\theta \mid \mathcal{D}_{t+1}) \;\propto\; p(y_{t+1} \mid \theta)\, p(\theta \mid \mathcal{D}_t), \qquad \hat{v}_t \;=\; v^\star\big(\mathbb{E}[\theta \mid \mathcal{D}_t]\big). \tag{4}$$

Proposition 1 (Posterior contraction sets the angular error)**.** *Suppose that, while a state θ_0 is probed, outcomes follow a correctly specified, regular and identifiable model $p(y \mid \theta)$, conditionally independent given θ_0, with accumulated Fisher information $\mathcal{I}_t \uparrow \infty$, and let $v^\star$ be C^1 with $L_v := \sup_\theta \|\nabla_\theta v^\star(\theta)\| < \infty$ near θ_0. Then $\mathrm{Var}(\theta \mid \mathcal{D}_t) \approx \mathcal{I}_t^{-1}$ and $\mathbb{E}\,\varepsilon_t \lesssim L_v\, \mathcal{I}_t^{-1/2}$.*

Proposition 2 (Error amplification is linear in step length)**.** *A step of length ℓ along $\hat{v}_t$ deviates from the geodesic direction by $d_\perp = \ell \sin \varepsilon_t \approx \ell\, \varepsilon_t$; at a given ε_t the only control on $d_\perp$ is ℓ.*

Proposition 1 describes an idealised estimator: in training the policy moves between probes, which are chosen adaptively, and Ri1A implements a one-dimensional special case (Section 4). We use it qualitatively — a refreshed estimate improves, a frozen one does not — not for a rate. Static labels hold a fixed angle $\alpha > 0$, leaving a residual $N\ell(1 - \cos\alpha)$ (Proposition 5; proofs in Appendix B).

Corollary 3 (Subdivision cancels unbiased error)**.** *Cover a displacement ℓ in m steps of length ℓ/m, and let e_j be the component of $\hat{v}_j$ orthogonal to the reference direction $v^\star$ of the segment from Q_0 to P. If (e_j) is a martingale-difference sequence with respect to the re-anchoring history, with $\mathbb{E}[\|e_j\|^2 \mid \mathcal{F}_{j-1}] \le \sin^2\varepsilon$, the accumulated orthogonal displacement $d_\perp^{(m)} = \frac{\ell}{m}\sum_j e_j$ satisfies*

$$\left(\mathbb{E}\,\|d_\perp^{(m)}\|^2\right)^{1/2} \;\le\; \ell \sin\varepsilon/\sqrt{m}. \tag{5}$$

If instead $\mathbb{E}[e_j \mid \mathcal{F}_{j-1}] = b \neq 0$, the bias contributes $\ell\|b\|$ whatever m.

Re-anchoring is meant to supply the martingale-difference condition; systematic misspecification, such as a base ranking that training re-orders, enters as the bias term, which subdivision does not remove.

The assumption behind the ranking. Proposition 2 bounds $d_\perp$ at a *given* ε_t, so on its own it does not rank a long step against a short one: a greedy filter admits the band with the most informative probes, and its longer step may buy a *smaller* ε_t. The ranking holds if $\partial \log \varepsilon_t / \partial \log \ell > -1$, i.e. if the angular error does not improve fast enough to pay for the longer step. We state this condition rather than derive it; a strong enough violation would reverse Table 19.

Corollary 4 (Curriculum gaps are proportional to distance travelled)**.** *After N steps the shortfall of a fixed-angle scheme and the accumulated off-geodesic deviation of a large-step scheme are, respectively,*

$$\underbrace{N\ell\,(1-\cos\alpha)}_{\textit{fixed angle } \alpha} \qquad \textit{and} \qquad \underbrace{\textstyle\sum_j \ell_j \sin\varepsilon_j}_{\textit{estimated direction}}, \tag{6}$$

both proportional to the total displacement $\sum_j \ell_j$ actually achieved. Hence the difference *between two curricula vanishes as the traversed distance vanishes.*

A rough empirical counterpart of distance travelled is the accuracy gained over base: plain DAPO moves Qwen3-1.7B by $+2.22$ pp on the pooled hard set but Qwen3-8B by $+5.13$ pp, and Ri1A's margin over DAPO grows across the same scales (Section 7.2). **Scope.** None of ε_t, $\mathcal{I}_t$ or $W_2(Q_t, P)$ is measured in this paper: pass rate, step count and the admitted fraction $c\ell_k$ stand in for competence, displacement and step length ℓ. The model is therefore a source of hypotheses rather than a description of RLVR dynamics. It commits to three orderings that could have come out otherwise: bounded increments should beat greedy jumps, smaller increments should beat larger ones, and all such differences should shrink on a model that barely moves. The main comparison is consistent with the first but does not isolate it (Section 8); the step-matched test bears on the second, varying $c\ell_k$ rather than ℓ; the scale trend is consistent with the third (Table 8).

4 Method: Ri1A

Ri1A is a stage-wise curriculum wrapped around unmodified DAPO. Let $\mathcal{D}$ denote the training pool and $p_\theta(x)$ the pass rate of policy θ on problem x over n rollouts. **Amortized difficulty.** Before training, every problem is scored once with the base policy θ_0 and the pool is sorted by p_{θ_0}, easiest first. This single pass is amortised over the run, whereas per-batch filtering pays again for every discarded prompt at every step. **Bounded admission (i+1).** Stage i trains on the easiest $K = \lfloor c\ell_k \cdot |\mathcal{D}| \rfloor$ problems of the remaining pool for `stage_epochs` epochs. The increment $c\ell_k$ is the

Algorithm 1 Ri1A.

Require: pool $\mathcal{D}$, base policy θ_0, increment $c\ell_k$, scale s, deferral cap r, epochs per stage E

Score every $x \in \mathcal{D}$ once with θ_0; $\mathcal{R} \leftarrow \mathcal{D}$ sorted by p_{θ_0}, easiest first
$K \leftarrow \lfloor c\ell_k \cdot |\mathcal{D}| \rfloor$; $\quad \theta \leftarrow \theta_0$; $\quad \tau_0 \leftarrow \infty$
for $i = 0, 1, 2, \ldots$ until $\mathcal{R} = \varnothing$ **do**
  $\mathcal{C}_i \leftarrow$ the K easiest problems of $\mathcal{R}$
  $\mathcal{H}_i \leftarrow \{x \in \mathcal{C}_i : \delta_{\theta_0}(x) > \tau_i\}$, hardest $r\,|\mathcal{C}_i|$ at most ▷ deferred
  $\mathcal{T}_i \leftarrow \mathcal{C}_i \setminus \mathcal{H}_i$
  $\theta \leftarrow \mathrm{DAPO}(\theta, \mathcal{T}_i, E \text{ epochs})$
  $\tau_{i+1} \leftarrow \bar{\delta}_i\,(1 + c\ell_k\, s)$, $\bar{\delta}_i$ the mean of $1 - p_\theta$ on $\mathcal{T}_i$ ▷ Eq. 7
  if $\mathcal{C}_i = \mathcal{R}$ **then**
    $\mathcal{R} \leftarrow \varnothing$ ▷ last stage: deferred problems are discarded
  else
    $\mathcal{R} \leftarrow \mathcal{R} \setminus \mathcal{T}_i$ ▷ $\mathcal{H}_i$ stays in the pool
  end if
end for

operational "+1": a fixed *fraction* of the pool rather than a jump to the estimated optimum, and hence the bounded step length of Proposition 2.

Re-anchoring. Write $\delta_\theta(x) = 1 - p_\theta(x)$ for difficulty. At the end of stage i, the *current* policy θ_i re-scores only the problems $\mathcal{T}_i$ that stage i trained on, and the deferral threshold for stage $i{+}1$ is anchored on that measurement:

$$\bar{\delta}_i \;=\; \frac{1}{|\mathcal{T}_i|} \sum_{x \in \mathcal{T}_i} \delta_{\theta_i}(x), \qquad \tau_{i+1} \;=\; \bar{\delta}_i\,(1 + c\ell_k \cdot s), \tag{7}$$

with $s = \texttt{drop_hard_scale}$. This is the one-dimensional special case of the estimation step in Eq. 4: under $p_\theta(x) = \sigma(a_\theta - b_x)$ training moves the ability a_θ but not the order of the b_x, so a scalar suffices and the base ranking of unseen problems stays valid. Re-scoring the unseen pool each stage would cost rollouts in $|\mathcal{R}|$; where training does re-order the pool, the base ranking behaves like a static label (Appendix B.3). **Deferral, not discard.** A candidate x is deferred when its base difficulty exceeds the threshold, $\delta_{\theta_0}(x) > \tau_i$, and returns to the pool for a later stage instead of being removed. At most a fraction $r = \texttt{drop_hard_max_ratio}$ of a stage's candidates is deferred, hardest first, so no stage can empty itself; stage 0 has no threshold, and $\tau_i \geq 1$ defers nothing. Deferral matters: the problems the base scan ranks hardest are *not* gradient-free (Figure 1b).

Algorithm 1 summarises the procedure. Ri1A has four settings — $c\ell_k$, s, r and the epochs per stage E — and needs no reference model, KL term or rehearsal buffer. All four are held at one value for every model (Section 6); only $c\ell_k$ is varied in this paper.

5 Measuring the problem

To show that the zero-gradient ends are harmful and not merely large, we rank Hendrycks-MATH by base pass rate, cut four 30% subsets of $2{,}250$ problems (**A-easy**, **A-mid**, **A-hard**, **A-random**), and train on each with plain DAPO for 142 steps on Qwen3-1.7B, with identical hyperparameters and seed so that only the band varies (Appendix C.2). Training on the easy band leaves the model *worse* on the hard benchmarks than the untrained base (HARD-AVG 0.2266 vs. 0.2329), and so does a random sample, effectively what plain DAPO trains on. The middle and hard bands instead gain $+1.42$ and $+1.29$ pp: budget spent on the zero-gradient ends costs accuracy, not just compute.

6 Experimental setup

Training. All runs use **DAPO** (Yu et al., 2025): group-relative advantages (Shao et al., 2024) with $\texttt{kl_loss_coef} = 0$ and no reference model, a single $7{,}500$-problem Hendrycks-MATH split (Hendrycks et al., 2021), lr $= 2 \times 10^{-6}$, $n = 8$ rollouts, batch size 16, a 2048-token response limit, bf16 and verl (Sheng et al., 2025). Ri1A adds $c\ell_k = 0.2$, $\texttt{drop_hard_scale} = 1$, $\texttt{drop_hard_max_ratio} = 0.5$ and $\texttt{stage_epochs} = 2$. The baselines are **DAPO** (random

Table 1: Main results: three models, five methods and the base model, mean@8; rows are methods, blocks are models. HARD-AVG, the macro-average over the four competition benchmarks, is the summary column; the two code benchmarks are out-of-domain probes and enter no claim. Within each block all runs share pool, batch size and response limit; the three curricula share a 5-hour cap on 2 H200s (binding at 4B, nominal at 1.7B), while DAPO and DAPO+static run their full two-epoch schedules at roughly twice the hours. Best per column and block in **bold**, second best underlined; Figure 2 marks which 4B margins resolve. The 1.7B Ri1A-vs-DAPO comparison reverses sign under the problem-level pooling of Section 7.2; at 4B and 8B the two views agree.

	in-domain, easier		**competition mathematics**				out-of-domain code		
	GSM8K	MATH-500	AMC23	AIME	Minerva	Omni	HumanEval	MBPP	**HARD-AVG**
Qwen3-1.7B									
base	0.8197	0.7093	0.4562	0.1042	0.2160	0.1550	0.5343	**0.5890**	0.2329
DAPO	**0.8424**	0.7418	0.4750	0.1139	0.2408	0.1825	0.5488	0.5841	0.2530
DAPO+static	0.8409	0.7415	0.4500	0.1181	**0.2413**	**0.1862**	0.5419	0.5846	0.2489
ODF	0.8304	0.7235	**0.4781**	0.1042	0.2284	0.1775	0.5450	0.5851	0.2471
AdaCuRL	0.8261	0.7308	0.4437	0.1069	0.2266	0.1825	0.5434	0.5803	0.2399
Ri1A	0.8400	**0.7442**	**0.4781**	**0.1347**	0.2325	0.1713	**0.5526**	0.5861	**0.2542**
Qwen3-4B									
base	0.9176	0.8235	0.6625	0.1806	0.2960	0.3025	0.6570	**0.7184**	0.3604
DAPO	0.9224	0.8430	0.7156	0.2236	0.3295	0.3375	0.6639	0.7087	0.4016
DAPO+static	**0.9241**	0.8448	0.7125	0.2056	**0.3323**	0.3400	0.6639	0.7169	0.3976
ODF	0.9211	0.8367	0.6719	0.1694	0.3176	0.3212	0.6562	0.7160	0.3700
AdaCuRL	0.9195	0.8303	0.6781	0.1972	0.3235	0.3225	0.6608	0.7116	0.3803
Ri1A	0.9219	**0.8515**	**0.7188**	**0.2361**	0.3318	**0.3513**	**0.6646**	0.7130	**0.4095**
Llama-3.2-3B									
base	0.7216	0.4358	0.2375	0.0569	0.1425	0.0537	0.4817	**0.5948**	0.1227
DAPO	0.7773	0.4670	0.2250	0.0792	0.1452	0.0737	0.4886	0.5895	0.1308
DAPO+static	0.7794	0.4780	0.2531	0.0792	0.1494	0.0575	**0.4893**	0.5924	0.1348
ODF	0.7695	0.4460	0.2687	0.0806	0.1438	**0.0825**	0.4787	0.5914	0.1439
AdaCuRL	0.7745	0.4662	0.2656	0.0778	0.1434	0.0688	0.4848	0.5914	0.1389
Ri1A	**0.7812**	**0.4835**	**0.2719**	**0.0917**	**0.1502**	0.0813	0.4878	0.5941	**0.1488**

order), **DAPO+static** (fixed buckets by dataset Level), **ODF** (Bae et al., 2026) and **AdaCuRL** (Li et al., 2025a), the last two at their recommended settings.

Compute budget. Unless stated otherwise, runs use 2 H200 GPUs (141 GB); the uncapped reruns of Table 4 use 4 A100 GPUs (80 GB). Ri1A, ODF and AdaCuRL share a 5-hour wall-clock cap at every model size, which binds at 4B and is nominal at 1.7B (Table 2); DAPO and DAPO+static run their full two-epoch schedules, roughly twice as long (Appendix C.7).

Evaluation. We report mean@8 (temperature 0.7, top-p 0.95, 8192 new tokens, 16384 for AIME, seed 42, vLLM (Kwon et al., 2023)) on GSM8K (Cobbe et al., 2021), MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024), AMC23, AIME23/24/25 (90 problems), Minerva (Lewkowycz et al., 2022) and Omni-MATH (Gao et al., 2025), plus HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021) as **out-of-domain probes** that enter no aggregate we make claims on. **Our headline metric is HARD-AVG**, the macro-average over the four competition benchmarks. Section 7.2 adds a **problem-level robustness check** pooling the same 502 problems; the two disagree in sign at 1.7B.

Statistics. We use a paired bootstrap over test problems (10,000 resamples) and the standard deviation across three training seeds ($43/53/33$; DAPO, DAPO+static and Ri1A on Qwen3-1.7B/4B, all methods on Llama-3.2-3B; Tables 13–14). Unless a caption says otherwise, tables show a separate seed-42 run held out of that triple (Appendix C.5).

7 RESULTS

7.1 MAIN RESULTS: QWEN3-1.7B, QWEN3-4B AND LLAMA-3.2-3B

Ri1A has the best HARD-AVG in every block. At 4B its margins over the published curricula are large: $+3.94$ and $+2.91$ pp over ODF and AdaCuRL, resolving individually on MATH-500 and

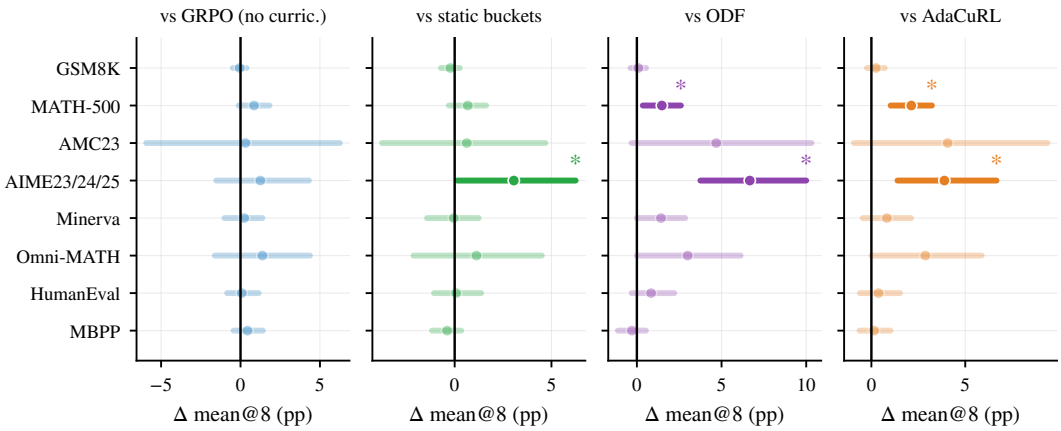


Figure 2: Qwen3-4B: Ri1A minus each baseline, per-problem paired bootstrap (10,000 resamples, 95% CI), stars marking intervals that exclude zero (GRPO and static in the panels denote DAPO and DAPO+static). Gains over ODF and AdaCuRL resolve on MATH-500 and AIME, and over DAPO+static on AIME; against plain DAPO none resolves.

Table 2: Wall-clock to the reported checkpoint on 2 H200 GPUs. The three curricula share a five-hour cap at every model size; the 4B row hits it, so it measures what fits in that budget rather than time-to-convergence. Fastest per row in bold.

	DAPO	DAPO+static	**Ri1A**	ODF	AdaCuRL
Qwen3-1.7B	7:43:38	7:44:40	3:31:12	**2:56:12**	4:08:59
Qwen3-4B	11:36:55	11:04:50	**5:01:21**	5:02:41	5:05:21

AIME ($p \leq .009$). The ordering holds at 1.7B ($+0.71$/$+1.43$ pp) and on Llama-3.2-3B (Grattafiori et al., 2024) ($+0.49$/$+0.99$ pp), with seed deviations below the Ri1A–ODF gap (Tables 13 and 14).

Against plain DAPO at its full schedule the picture depends on scale. HARD-AVG differs by $+0.12$, $+0.79$ and $+3.04$ pp at 1.7B, 4B and 8B, but only the 8B gap resolves on pooled problems (Section 7.2). At 1.7B and 4B, then, Ri1A matches DAPO in less than half the wall-clock rather than exceeding it.

7.2 A PROBLEM-LEVEL ROBUSTNESS CHECK

Because the competition benchmarks are small (AMC23 has 40 problems, AIME 90), we pool all four into one 502-problem set, weighting each *problem* equally where HARD-AVG weights each *benchmark* equally. This sharpens the picture: the gain over plain DAPO resolves at 8B ($+1.44$ pp, $p=.015$), is unresolved at 4B ($+0.65$ pp), and turns slightly negative at 1.7B (-0.27 pp), within seed variation. The trend is monotone in scale, consistent with Corollary 4 (Table 12).

7.3 WHERE THE COMPUTE GOES

Within the same five hours at 4B, Ri1A takes 846 optimizer steps against 234 for ODF and 262 for AdaCuRL: its steps are cheaper. A filter admitting a fraction ρ pays $1/\rho$ at *every* step, whereas Ri1A pays for one scan, so a Ri1A step costs $2.05\,Bn$ against $4.81\,Bn$ and $3.58\,Bn$ (Table 5).

Wall-clock tells the same story: Ri1A needs 46% of plain DAPO's time at 1.7B and 43% at 4B (Table 2). Run to self-termination, ODF spends most of its ≈69 h on selection (Table 4; Appendix C.7).

Table 3: Scale ablation: the same recipe at 8B and 9B, mean@8. The GLM-4-9B block runs a shorter baseline schedule, establishing architecture transfer rather than a step-matched comparison; no ODF or AdaCuRL run exists at either scale.

	in-domain, easier		**competition mathematics**				out-of-domain code		
	GSM8K	MATH-500	AMC23	AIME	Minerva	Omni	HumanEval	MBPP	**HARD-AVG**
Qwen3-8B									
base	0.9339	0.8227	0.6625	0.1972	0.2794	0.2963	**0.7477**	0.7753	0.3588
DAPO	0.9372	0.8612	0.7063	0.2375	0.3373	0.3425	0.7439	0.7763	0.4059
DAPO+static	0.9378	0.8582	0.7000	0.2306	**0.3483**	0.3375	0.7355	**0.7782**	0.4041
Ri1A	**0.9400**	**0.8660**	**0.7781**	**0.2639**	0.3355	**0.3675**	0.7454	0.7758	**0.4362**
GLM-4-9B-Chat									
base	0.8170	0.4758	0.5250	0.0236	0.1452	0.0450	0.6494	0.7223	0.1847
DAPO	0.8750	0.7780	0.6100	0.1720	0.2940	0.2900	0.7020	**0.7260**	0.3415
DAPO+static	0.8790	0.7840	0.6220	0.1810	0.2980	0.2970	0.7040	0.7220	0.3495
Ri1A	**0.8850**	**0.7920**	**0.6400**	**0.1940**	**0.3010**	**0.3050**	**0.7070**	0.7240	**0.3600**

Table 4: All three methods run to self-termination at 4B with no step limit, on the same 4 A100 GPUs at gbs 16 and response limit 2048. Each method stops by its own rule, so the table compares end points rather than equal compute. Neither baseline reaches the limit its own configuration computes (479 for ODF, 948 for AdaCuRL): AdaCuRL stops almost where the five-hour cap left it (265 vs 262), and ODF runs further (375 vs 234) but still halts on its own, so the short schedules are algorithmic rather than imposed by the cap. This hardware differs from Table 2 (Appendix C.7).

	stopped at	reason	runtime	HARD-AVG
ODF	375	no examples admitted	≈69 h	0.3841
AdaCuRL	265	buckets merged, pool spent	≈6.8 h	0.3901
Ri1A	**920**	stage schedule complete	**≈16.25 h**	**0.4290**

7.4 SCALE ABLATION: QWEN3-8B AND GLM-4-9B

Table 3 extends the recipe to 8B and to a third architecture. At 8B the gap over DAPO widens to $+3.04$ pp HARD-AVG and $+1.44$ pp pooled (p=.015), the only scale at which this comparison resolves. On GLM-4-9B (GLM Team et al., 2024) the ordering holds under a shorter baseline schedule (Appendix C).

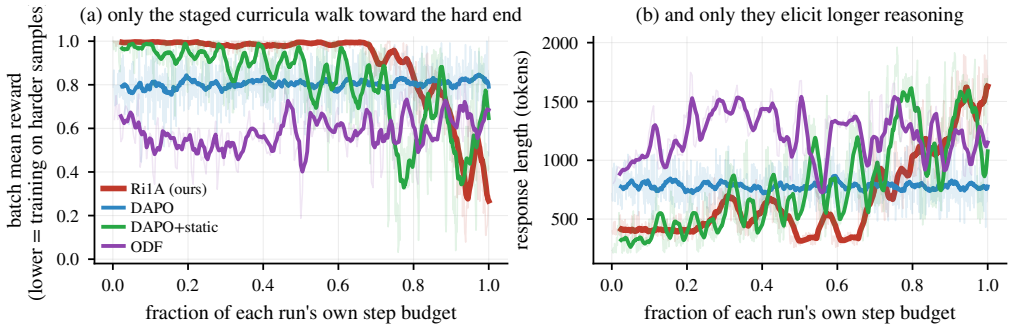


Figure 3: Qwen3-4B; the x-axis is each run's own step budget, so the panels are not compute-aligned. **(a)** Batch mean reward, lower being harder: only Ri1A descends smoothly ($0.99 \rightarrow 0.27$), DAPO stays flat, ODF is pinned inside its band, DAPO+static lurches at each bucket switch. **(b)** Only the staged curricula lengthen reasoning (Appendix C.11).

Table 5: Rollout complexity. The last column is the distinction that matters: ODF's overhead is a constant multiplier, Ri1A's an additive term that amortises as training lengthens. AdaCuRL's estimation term is its coarse pass (5 generations over the pool) plus its fine pass (N_{fine}=100 over the selected subset $\mathcal{S}$); derivation in Appendix C.6.

Method	Rollouts per step	Total rollouts	Overhead vs. no curriculum
DAPO, DAPO+static	Bn	$S\,Bn$	—
ODF	Bn/ρ	$S\,Bn/\rho$	$1/\rho = 4.8$ (measured), **multiplicative**
AdaCuRL	Bn/ρ + ref	$5N + N_{\text{fine}}\lvert\mathcal{S}\rvert + S\,Bn/\rho$ + ref	$1/\rho = 3.6$ (measured)
Ri1A	$Bn\left(1+\frac{1}{E_s}\right)+\frac{Nn}{S}$	$Nn + S\,Bn\left(1+\frac{1}{E_s}\right)$	**additive,** $\Theta(1/S)$

Table 6: **The same curriculum in SFT** (Qwen2.5-7B-Instruct, trained on GSM8K). An inverse-perplexity weight replaces the pass rate; single run, no intervals (Appendix C.10).

	sequences	mean@8		
		AMC23	MATH-500	Minerva
base (untrained)	—	0.4750	0.7300	0.2904
full-data SFT	13,452	0.5250	0.7320	0.2941
Ri1A-SFT	9,547	0.5500	0.7460	0.3100

8 Ablations

Increment size $c\ell_k$ is the ablation closest to Proposition 2. The sweep alone is confounded, since a smaller $c\ell_k$ also yields more steps, so we use a step-matched pair (1.7B, 706 steps): the smaller increment wins by $+1.27$ pp and leads on all eight benchmarks (Table 19; sweep in Appendix C.9).

Deferral vs. discard. A hard band filter that also discards mastered problems lowers HARD-AVG by 2.15 pp and costs 1.7 and 2.8 pp on GSM8K and MATH-500 (Table 21). Since 30.7% of the pool is mastered at base, discarding also halves training to 214 steps and shortens the learning-rate schedule, so this compares two recipes as wholes rather than isolating deferral. Likewise, the comparison with ODF and AdaCuRL measures practical utility; those methods differ in several mechanisms at once, and only the step-matched pair varies $c\ell_k$ alone.

Re-anchoring vs. a frozen anchor. Scoring the anchor with the base policy instead of the current one switches off the threshold update. At 1.7B the matched pair does not differ (pooled $+0.14$ pp, p=.56); at 4B and 8B re-anchoring leads by $+0.84$ and $+1.61$ pp on the hard benchmarks, without intervals (Table 22). Only the threshold is tested.

SFT. Table 6 runs the construction in SFT with an inverse-perplexity probe; objective and corpus change together, so it is not evidence of transfer (Appendix C.10).

9 Conclusion and Limitations

In a stylised transport model, keeping each step informative is not enough: the direction of travel must be estimated, and deviation from the geodesic scales with step length. The resulting method beats both published curricula at equal wall-clock and matches plain DAPO in under half the time, pulling ahead at 8B; measuring ε_t directly is the natural next step.

Table 7 maps every claim to its evidence. **Seed replication is partial**: only Tables 13 and 14 are three-seed; Qwen3 ODF/AdaCuRL, 8B/9B and all ablations are single-seed. **One regression reproduces**: Ri1A trails DAPO on Minerva and Omni-MATH at 1.7B, which Corollary 4 does not predict (Appendix C.12). **The effect of $c\ell_k$ is established only at 1.7B, and $c\ell_k$ is not a metric step length**; with ε_t, $\mathcal{I}_t$ and W_2 unmeasured, the transport model is a stylised account whose conditions we state but do not verify. **Two gaps are set by compute.** *Re-estimation is partial*: unseen problems keep their base ranking (Section 4), since re-scoring the pool each stage (60,000 rollouts early on) does not fit the five-hour cap; where training re-orders the pool, that ranking acts as a static label (Appendix B.3). *Scale is limited*: nothing beyond 8B–9B is verified.

Reproducibility statement

Every number comes from a stored evaluation artifact. The full training code — the curriculum implementation, the training-launch scripts for every run reported here, the difficulty-scoring script, the evaluation harness and the scripts that build every figure from those artifacts — is included in the supplementary material and at `https://anonymous.4open.science/r/Ri1A-F62D`. Section 6 and Appendix C give each run's configuration and which runs lack per-problem records; proofs are in Appendix B. Bootstrap intervals throughout reflect test-problem resampling only; training-seed variance is reported separately and never pooled with them, and the three tables that report seed spread rather than a bootstrap interval (Tables 13, 14 and 19) say so in their captions.

Ethics statement

This work fine-tunes open-weight language models on public mathematics benchmarks and introduces no new data collection, human subjects, or personally identifying information. The curriculum changes only the order in which existing problems are presented, so it inherits the risks of the underlying RLVR recipe rather than adding new ones.

AI use statement

We used generative AI tools as coding and writing assistants — refactoring experiment-launching and figure-generating scripts, and copy-editing prose we drafted — but not to generate research ideas, to produce or impute any experimental number, or to write the proofs. All AI-assisted code was reviewed and its outputs re-derived from stored artifacts by the authors, who take responsibility for the final content.

References

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models, 2021. URL `https://arxiv.org/abs/2108.07732`.

Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, JeongYeon Nam, and Donghyun Kwak. Online difficulty filtering for reasoning oriented reinforcement learning. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 700–719, 2026. URL `https://arxiv.org/abs/2504.03380`.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 41–48, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553380. URL `https://doi.org/10.1145/1553374.1553380`.

Pengxiang Cai, Tianchen Fang, Xiaohan Li, Qingyuan Zeng, Guocong Li, and Jintai Chen. Curriculum reinforcement learning can incentivize reasoning capacity in llms beyond the base model, 2026. URL `https://arxiv.org/abs/2606.22317`.

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`.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents. In *Proceedings of the 35th International Conference on Machine Learning (ICML)*, 2018. URL https://arxiv.org/abs/1705.06366.

Bofei Gao, Feifan Song, Zhe Yang, Zefan Cai, Yibo Miao, Qingxiu Dong, Lei Li, Chenghao Ma, Liang Chen, Runxin Xu, Zhengyang Tang, Benyou Wang, Daoguang Zan, Shanghaoran Quan, Ge Zhang, Lei Sha, Yichang Zhang, Xuancheng Ren, Tianyu Liu, and Baobao Chang. Omni-MATH: A universal olympiad level mathematic benchmark for large language models. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025. URL https://arxiv.org/abs/2410.07985.

GLM Team, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, et al. ChatGLM: A family of large language models from GLM-130B to GLM-4 All Tools. *arXiv preprint arXiv:2406.12793*, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Alex Graves, Marc G. Bellemare, Jacob Menick, Remi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *Proceedings of the 34th International Conference on Machine Learning (ICML)*, 2017. URL https://arxiv.org/abs/1704.03003.

Kevin R. Gregg. Krashen's monitor and Occam's razor. *Applied Linguistics*, 5(2):79–100, 1984.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z.

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 *Advances in Neural Information Processing Systems (NeurIPS), Datasets and Benchmarks Track*, 2021. URL https://arxiv.org/abs/2103.03874.

Kimi Team, Angang Du, Bofei Gao, Bowei Xing, Changjiu Jiang, Cheng Chen, Cheng Li, Chenjun Xiao, Chenzhuang Du, Chonghua Liao, et al. Kimi k1.5: Scaling reinforcement learning with LLMs. *arXiv preprint arXiv:2501.12599*, 2025.

Stephen Krashen. *Principles and Practice in Second Language Acquisition*. Pergamon Press, Oxford, 1982. URL https://api.semanticscholar.org/CorpusID:62756438.

M. Pawan Kumar, Benjamin Packer, and Daphne Koller. Self-paced learning for latent variable models. In J. Lafferty, C. Williams, J. Shawe-Taylor, R. Zemel, and A. Culotta (eds.), *Advances in Neural Information Processing Systems*, volume 23. Curran Associates, Inc., 2010. URL https://proceedings.neurips.cc/paper_files/paper/2010/file/e57c6b956a6521b28495f2886ca0977a-Paper.pdf.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, 2023.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. URL https://arxiv.org/abs/2206.14858.

Ming Li, Pei Chen, Zhenhao Zhang, Tao Yang, Xinyang Zhang, Han Li, Tianyu Cao, Ming Zeng, Zhuofeng Wu, Meng Jiang, Huasheng Li, Lihong Li, and Bing Yin. Mitigating lost in multi-turn conversation via curriculum RL with verifiable accuracy and abstention rewards. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 33364–33381, San Diego, California, United States, July 2026. Association for Computational

Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1540. URL https://aclanthology.org/2026.acl-long.1540/.

Renda Li, Hailang Huang, Fei Wei, Feng Xiong, Yong Wang, and Xiangxiang Chu. AdaCuRL: Adaptive curriculum reinforcement learning with invalid sample mitigation and historical revisiting, 2025a. URL https://arxiv.org/abs/2511.09478.

Xuefeng Li, Haoyang Zou, and Pengfei Liu. LIMR: Less is more for RL scaling. *arXiv preprint arXiv:2502.11886*, 2025b.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://arxiv.org/abs/2305.20050.

Junhao Liu, Jian-Wei Zhang, Tao Huang, Miles Yang, Zhao Zhong, and Liefeng Bo. Think, plan, paint: Layout-aware reasoning for controllable image generation in unified models, 2026. URL https://arxiv.org/abs/2607.16409.

Kailin Lyu, Di Wu, Pengwei Zhang, Yuhang Zheng, Yingxin Lai, Long Xiao, Kangyi Wu, Pengna Li, Chen Gao, Lianyu Hu, Xiaobin Hu, Jie Hao, Ce Hao, Weihao Yuan, and Shuicheng Yan. TouchThinker: Scaling tactile commonsense reasoning to the open world with large-scale data and action-aware representation, 2026. URL https://arxiv.org/abs/2606.11637.

Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher–student curriculum learning. *IEEE Transactions on Neural Networks and Learning Systems*, 31(9):3732–3740, 2020. doi: 10.1109/TNNLS.2019.2934906.

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 *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 20275–20321, 2025. URL https://arxiv.org/abs/2501.19393.

Shubham Parashar, Shurui Gui, Xiner Li, Hongyi Ling, Sushil Vemuri, Blake Olson, Eric Li, Yu Zhang, James Caverlee, Dileep Kalathil, and Shuiwang Ji. Curriculum reinforcement learning from easy to hard tasks improves llm reasoning. 2026. URL https://arxiv.org/abs/2506.06632.

Emmanouil Antonios Platanios, Otilia Stretcu, Graham Neubig, Barnabas Poczos, and Tom Mitchell. Competence-based curriculum learning for neural machine translation. In Jill Burstein, Christy Doran, and Thamar Solorio (eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 1162–1172, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1119. URL https://aclanthology.org/N19-1119/.

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.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems (EuroSys)*, 2025. arXiv:2409.19256.

Taiwei Shi, Yiyang Wu, Linxin Song, Tianyi Zhou, and Jieyu Zhao. Efficient reinforcement finetuning via adaptive curriculum learning. 2026. URL https://arxiv.org/abs/2504.05520.

Georgios Tzannetos, Bárbara Gomes Ribeiro, Parameswaran Kamalaruban, and Adish Singla. Proximal curriculum for reinforcement learning agents. *Transactions on Machine Learning Research*, 2023. URL https://arxiv.org/abs/2304.12877.

L. S. Vygotsky. *Mind in Society: Development of Higher Psychological Processes*. Harvard University Press, 1978. ISBN 9780674576285. URL http://www.jstor.org/stable/j.ctvjf9vz4. Edited by M. Cole, V. John-Steiner, S. Scribner, and E. Souberman.

Zhiheng Xi, Wenxiang Chen, Boyang Hong, Senjie Jin, Rui Zheng, Wei He, Yiwen Ding, Shichun Liu, Xin Guo, Junzhe Wang, Honglin Guo, Wei Shen, Xiaoran Fan, Yuhao Zhou, Shihan Dou, Xiao Wang, Xinbo Zhang, Peng Sun, Tao Gui, Qi Zhang, and Xuanjing Huang. Training large language models for reasoning through reverse curriculum reinforcement learning. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan 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, Mu Qiao, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. URL https://arxiv.org/abs/2503.14476.

Qinjian Zhao, Zhihao Dou, Dinggen Zhang, Xiangyu Li, Chaoda Song, Zhongwei Wan, Xinpeng Li, Yanyan Zhang, Kaijie Chen, Qingtao Pan, Chengcheng Feng, Zhiqiang Gao, and Xiaoyu Xia. Stride: Strategic trajectory reasoning via discriminative estimation for verifiable reinforcement learning, 2026. URL https://arxiv.org/abs/2606.15866.

A Antecedents of the $i{+}1$ construction, and related curricula

The pedagogical antecedents. Krashen's input hypothesis (Krashen, 1982) is the source of the $i{+}1$ framing used in the introduction; Vygotsky's zone of proximal development (Vygotsky, 1978) advances a similar claim, and both have been operationalised before — for RL agents as a proximal curriculum (Tzannetos et al., 2023), and for language modelling as competence-based scheduling (Platanios et al., 2019). The standing objection is Gregg's (Gregg, 1984): neither i nor $i{+}1$ was ever defined in measurable terms, which is precisely what a pass rate supplies.

Curriculum learning and data selection, in full. Self-paced learning (Kumar et al., 2010) and curriculum learning (Bengio et al., 2009) order examples by a fixed difficulty proxy; teacher–student curricula (Matiisen et al., 2020) learn the proxy instead; and automated curricula follow learning progress (Graves et al., 2017) or generate goals at the competence frontier (Florensa et al., 2018). The last of these is closest in spirit to a moving band, but none reads the band off a quantity as directly measurable as a group pass rate. Within RLVR, staged introduction of harder problems appears in Kimi k1.5 (Kimi Team et al., 2025), while a parallel line reduces the training set rather than reordering it (Li et al., 2025b; Muennighoff et al., 2025) — selecting *once* against a fixed proxy, so the selection cannot track a policy that improves under it.

B Proofs, alternative schemes, and what each claim rests on

B.1 Proof of Proposition 1 (posterior contraction)

Proof of Proposition 1. Two steps: Bernstein–von Mises controls the parameter error, the delta method transports it to the angle.

Step 1 (parameter error). Under the stated conditions (a correctly specified, regular and identifiable model, outcomes conditionally independent given a fixed θ_0, and $\mathcal{I}_t \uparrow \infty$), Bernstein–von Mises gives

$$\theta \mid \mathcal{D}_t \;\approx\; \mathcal{N}\big(\hat{\theta}_t,\, \mathcal{I}_t^{-1}\big), \tag{8}$$

so by Jensen's inequality

$$\mathbb{E}\|\hat{\theta}_t - \theta_0\| \;\le\; \big(\mathbb{E}\|\hat{\theta}_t - \theta_0\|^2\big)^{1/2} \tag{9}$$

$$\approx\; \big(\operatorname{tr}\mathcal{I}_t^{-1}\big)^{1/2} \;=:\; \mathcal{I}_t^{-1/2}, \tag{10}$$

where throughout we write the scalar $\mathcal{I}_t^{-1/2}$ for $(\operatorname{tr}\mathcal{I}_t^{-1})^{1/2}$. This is the quantity the Jensen bound produces; $\lambda_{\max}(\mathcal{I}_t^{-1})^{1/2}$, the worst-direction scale, is a different quantity, and the two agree only up to the dimension factor $\lambda_{\max} \le \operatorname{tr} \le d\,\lambda_{\max}$. Since we only ever use the bound up to constants in d, either choice supports the $\mathcal{I}_t^{-1/2}$ rate; we fix the trace version for definiteness.

Step 2 (angle). Since $v^\star \in C^1$, a first-order expansion gives

$$v^\star(\hat{\theta}_t) - v^\star(\theta_0) \;=\; \nabla_\theta v^\star(\theta_0)\,(\hat{\theta}_t - \theta_0) + o\big(\|\hat{\theta}_t - \theta_0\|\big), \tag{11}$$

and for unit vectors $\|u - w\| = 2\sin\frac{1}{2}\angle(u,w)$, so that Jordan's inequality gives $\angle(u,w) \le \frac{\pi}{2}\|u - w\|$ and hence

$$\varepsilon_t \;\lesssim\; L_v\|\hat{\theta}_t - \theta_0\|. \tag{12}$$

Taking expectations and inserting Step 1 yields the claim. □

B.2 Proof of Proposition 2 (linear amplification)

Figure 4 is the picture the proposition formalises: one angular error, two step lengths.

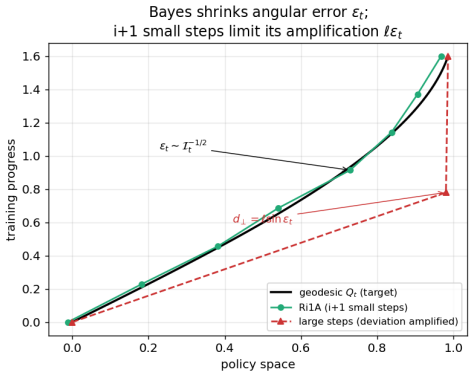


Figure 4: One angular error, two step lengths. Since $d_\perp = \ell \sin\varepsilon_t$ is linear in ℓ, a large step at the *same* angle is thrown far off the geodesic; $\varepsilon_t > 0$ being unavoidable, only bounded steps keep the trajectory on it.

Proof of Proposition 2. Let $v^\star$ be the unit tangent to the geodesic at Q_t and $\hat{v}_t$ the unit estimate at angle ε_t. A step $\ell\hat{v}_t$ decomposes as

$$\ell\,\hat{v}_t \;=\; \underbrace{\ell\cos\varepsilon_t}_{\text{along } v^\star}\,v^\star \;+\; \underbrace{\ell\sin\varepsilon_t}_{=\,d_\perp}\,v^\perp, \tag{13}$$

with $v^\perp$ the unit vector orthogonal to $v^\star$ in the plane they span, so $d_\perp$ is proportional to ℓ at fixed ε_t. All vectors lie in $L^2(0,1)$ under the quantile embedding of Section 3, so the decomposition needs no parallel transport. □

Proof of Corollary 3. In the flat embedding the reference segment from Q_0 to P has a fixed direction $v^\star$, and the orthogonal displacement after m steps is $d_\perp^{(m)} = \frac{\ell}{m}\sum_{j\le m} e_j$. For a martingale-difference sequence the cross terms vanish, $\mathbb{E}\langle e_j, e_k\rangle = \mathbb{E}\langle e_j, \mathbb{E}[e_k \mid \mathcal{F}_{k-1}]\rangle = 0$ for $j < k$, so

$$\mathbb{E}\,\|d_\perp^{(m)}\|^2 \;=\; \frac{\ell^2}{m^2}\sum_{j\le m}\mathbb{E}\,\|e_j\|^2 \;\le\; \frac{\ell^2\sin^2\varepsilon}{m}. \tag{14}$$

With a conditional mean b, write $e_j = b + \xi_j$ with (ξ_j) a martingale difference; the mean part sums to ℓb exactly, and only the ξ_j part is reduced by $1/\sqrt{m}$. Steps are assumed small enough to stay inside the cone of quantile functions. □

B.3 THE TWO ALTERNATIVES, AS PROPOSITIONS

Ri1A is one of three one-line design choices, and the other two can be described in the same terms. The three differ only in how the admission threshold for the next round is obtained, and Ri1A's re-anchoring step (Algorithm 1, line 8) is the only place the probe information $\mathcal{I}_i$ accumulates, so an arm that skips it cannot shrink ε_i however many stages it runs. In Ri1A that information enters only through the scalar threshold, and it comes from problems the policy has just trained on, whose pass rates approach 1 and whose per-probe information about the ability therefore shrinks (Eq. 20). The contraction Ri1A actually realises is thus at most that of a one-dimensional ability estimate.

Why the unseen pool is not re-scored. The update of Eq. 4 would re-estimate the current policy's difficulty on the *unseen* problems as well. Doing so every stage costs rollouts in proportion to $|\mathcal{R}|$ rather than $|\mathcal{T}_i|$, which in early stages is the whole pool, so Ri1A re-uses the base-policy ranking of $\mathcal{R}$ and re-estimates only the threshold from problems it has already paid to roll out. Under a one-dimensional model $p_\theta(x) = \sigma(a_\theta - b_x)$ this loses nothing: p_θ is monotone in $a_\theta - b_x$, so training changes every problem's pass rate but not their order, and updating the scalar threshold is equivalent to updating a_θ. Where difficulty is not one-dimensional and training re-orders the pool, the ranking of $\mathcal{R}$ inherits the fixed angle of static labels (Proposition 5) and only the step length is re-estimated. The frozen-anchor ablation (Table 22) keeps the base ranking in both arms, so it tests the threshold and not a re-ranking of the pool.

Proposition 5 (Static labels: a fixed angle)**.** *Human difficulty labels encode a reference family's ordering, giving an angle $\alpha = \angle(v_{\mathrm{hum}}, v^\star) > 0$ that never shrinks, because the labels are never re-estimated against θ_t. Per-step progress is $\ell \cos\alpha < \ell$: reaching P needs path length $L/\cos\alpha > L$, or within N steps leaves residual $W_2(Q_N, P) \geq N\ell(1 - \cos\alpha) > 0$.*

Proof. Project each step onto $v^\star$. With α fixed,

$$\text{per-step gain} \;=\; \ell\cos\alpha \;<\; \ell, \tag{15}$$

$$\text{path length to cover } L \;=\; L/\cos\alpha \;>\; L, \tag{16}$$

$$\text{residual after } N \text{ steps} \;=\; N\ell\,(1-\cos\alpha) \;>\; 0. \tag{17}$$

Because the labels are never re-estimated, α and the target offset stay fixed and the residual persists as $N \to \infty$. □

Proposition 6 (Random ordering: $\sqrt{N}$ drift)**.** *With isotropic step directions the net displacement along $v^\star$ after N steps is $O(\sqrt{N}\ell)$ rather than $N\ell$, so reaching P takes $\sim (L/\ell)^2$ steps.*

Proof. With isotropic directions $\mathbb{E}\cos\varepsilon = 0$ (or $O(1/\sqrt{d})$ under weak bias), so the along-$v^\star$ displacement after N steps is a random walk:

$$\Big\langle \textstyle\sum_{j\leq N} \ell\,\hat{v}_j,\; v^\star \Big\rangle \;=\; O\big(\sqrt{N}\,\ell\big) \;\neq\; N\ell, \tag{18}$$

$$\text{so } \sqrt{N}\,\ell \sim L \;\Longrightarrow\; N \;\sim\; (L/\ell)^2. \tag{19}$$

□

The isotropy assumption is too strong, and we do not rely on this proposition. A random *ordering* over a fixed pool is not a random *direction* in ability space: training on any sample drawn from the pool still raises competence on that pool, so $\mathbb{E}\cos\varepsilon > 0$ strictly, and the displacement is a biased walk with a linear drift term rather than the pure $\sqrt{N}$ walk above. Proposition 6 should therefore be read as the limiting case in which the curriculum signal carries no information about $v^\star$, not as a model of the DAPO baseline — which is precisely why we do *not* cite it when interpreting the random-order results, and why Table 8 lists no experiment against it. The measured behaviour of random ordering is the A-random band of Table 15, which loses to the mid and hard bands by ≈ 2 pp but is nowhere near the quadratic slowdown this proposition would imply.

B.4 WHY GREEDY FILTERING IS THE SINGLE-STEP OPTIMUM, AND WHY THAT IS NOT ENOUGH

Greedy filtering is the single-step optimum. With n rollouts per problem the rewards are $\mathrm{Bernoulli}(p)$, and two facts should be kept apart. The first is exact: under group-relative advantages a group contributes zero gradient precisely when all n rewards agree, so the probability that a group is live is

$$\Pr[\text{live}] \;=\; 1 - p^n - (1-p)^n, \qquad I_a(p) \;=\; n\,p(1-p) \;\text{ for } p = \sigma(a-b). \tag{20}$$

The second depends on the parametrisation. Under the one-dimensional model of Section 4, the Fisher information that n rollouts carry about the ability a is $I_a(p)$ above; in the probability parametrisation it would be $n/(p(1-p))$, which does not vanish. We state the information claim for the ability parametrisation, the quantity Ri1A's scalar re-anchoring estimates. There both $\Pr[\text{live}]$ and I_a vanish at $p \in \{0, 1\}$ and peak at $p = 1/2$, so the gradient bottleneck and the direction blind spot of Proposition 1 occur together, though they are different quantities. Admitting the band that maximises either is the single-step optimum ODF targets (Bae et al., 2026). It is myopic in the transport model: jumping to the currently most informative difficulty takes the largest admissible ℓ and so, by Proposition 2, amplifies whatever ε_t remains. ODF optimises the step; the transport model asks about the path.

B.5 THE $c\ell_k \mapsto \ell$ PROXY, AND WHY WE DO NOT REPAIR IT

Proposition 2 bounds the off-geodesic deviation of a step of *metric* length ℓ; Ri1A controls the *quantile* width $c\ell_k$ of the admitted slice. Writing F for the c.d.f. of the difficulty law over the pool, a stage that advances the admission threshold from quantile u to $u + c\ell_k$ moves it in difficulty units by

$$\Delta d \;=\; F^{-1}(u + c\ell_k) - F^{-1}(u) \;\approx\; \frac{c\ell_k}{f\big(F^{-1}(u)\big)}, \tag{21}$$

with f the difficulty density. The induced W_2 displacement of Q_t is therefore proportional to $c\ell_k$ only where f is locally flat. Figure 1a shows f is bimodal, with most mass at the two zero-gradient ends and a thin middle, so $1/f$ varies by a large factor across the sweep's range: the same $c\ell_k$ buys a small ℓ inside the mastered mode and a large one crossing the sparse middle. Two consequences follow. First, the mapping $c\ell_k \mapsto \ell$ need not be monotone in the way Proposition 2 requires, so a non-monotone HARD-AVG in $c\ell_k$ is not by itself evidence against the proposition — it is equally evidence that the proxy is mis-specified. This is a testable statement rather than a disclaimer, and Figure 7b tests it on the 1.7B sweep by computing $\max_i \Delta d_i$ per run from the base-policy difficulty c.d.f.: the corrected axis preserves the $c\ell_k$ ordering up to one tie ($c\ell_k$=0.20 and 0.30 share the same $\max_i \Delta d_i$; Spearman $\rho = -0.97$). On that sweep, then, the base-policy proxy for Δd does not change the ordering, though it measures neither ℓ nor the exogenous difficulty law P itself. We have not run the same correction at 4B, so we do not know whether it would account for the disagreement there, and we do not appeal to it as though it would. Second, a principled version of Ri1A would admit a slice of fixed Δd (or fixed estimated W_2 step) rather than fixed quantile width, which is a different and untested algorithm. We keep the quantile form because it is what we ran, and record the gap here rather than presenting $c\ell_k$ as though it were ℓ.

B.6 STATUS OF THE GEOMETRIC ARGUMENT

Section 3 summarises the standing of the model; we expand it here. Three proxies substitute for unmeasured quantities: the k-rollout pass rate for competence μ_t, optimizer step counts for displacement, and the admitted fraction $c\ell_k$ for step length ℓ. The last is the weakest joint, since $c\ell_k$ and ℓ are proportional only where difficulty mass is uniform — which Figure 1a shows it is not, and which Appendix B.5 treats in full.

A fourth gap is not a proxy but an assumption: Proposition 2 compares steps at a *fixed* angular error, while a greedy filter may buy a smaller ε_t with its longer step, so the ranking of long against short rests on the condition $\partial \log \varepsilon_t / \partial \log \ell > -1$ of Section 3, which we state and do not prove. Since ε_t is never measured, the transport account is a stylised model that makes falsifiable predictions, not a theory of RLVR training dynamics.

What the geometry is, and is not, doing. We do not claim it as a verified account of RLVR training. Its role is narrower: it (i) motivates bounded increments rather than assuming them, (ii) says *in advance* when the effect must vanish (Corollary 4), and (iii) predicts what a per-step criterion denies — the less informative increment winning at equal steps. A reader who rejects the picture entirely is left with (iii) as a bare empirical fact, and Table 19 is the same result either way; what the geometry adds is a reason to have expected it.

Two roles, kept separate. The Ri1A-versus-published comparisons and the step-matched test do different jobs. The former ask whether a curriculum built on the transport picture is *competitive* against curricula built on the per-step one; the answer is yes at every scale where they were run, but they cannot attribute the win to step length, since Ri1A and ODF differ in at least four ways at once (amortised versus per-batch scoring, deferral versus discarding, a band that advances versus one that does not, 846 versus 234 steps). The latter asks whether the *mechanism* is real and varies one thing only. Only the second is evidence for the geometry; the first is evidence that it led somewhere useful.

The argument earns its place by committing in advance to three orderings that could have come out otherwise. *Bounded increments should beat greedy jumps*: consistent with Ri1A leading both published curricula at every scale where they were run, though for the reason just given those runs do not isolate it. *Smaller increments should beat larger*: holding optimizer steps fixed and varying only the increment isolates $c\ell_k$, which stands in for ℓ through the proxy above, and the prediction holds at 1.7B by $+1.27$ pp with non-overlapping two-seed ranges (Table 19) — the one prediction of the theory we test cleanly, at one scale, with $n{=}2$ seeds and a batch size used nowhere else in the paper. *All curriculum differences should shrink on a model that barely moves*: the 1.7B non-differences of Table 12 are consistent with Corollary 4, though three scales cannot establish the relation. Table 8 pairs each proposition with the experiment bearing on it.

B.7 What each claim rests on

Table 7 inventories the *empirical* claims and the strength of the evidence behind each; Table 8 does the same for the *theoretical* ones, pairing every proposition with the experiment that bears on it. Read together, they are the paper's own account of what is resolved, what is directional, and what is not established.

C Additional results and experimental detail

C.1 Configurations, batch semantics and step counts

Baseline reimplementation. Neither ODF nor AdaCuRL is run at its published configuration (ODF: Qwen2.5-3B with an SFT cold start; AdaCuRL: Qwen2.5-Math-1.5B/7B), because a matched comparison requires a shared substrate; every number we report for them is a margin over our implementation on our pool and base models, not a reproduction of their published result. Our ODF run otherwise follows their Algorithm 1, including the re-queueing of the pool by visit count.

Batch semantics. In verl terms, `data.train_batch_size` is the number of prompts per rollout batch, `actor.ppo_mini_batch_size` the number of prompts per optimizer update, and `actor.ppo_micro_batch_size_per_gpu` the gradient-accumulation granularity. We set mini $=$ gbs throughout, giving strictly on-policy updates with no inner PPO epochs; the micro batch is a memory setting only and does not change the optimisation.

Configurations and step counts. Table 9 lists the hyperparameters shared by every RL run, and Table 10 the settings on which each experiment deviates from them, with its step counts. Two rows need a word. The step-matched pair of Table 19 runs at gbs 8 on a single H200 GPU, so that 706 steps consume half the prompts 706 steps consume elsewhere; it is internally controlled — same batch, hardware and step count, two seeds, differing only in $c\ell_k$ — and we use it only for the difference between its arms, never alongside a number from another table. The sweep round's $c\ell_k{=}0.2$ run at 1.7B is a separate launch from the main-table run and ran 846 steps rather than 428, which is why Table 17 and Table 1 report different numbers for the same nominal setting.

The four band arms of Section 5 are a separate, self-contained study: they run at gbs 32 rather than 16, on 2,250-problem subsets, so their absolute accuracies are not comparable with the main

Table 7: Empirical claims and the strength of the evidence behind each. *Resolved* means a confidence interval or a multi-seed separation excludes zero.

Claim	Evidence	Status
Zero-gradient mass is large and costly	Pool measurement; band study at equal data and steps	**Resolved**; two scoring protocols agree on the total
Beats matched ODF / AdaCuRL on hard mathematics	Qwen3-4B, five methods on identical data and budget; per-problem paired bootstrap	**Resolved** ($p < 10^{-3}$ pooled); over static bucketing positive but unresolved ($+0.80$ pp, p=.19)
Matches plain DAPO at lower per-step cost	846 vs. 938 steps; measured admission rates; matched wall-clock	Steps and wall-clock exact; cost model **derived, not measured**
Beats plain DAPO	HARD-AVG and pooled problems, three scales	HARD-AVG **positive at all three scales** ($+0.12/+0.79/+3.04$ pp); **resolved only at 8B** on pooled problems ($+1.44$ pp, p=.015), no difference at 1.7B, where the two metrics disagree in sign
Gain grows with model scale	Three scales, pooled	**Trend**, no CI on the relation
Smaller increments beat larger ones	Step-matched endpoint test at 1.7B (2 seeds, equal 706 steps); $c\ell_k$ sweep, five points at 1.7B, four at 4B	**Supported at 1.7B**: $+1.27$ pp for $c\ell_k$=0.10 over 0.50 at equal steps, all eight benchmarks, two-seed ranges non-overlapping on the four competition benchmarks; n=2 seeds, so no CI. **Untested at 4B**, where the unmatched sweep is non-monotone
Ri1A beats a discard-both-ends band filter	Band-filter comparison at 1.7B	**Supported at 1.7B** as a recipe comparison: $+2.15$ pp HARD-AVG, $+1.7/+2.8$ pp on GSM8K/MATH-500; steps and schedule differ, so deferral itself is not isolated; single seed
The deferral cap is set well	—	**Untested**: never swept; all runs use 0.5. Two earlier cap sweeps withdrawn (they varied a different hyperparameter)
Re-anchoring beats a frozen anchor	Frozen-anchor runs at three scales	**Not established**: no difference at 1.7B, no CIs at 4B/8B
Ordering effect transfers off Hendrycks-MATH	SFT on GSM8K, competence from inverse perplexity, three harder OOD benchmarks	**Not established**: objective and corpus vary together, so the gain is not attributable to ordering; single run, no intervals

tables and we use them only for the difference between bands. The batch size matters for reading Figure 1b, whose per-step values average 32 indicators (Appendix C.4).

Two further conventions. The no-curriculum baselines are always compared at their full two-epoch checkpoints (938/942 steps), never mid-run. Wall-clock figures (Table 2) are 2-H200 totals to the checkpoint each table reports, timed end-to-end including the one-off difficulty scan; the three curricula share a five-hour cap at every model size for fairness, not only at 4B. The 4B row hits the cap by construction and is therefore a fixed-budget comparison rather than a time-to-convergence measurement, while at 1.7B all three curricula converge before the cap binds.

Table 8: Each theoretical claim and the experiment that bears on it. The two factors of $\ell\,\varepsilon_t$ are separated because our experiments constrain one and not the other.

Claim	Test	Outcome
Prop. 2: deviation $\propto$ step length; bounded increments beat greedy jumps	Ri1A vs. ODF at 4B; step-matched endpoint test at 1.7B; $c\ell_k$ sweep at 1.7B and 4B	**Supported at 1.7B.** The step-matched, two-seed endpoint test gives $+1.27$ pp for $c\ell_k$=0.10 over 0.50 at equal 706 steps, non-overlapping two-seed ranges, leading on all eight benchmarks (Table 19). The unmatched sweep agrees at 1.7B ($\rho = -1.00$; -0.97 under the Δd correction, one tie) but not at 4B, where no step-matched test exists. The bounded-vs-greedy contrast is separately resolved ($+2.94$ pp pooled, $p<10^{-4}$) but confounds step length with amortisation
Prop. 5: static labels hold a fixed angle	Ri1A vs. static bucketing, three scales	**Directional**: positive on HARD-AVG at all three scales and growing with scale as Cor. 4 predicts, but unresolved on pooled problems at every scale
Cor. 4: gaps $\propto$ distance travelled	Curriculum margin vs. base-to-DAPO displacement across scale	**Consistent**, 3 points, no CI on the relation itself
Prop. 1: Bayesian re-anchoring shrinks ε_t	Frozen-anchor ablation, three scales (threshold only; unseen problems keep their base ranking)	**Not established**: no difference at 1.7B (as Cor. 4 predicts), point estimates without CIs at 4B/8B

Table 9: Training hyperparameters shared by every RL run unless Table 10 states otherwise.

Optimisation (DAPO)		*Ri1A curriculum*	
Framework / precision	verl, bf16	increment $c\ell_k$	0.2
Learning rate	2×10^{-6} (cosine)	`drop_hard_scale` s	1
Rollouts per prompt n	8	`drop_hard_max_ratio`	0.5
Prompts per batch (gbs)	16	`stage_epochs`	2
Mini batch	= gbs (on-policy)	difficulty scan	base policy, n=8, once
`kl_loss_coef` / reference model	0 / none	scan protocol	thinking off, 2048 tokens
Response limit	2048 tokens		
Training pool	Hendrycks-MATH, 7,500	*Evaluation*	
Epochs (no-curriculum runs)	2	metric	mean@8, T=0.7, top-p 0.95
Hardware	2 H200 (141 GB)	max new tokens	8192 (16384 for AIME)
Seed	42 (seed runs: 43/53/33)	engine / seed	vLLM, 42

C.2 How the difficulty bands are defined

Pass rate. For a problem x and policy θ, the pass rate $p_\theta(x)$ is the fraction of n=8 sampled rollouts whose final answer verifies against the reference, so $p_\theta(x) \in \{0, \frac{1}{8}, \ldots, 1\}$. Every band below is defined by p_{θ_0} under the *base* policy, scored once before training and never re-scored.

Scoring protocol. The scan runs with thinking mode disabled under a 2048-token response limit, the generation settings the curriculum itself consumes at training time. The match matters. An earlier scan of the same pool left thinking mode enabled at the same 2048-token limit, which truncates a large share of rollouts inside the reasoning block before any answer is emitted, and a truncated rollout scores as a failure. The two protocols agree on the total zero-gradient mass (62.3% vs 60.7%, Figure 1a) but disagree on where that mass sits, and they induce materially different bands (Appendix C.3). Every band statistic we report, and every band arm we train, uses the training-consistent protocol.

Table 10: Per-experiment deviations from Table 9 and optimizer steps. A dash means the shared setting.

Experiment	Tables	gbs	Hardware	Other deviations	Optimizer steps
Main comparison, 5-hour cap	1, 2, 3	—	—	curricula capped at 5 h	DAPO 938, static 942; ODF 234, AdaCuRL 262 (4B); Ri1A 428 / 846 / 846 (1.7B / 4B / 8B)
Uncapped rerun, 4B	4	—	4 A100 (80 GB), TP=4	no step limit	ODF 375 of 479, AdaCuRL 265 of 948, Ri1A 920
$c\ell_k$ sweep	17, 18	—	—	$c\ell_k \in \{0.10, 0.20, 0.30, 0.33, 0.50\}$; reference model disabled	$950/846/830/782/706$ (1.7B)
Step-matched endpoints	19	8	1 H200	$c\ell_k \in \{0.10, 0.50\}$; seeds $42/43$	706 (both arms)
Deferral vs. discard	21	—	2 H200	band filter also discards mastered problems	Ri1A 428, band 214
Drop-hard scale s	20	—	—	$s \in \{0, 1\}$; lr 1×10^{-6}, `kl_loss_coef` 0.001, `stage_epochs` 1	—
Band study (1.7B)	15, 16	32	—	2,250-problem subsets, plain DAPO	142 (all arms)

Bands. Problems are sorted by p_{θ_0} ascending and cut into four subsets of $k = \lfloor 0.30N \rfloor = 2{,}250$ problems. **hard** is the k of lowest pass rate, **easy** the k of highest, and **mid** the k centred in the ordering, taken from the middle $N-2k$ once both ends are removed so that it is centred on the pass-rate distribution rather than being whatever the ends leave behind. **random** is k problems drawn uniformly from the full pool irrespective of pass rate (seed 42), spanning the whole range as a control for subset size and step count.

The cut is on rank, not on a fixed pass-rate value, so the bands are quantiles rather than thresholds and whether one comes out internally uniform depends on how the pass-rate mass happens to fall. Under the training-consistent protocol the subsets have mean pass rates 0.306 (hard), 0.765 (mid), 0.964 (easy) and 0.688 (random) — so **hard** is not a set of unsolvable problems. The base policy already clears roughly 31% of its rollouts, and a curriculum that discards the hard end therefore discards problems carrying gradient from the first step, which is the empirical case for deferring them instead.

C.3 The two scoring protocols, compared

Table 11 scores the same four subsets under both protocols. The disagreement is not a uniform rescaling; it changes which problems look degenerate, and by how much.

Table 11: Mean base pass rate of each 2,250-problem subset, Qwen3-1.7B on Hendrycks-MATH, $n = 8$. Under the superseded protocol the ends are saturated by construction; under the training-consistent protocol the hard end is far from unsolvable. Band definitions and all band statistics in this paper use the right-hand column.

subset	thinking on, 2048 cap (superseded)	**thinking off,** 2048 **cap** (training-consistent)
hard	0.000	0.306
mid	0.440	0.765
easy	1.000	0.964
random	0.484	0.688

Under the superseded protocol the hard subset scores a uniform $0/8$ and the easy subset a uniform $8/8$, inviting the reading that both ends are degenerate by nature. At the hard end this is an artifact of truncation: those problems are solved about 31% of the time once the model is allowed to answer under the settings training actually uses. The easy end survives the change nearly intact ($1.000 \rightarrow 0.964$), so the two ends are not symmetric — mastery is robust to the protocol, apparent unsolvability is not.

This is the concrete form of the estimation error Section 3 treats abstractly. A single $n{=}8$ scan, and especially one whose generation settings differ from training, does not reliably identify which problems a policy cannot learn from; the failure is one-sided, always overstating difficulty and never understating it, since truncation and undersampling can only convert successes into recorded failures. A filter acting on such an estimate discards live gradient, which is why Ri1A defers the hard end rather than removing it.

C.4 The non-zero-advantage rate, exactly

Figure 1b reports a quantity we call the non-zero-advantage rate. Because it is the only training-side metric we draw conclusions from, we state precisely what is computed.

Within a group. At each optimizer step the policy generates $n{=}8$ rollouts for every prompt in the batch, and each rollout receives a scalar reward from the verifier. The 8 rewards belonging to one prompt form that prompt's *group*. Under group-relative advantages the baseline subtracted from each rollout is the group mean, so a group whose rewards are constant — all 8 correct or all 8 wrong — yields an advantage of exactly zero for every one of its rollouts, and contributes nothing to the gradient regardless of the compute it consumed. A group is *live* when its rewards are not constant, which we detect as $\mathrm{std} > 10^{-6}$ over the 8 values.

Within a batch. The motivation arms use a batch of $B{=}32$ prompts, so one step produces $32 \times 8 = 256$ rollouts. The rewards are reshaped to 32×8, a standard deviation is taken along the rollout axis to give one value per prompt, and each is thresholded to a live/dead indicator. The step's reported value is the mean of those 32 indicators: with 24 live groups the step scores $24/32 = 0.75$. Formally, for step t with rewards r_{ij},

$$\mathrm{NZA}_t \;=\; \frac{1}{B}\sum_{i=1}^{B} \mathbf{1}\left[\mathrm{std}_{j\le n}\left(r_{ij}\right) > 10^{-6}\right]. \tag{22}$$

Across steps. Equation 22 yields one number per optimizer step. The faint traces in Figure 1b are these raw per-step values, whose visible jitter is expected since each is an average of only 32 indicators; the bold curves are a 9-step moving average applied for legibility, and the number in the legend is the unsmoothed mean over all steps of the run.

Two properties follow from the definition and matter for how the figure should be read. First, NZA_t measures only whether a group's rewards vary, never how often they are correct: a band at pass rate 0.5 and a band at pass rate 0.9 can both score high, so the metric is deliberately decoupled from accuracy and is not a proxy for it. Second, it is bounded by the sampling budget. A prompt whose true pass rate is far below $1/n$ will usually return 8 failures and register as dead even though it is not unsolvable, so larger n raises the measured rate without the policy having changed. The values in Figure 1b should therefore be compared across bands at fixed n, as we do, and not read as absolute properties of the prompts.

C.5 Statistics: what each measure asks

DAPO is the one baseline we neither reimplemented nor tuned, giving it the strongest claim to external validity, and it is the one comparison whose per-problem intervals close only at 8B.

The two measures are: a paired bootstrap over test problems ($10{,}000$ resamples) on per-problem correct-count records, and the standard deviation across three training seeds (43, 53, 33) at 1.7B and 4B, which is typically 0.002–0.006. They answer different questions and at 4B they disagree. The paired bootstrap over test problems asks whether a result survives a different draw of *test* problems; the seed standard deviation asks whether it survives a different *training* run. Neither substitutes for the other, and at 4B AIME the two disagree: the seeds separate Ri1A from DAPO by $\approx 6\sigma$, while

the bootstrap interval does not exclude zero. Every per-benchmark table reports the seed-42 run, so that all runs in one table share a single set of checkpoints. That run is held out of the $43/53/33$ triple used for the seed spread, so it is an independent draw rather than a member of the mean it is compared against: at 4B it sits about 1.5 standard deviations *below* the three-seed mean on MATH-500 and Omni-MATH and about 1.5 *above* on AMC23, with Minerva and AIME in between, so the single-seed tables carry no systematic bias in Ri1A's favour. Table 12 gives the pooled bootstrap in full, Table 13 the seed spread at 1.7B and 4B, and Table 14 the seed spread on Llama-3.2-3B.

Table 12: Problem-level paired bootstrap ($10{,}000$ resamples), Ri1A minus baseline in pp. Hard pool: AMC23, AIME23/24/25, Omni-MATH, Minerva (502 problems); all math adds MATH-500 ($1{,}002$). Bold marks intervals excluding zero. Comparisons against the untrained base all resolve at $p < 10^{-3}$ and are omitted.

		hard pool (n=502)		all math (n=1002)	
Scale	Comparison	Δ (pp)	p	Δ (pp)	p
1.7B	vs. DAPO	-0.27	.63	-0.01	.98
	vs. DAPO+static	-0.25	.69	$+0.01$	.99
4B	vs. DAPO	$+0.65$	.29	$+0.75$	.053
	vs. DAPO+static	$+0.80$	.19	$+0.74$	.061
	vs. ODF	$+2.94$	**<.0001**	$+2.21$	**<.0001**
	vs. AdaCuRL	$+2.04$	**.0008**	$+2.08$	**<.0001**
8B	vs. DAPO	$+1.44$	**.015**	$+0.96$	**.014**
	vs. DAPO+static	$+1.12$	.066	$+0.95$	**.016**

Table 13: Three training seeds (43, 53, 33; distinct from the seed-42 run of Table 1), mean@8 $\pm$ standard deviation across seeds; +static abbreviates DAPO+static. Seed standard deviation is typically 0.002–0.006.

	Qwen3-4B				Qwen3-1.7B			
	base	DAPO	+static	**Ri1A**	base	DAPO	+static	**Ri1A**
GSM8K	0.919±.001	0.920±.004	0.918±.003	**0.920**±.005	0.817±.002	0.842±.004	0.835±.004	**0.844**±.003
MATH-500	0.825±.001	0.836±.003	0.844±.004	**0.856**±.003	0.709±.001	0.740±.001	0.740±.004	**0.742**±.002
AMC23	0.660±.003	**0.720**±.002	0.679±.003	0.710±.006	0.457±.000	0.473±.006	0.453±.006	**0.480**±.004
AIME	0.179±.001	0.204±.005	0.201±.005	**0.236**±.002	0.106±.003	0.113±.002	0.119±.004	**0.128**±.005
Minerva	0.298±.003	0.320±.002	0.321±.003	**0.330**±.003	0.215±.002	**0.243**±.002	0.238±.003	0.231±.005
Omni-MATH	0.303±.003	0.331±.002	0.304±.003	**0.356**±.003	0.157±.002	**0.184**±.003	0.183±.003	0.173±.002
HumanEval	0.654±.003	0.657±.002	0.657±.001	**0.668**±.004	0.536±.001	0.547±.002	0.539±.009	**0.552**±.002
MBPP	0.719±.001	0.719±.009	**0.722**±.003	0.712±.004	**0.589**±.004	0.576±.006	0.586±.001	0.587±.003

Table 14: Llama-3.2-3B-Instruct, HARD-AVG of the seed-42 run of Table 1, $\pm$ the standard deviation over three further seeds (43, 53, 33); per-benchmark accuracies are in Table 1. The deviations are smaller than the Ri1A-minus-ODF gap of 0.5 pp, so the ordering is seed-stable.

	base	DAPO	DAPO+static	ODF	AdaCuRL	**Ri1A**
HARD-AVG	0.1227±.002	0.1308±.002	0.1348±.003	0.1439±.003	0.1389±.003	**0.1488**±.004

The trend these numbers describe is plotted in Figure 5: monotone in the pooled difference, and *not* monotone benchmark by benchmark.

C.6 THE COST MODEL IN FULL

The admitted fractions are measured, not assumed: at 4B ODF admitted $1{,}557$ of $7{,}493$ prompts ($\rho = 0.208$) and AdaCuRL $2{,}093$ (0.279). The crossover at which Ri1A's amortised scan becomes cheaper per step than a recurring filter is $S \gtrsim 142$ optimizer steps, which every run in this paper exceeds.

Let N be the pool size, B the prompts per batch, n the rollouts per prompt, S the optimizer steps, E_s the epochs a stage trains its candidate set, and ρ the fraction of sampled prompts an algorithm

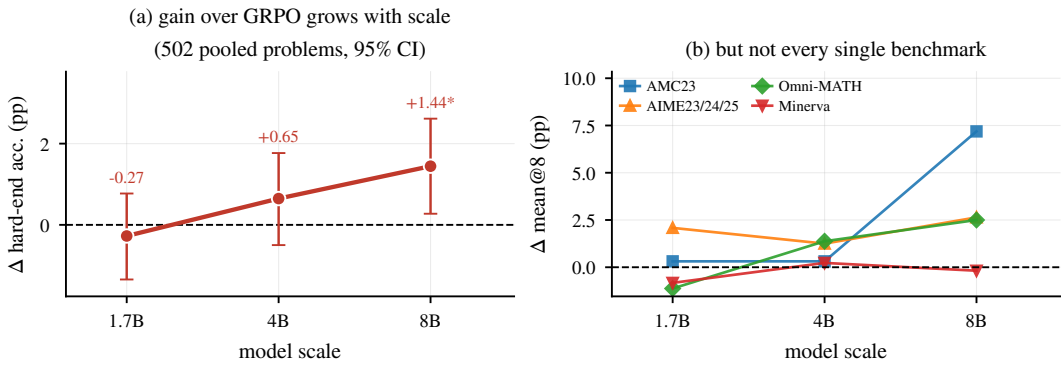


Figure 5: Ri1A minus the full-schedule no-curriculum run (DAPO, labelled GRPO in the panels) across scale. **(a)** Pooled difference over the 502 competition problems with 95% CI: $-0.27/+0.65/+1.44$ pp, resolved only at 8B ($*$). **(b)** Individual benchmarks do *not* all grow monotonically.

admits into the gradient. The re-scoring term deserves a word: over a whole run the re-scored sets total SB/E_s problems, which is what makes the second factor of Ri1A's per-step cost $1/E_s$ rather than 1. Combining that with the one-off Nn scan, Ri1A is cheaper per step than ODF once

$$\frac{C_{\text{step}}^{\text{Ri1A}}}{Bn} = \left(1+\tfrac{1}{E_s}\right)+\frac{N}{SB} < \frac{1}{\rho} = \frac{C_{\text{step}}^{\text{ODF}}}{Bn}, \tag{23}$$

$$\text{that is, once} \qquad S > \frac{N}{B\left(\frac{1}{\rho}-1-\frac{1}{E_s}\right)}, \tag{24}$$

which with the measured $\rho = 0.208$, E_s=2, N=7500 and B=16 is $S \gtrsim 142$ steps — a threshold every Ri1A run in this paper (846 at 4B and 8B, 428 at 1.7B) and ODF's own 234-step run pass. The admission rates are logged per run: at 4B, $\rho = 0.208$ (ODF) and 0.279 (AdaCuRL); at 1.7B, $\rho = 0.250$ and 0.399.

Caveats on the cost model. The model accounts for where the *extra* rollouts go; it is not a wall-clock simulator, since it ignores sequence length (the runs do not generate equal-length responses, Figure 3b) and ODF's 234 steps are where the five-hour cap stopped it, so that step ratio mixes cost with budget. Steps are comparable only *within* methods spending the same Bn per step — which is why 846 against 938 is a meaningful saving and 846 against 234 is not — and we quantify the saving in steps and wall-clock, never in distinct problems consumed, because a deferred problem re-enters a later stage and our logs do not record that count. Two further qualifications specific to the substrate. First, the Bn-per-step entry for the no-curriculum runs assumes DAPO's dynamic sampling is inactive; where it is active the baseline also oversamples to refill a batch, so its per-step cost rises towards Bn/ρ and the gap in Table 5 widens in our favour. We report the conservative figure, and note that Yu et al. (2025) argue the oversampling need not cost wall-clock in a synchronous pipeline, since generation time is dominated by long-tail samples. Second, AdaCuRL's estimation cost in Table 5 is the coarse pass over the pool plus the fine pass over the selected subset, not N=100 generations for every problem.

C.7 Two asymmetries in the budget comparison

Ri1A is *not* the fastest method in total at 1.7B — ODF finishes in 2:56 — but that is a difference in stopping point rather than in cost per step.

Every method run to self-termination. Since the five-hour cap binds at 4B, we re-ran all three on 4 A100 GPUs with the step limit removed (Table 4). Neither baseline approaches the limit its own configuration computes — ODF halts at 375 of 479 when a round admits no examples, AdaCuRL at 265 of 948 once its buckets merge and the pool is spent — the latter almost where the cap left it — while Ri1A runs 920 steps and leads by $+4.49$ and $+3.89$ pp HARD-AVG, wider than under the cap, so the accuracy ordering is not an artifact of the budget.

Where ODF's wall-clock goes. Each of ODF's 55 scoring rounds re-scores the *entire* remaining pool at 8 samples per problem with no cache and then runs a full validation pass unconditionally, the last round still holding 4,717 unscored problems; AdaCuRL caches its scores, re-scores only a fixed fraction of the pool each round, and validates on the usual `test_freq` schedule, which is what separates 6.8 h from 69. Most of ODF's generation budget therefore buys difficulty estimates rather than gradient updates, so the measured $1/\rho$=4.8, which counts admission only, makes $4.81\,Bn$ a lower bound on its true per-step cost.

Section 7.3 defers two qualifications here, both of which cut against us.

The cap is not applied uniformly. The three curricula share the five-hour cap; DAPO and DAPO+static run their full two-epoch schedules at roughly twice the hours. The curricula are thus constrained in a way the strongest baseline is not, so our ODF and AdaCuRL margins are margins under a cap rather than at convergence. A reader who wants the convergence comparison should treat Table 1's ODF and AdaCuRL rows as lower bounds on what those methods reach with unlimited time.

Capped step counts mix cost with budget, and we measured both. Under the cap, the step ratio 846:234 conflates two quantities — how expensive each step is, and how many steps fit in the budget. Only the first is a property of the algorithm, and only the first is what our cost model addresses; we therefore quantify efficiency as the per-step rollout cost ($2.05\,Bn$ against $4.81\,Bn$, both computed from admission rates the baselines themselves log) and as wall-clock against DAPO.

The control this argument previously lacked — every method given unlimited time at 4B — is Table 4. We re-ran all three with `total_training_steps` left at -1, so each computes its own limit and stops itself, on 4 A100 GPUs at gbs 16, response limit 2048 and TP=4; every algorithm hyperparameter is unchanged from the capped runs, and `stage_epochs`=2 throughout. ODF terminated at 375 of a possible 479 steps when a round admitted no examples under its t_{low}=0.2, t_{high}=0.8 filter; AdaCuRL terminated at 265 of 948 once its three buckets merged and the training pool was consumed (2,110 unique problems); Ri1A completed its stage schedule at 920 steps. Both baselines stopped well short of their computed limits, AdaCuRL almost exactly at its capped step count (265 vs 262) and ODF further on (375 vs 234), with per-benchmark results agreeing to within a few points across the two conditions. Their short schedules are therefore a property of consume-once curricula at this pool size rather than an effect of our cap, and the accuracy ordering survives when no method is constrained — by a wider margin than under the cap, since Ri1A reaches 0.4290 HARD-AVG here against 0.4095 in Table 1. Two things separate those runs: the uncapped one completes its schedule at 920 steps rather than the 846 the cap allowed, and it ran on 4 A100s rather than 2 H200s. Most of the +1.95 pp difference falls on AMC23 (+5.00 pp on 40 problems), so we compare methods only among runs sharing a hardware and budget condition, and read Table 4 as corroborating the capped ordering rather than as a second estimate of Ri1A's absolute accuracy.

One caveat. These runs used 4 A100s rather than the 2 H200s of Table 2, so runtimes are not comparable across the two tables; within this table, however, all three methods share the same four GPUs, so their runtimes are directly comparable with one another. The dominant term in ODF's ≈69 h is its own selection overhead: it re-scores the entire remaining pool every round without caching and validates in full after each, which Section 7.3 quantifies.

C.8 THE CONTROLLED BAND STUDY

Table 15 is the hard-end view of the four-band experiment of Section 5, and Table 16 the easy-end and code benchmarks, which move far less (under 1 pp for every run except A-mid on GSM8K and MATH-500, +1.0 and +1.7 pp) — the reason the effect is reported on the hard end only. The study varies only *which segment* of the pool is trained on, so its conclusion does not depend on the absolute calibration of the difficulty scores: the two scoring protocols of Figure 1a agree on the total zero-gradient mass but not on its split between the two ends, which is why we claim only the total.

Figure 6 asks the same question on the test side, per problem rather than per band. It is directional only: the sole positive bin is $(0.25, 0.5]$ and its interval crosses zero.

Table 15: Controlled band study, Qwen3-1.7B: equal data (2,250 problems), equal steps (142), plain DAPO, seed 42. The last column is the change in HARD-AVG against the untrained base, in pp. Only the middle and hard bands improve on base.

	AMC23	AIME	Minerva	Omni-MATH	HARD-AVG	Δ vs. base
base (ref.)	0.4562	0.1042	0.2160	0.1550	0.2329	—
A-easy	0.4281	0.0944	0.2224	0.1613	0.2266	−0.63
A-random	0.4250	**0.1097**	0.2155	0.1562	0.2266	−0.63
A-hard	**0.5000**	0.0931	0.2224	0.1675	0.2457	+1.29
A-mid	0.4813	0.1000	**0.2270**	**0.1800**	**0.2471**	+1.42

Table 16: Band study of Table 15, easy-end and code benchmarks: every run moves by less than 1 pp against base except A-mid on GSM8K (+1.0) and MATH-500 (+1.7).

	GSM8K	MATH-500	HumanEval	MBPP
base (ref.)	0.8197	0.7093	0.5343	0.5890
A-easy	0.8234	0.7170	0.5373	**0.5905**
A-random	0.8223	0.7180	0.5412	**0.5905**
A-hard	0.8202	0.7175	0.5358	0.5856
A-mid	**0.8301**	**0.7265**	**0.5442**	0.5875

C.9 ABLATIONS IN DETAIL

Increment size $c\ell_k$. Table 17 gives the full five-point sweep at 1.7B and Table 18 the four-point sweep at 4B, both at the recipe's other settings (`drop_hard_max_ratio`= 0.5, `drop_hard_scale`= 1, `stage_epochs`= 2, seed 42). At 1.7B the ordering on HARD-AVG is strictly decreasing in the increment, $0.10 > 0.20 > 0.30 > 0.33 > 0.50$ ($0.2654 \to 0.2524$; Spearman $\rho = -1.00$ over the five points), which is the direction Proposition 2 predicts. At 4B it is not: $0.20 \gtrsim 0.10 > 0.50 > 0.30$, so the recipe value is competitive with the smallest increment and both clear the other two by 1–2 pp, but the curve does not decrease in $c\ell_k$.

Two confounds bar reading the sweep *as run* as a test of the mechanism, and both are visible in the figure. *Steps are not matched*: a smaller $c\ell_k$ produces more, shorter stages and hence more optimizer steps ($950/846/830/782/706$ across the sweep), and that step count is itself perfectly rank-correlated with HARD-AVG (Figure 7c, $\rho = +1.00$). Within this sweep, then, "smaller increments win" and "longer runs win" make identical predictions, and the confound points the same way the hypothesis does. *Seeds are not replicated*: every point is a single seed, and the spread between adjacent points (0.002–0.005 HARD-AVG) is of the same order as the 0.002–0.006 training-seed standard deviation of Table 13. We therefore do not read the sweep itself as evidence for Proposition 2; the test that separates the hypothesis from both confounds is the step-matched, two-seed endpoint comparison of Table 19 below, which is what we rely on.

Reading the sweep on a metric axis. Appendix B.5 argues that $c\ell_k$ is a quantile width and Proposition 2 is about a metric step length, the two related by $\Delta d \approx c\ell_k / f$. Figure 7b carries out that correction on the 1.7B sweep, using the base-policy difficulty c.d.f. measured before training, and the result leaves the conclusion unchanged: the $\max_i \Delta d_i$ axis preserves the $c\ell_k$ ordering up to one tie ($\rho = -0.97$), so on this sweep re-expressing $c\ell_k$ on a base-policy difficulty axis does not change the conclusion. We report it because the check was cheap, was specified in advance by Appendix B.5, and could have come out otherwise. We have not run it at 4B.

The step-matched endpoint test. The step-count confound is removable, and Table 19 removes it. We rerun the two endpoints of the sweep, $c\ell_k \in \{0.10, 0.50\}$, as a self-contained pair: two training seeds each ($42/43$), gbs 8 on a single H200 GPU, and both arms evaluated at the *same* 706 optimizer steps. Holding the step count fixed is what matters here, since Figure 7c shows it to be the quantity that tracks the outcome perfectly across the sweep; with it fixed, the increment is the only thing left varying. The smaller batch does mean the pair is comparable to itself and to nothing else, so we read only the difference between its two columns and never place either column beside Table 17 or Table 1.

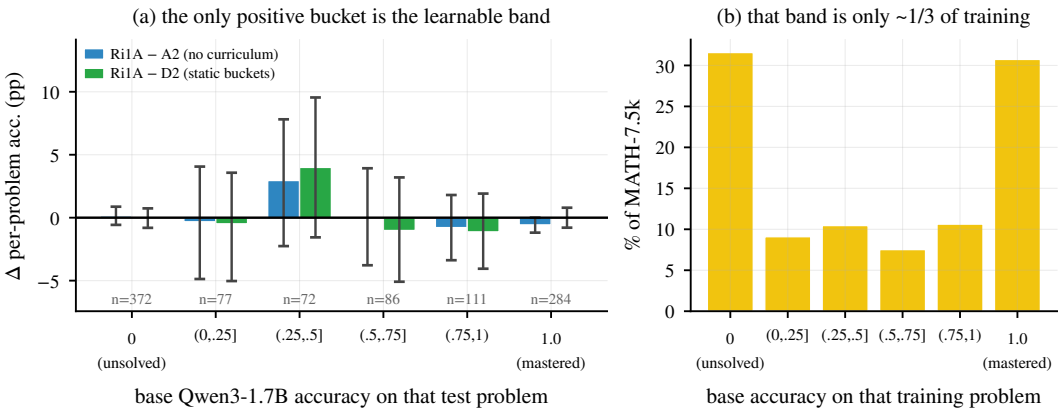


Figure 6: Test-side learnable band, Qwen3-1.7B. **(a)** Per-problem accuracy change of Ri1A over two baselines (A2 = DAPO, D2 = DAPO+static; the panel titles are descriptive, not claims), binned by base accuracy (5 math benchmarks pooled, 95% CI). The only positive bin is $(0.25, 0.5]$, and all intervals cross zero. **(b)** The same binning on the training set. **This figure supports no claim** — every interval covers zero, so it is reported for completeness and because the binning is the natural test-side analogue of the training-side band study, not as evidence for a learnable band on the test distribution.

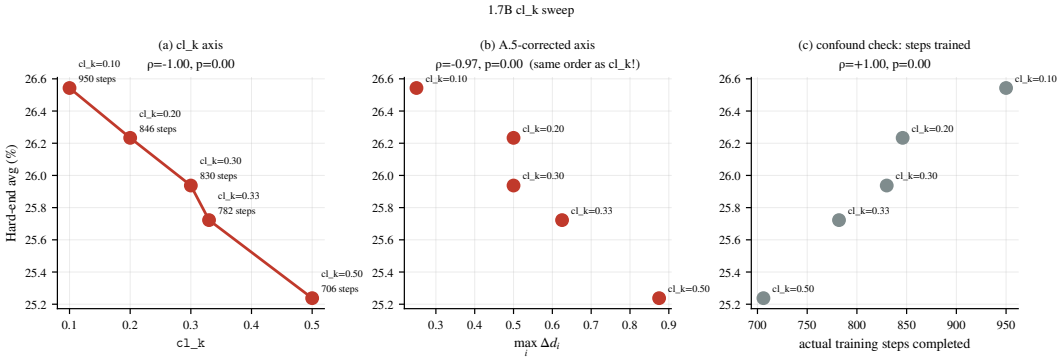


Figure 7: The 1.7B $c\ell_k$ sweep read three ways. **(a)** HARD-AVG against $c\ell_k$: strictly decreasing, Spearman -1.00. **(b)** The same five runs against $\max_i \Delta d_i$, the largest metric difficulty span any stage admits, computed from the base-policy difficulty c.d.f. by the $\Delta d \approx c\ell_k / f$ relation of Appendix B.5. The correction preserves the ordering up to one tie ($c\ell_k$=0.20 and 0.30 coincide; Spearman -0.97; the panel label "A.5" refers to Appendix B.5), so on this sweep the ordering survives the change of axis — a check that could have failed and did not, rather than a measurement of ℓ. **(c)** The confound: steps completed rank-correlate with HARD-AVG at $\rho = +1.00$, because a smaller $c\ell_k$ mechanically yields more optimizer steps. Panels (a) and (c) fit these five runs equally well, which is why we do not read the sweep itself as a test of Proposition 2 and instead rerun its endpoints at matched steps and two seeds (Table 19), where the smaller increment wins by $+1.27$ pp with the step count held fixed.

The smaller increment wins by $+1.27$ pp HARD-AVG (0.2645 vs 0.2518), leads on *all eight* benchmarks, and its two-seed range clears the larger increment's on all four competition benchmarks. The spread is across *training* seeds rather than test-problem resampling — the stricter of the two measures we report — though with two seeds per arm it is a range and not an interval, so we claim only that the gap is well outside the run-to-run variation we can see (± 0.0024 on HARD-AVG, inside the 0.002–0.006 band of Table 13) and attach no p-value to it. The gap also survives on the two out-of-domain code probes, which is mild further evidence: a mechanism that merely trains longer has no particular reason to move held-out code benchmarks.

Table 17: Increment-size sweep, Qwen3-1.7B, gbs 16 / response limit 2048, `drop_hard_max_ratio`$= 0.5$.

$c\ell_k$	GSM8K	MATH-500	AMC23	AIME23/24/25	Minerva	Omni-MATH	HumanEval	MBPP	HARD-AVG
0.10	**0.8457**	0.7500	**0.4906**	**0.1319**	0.2454	**0.1938**	0.5556	0.5793	**0.2654**
0.20 (recipe)	0.8413	**0.7548**	0.4875	0.1222	**0.2459**	**0.1938**	**0.5678**	0.5778	0.2623
0.30	0.8414	0.7472	0.4781	0.1264	0.2417	0.1913	0.5495	**0.5919**	0.2593
0.33	0.8350	0.7442	**0.4906**	0.1153	0.2330	0.1900	0.5480	0.5880	0.2572
0.50	0.8384	0.7438	0.4719	0.1306	0.2321	0.1750	0.5465	0.5875	0.2524

Steps completed, in $c\ell_k$ order: $950/846/830/782/706$. A smaller increment mechanically yields more optimizer steps, the confound discussed in the text and plotted in Figure 7c. The $c\ell_k$=0.2 entry is this sweep round's 846-step run; the main-table Ri1A run of Table 1 shares the setting but is a separate 428-step launch reporting 0.2542, and the two are not interchangeable.

Table 18: Increment-size sweep, Qwen3-4B, gbs 16 / response limit 2048, `drop_hard_max_ratio`$= 0.5$.

$c\ell_k$	GSM8K	MATH-500	AMC23	AIME23/24/25	Minerva	Omni-MATH	HumanEval	MBPP	HARD-AVG
0.10	**0.9233**	0.8468	0.7156	0.2264	**0.3382**	**0.3563**	0.6593	0.7135	0.4091
0.20 (recipe)	0.9219	**0.8515**	**0.7188**	**0.2361**	0.3318	0.3513	0.6646	0.7130	**0.4095**
0.30	0.9216	0.8445	0.7125	0.1931	0.3244	0.3237	**0.6662**	**0.7194**	0.3884
0.50	0.9229	0.8462	**0.7188**	0.2181	0.3249	0.3250	0.6654	0.7111	0.3967

This is the most direct test of Proposition 2 we have, and it passes. Three things bound how far that carries — two seeds rather than three, a batch size that isolates the pair from every other table, and one model scale — so Table 8 records the prediction as **supported at 1.7B** rather than established in general.

Table 19: Step-matched endpoint test, Qwen3-1.7B: $c\ell_k$=0.10 vs 0.50, two training seeds $(42/43)$ each, both arms evaluated at 706 optimizer steps. Entries are the two-seed mean, with brackets giving mean $\pm$ one standard deviation *across training seeds* — the measure that asks whether a result survives a different training run rather than a different draw of test problems (Appendix C); with n=2 these are descriptive ranges, not confidence intervals. Increment size is the only difference between the arms, the step-count confound of Figure 7c being held fixed by construction. **Both arms run gbs 8 on a single H200 GPU, unlike the main tables (gbs 16, 2 GPUs), so the two columns are comparable to each other and to nothing else**; the quantity we claim is their difference. The smaller increment leads on every benchmark, and on the four competition benchmarks the ranges do not overlap.

	$c\ell_k = 0.10$	$c\ell_k = 0.50$
GSM8K	**0.8432** [0.8401, 0.8467]	0.8371 [0.8340, 0.8396]
MATH-500	**0.7486** [0.7452, 0.7521]	0.7420 [0.7394, 0.7445]
AMC23	**0.4873** [0.4835, 0.4912]	0.4704 [0.4671, 0.4736]
AIME23/24/25	**0.1304** [0.1270, 0.1336]	0.1201 [0.1168, 0.1233]
Minerva	**0.2439** [0.2410, 0.2468]	0.2314 [0.2286, 0.2341]
Omni-MATH	**0.1916** [0.1892, 0.1941]	0.1742 [0.1719, 0.1766]
HumanEval	**0.5531** [0.5495, 0.5566]	0.5432 [0.5400, 0.5458]
MBPP	**0.5834** [0.5794, 0.5870]	0.5811 [0.5775, 0.5842]
HARD-AVG	**0.2645** [0.2621, 0.2669]	0.2518 [0.2494, 0.2541]

The deferral cap is unswept, and we withdraw the sweep entirely. An earlier draft reported a five-point sweep over `drop_hard_max_ratio` showing an inverted U peaking at the recipe value; a later draft replaced it with a three-point sweep $(0/0.5/1.0)$ reported as monotone decreasing. In auditing the run artifacts for this version we established that *neither sweep exists*: every training run in this paper, without exception, was launched with `cl_drop_hard_max_ratio`$= 0.5$. The runs the earlier tables drew on vary `cl_drop_hard_scale` — the multiplier s in Eq. 7, which shifts the admission *threshold* — and not the cap on how much of a stage may be deferred. Both sweeps are therefore withdrawn, and with them any claim about the cap's optimum.

What the artifacts do support is a two-point comparison on s, which we report in Table 20 with its confounds stated rather than repaired. The two runs share $c\ell_k$=0.2, seed and hardware, but were launched in an early round at lr $= 1 \times 10^{-6}$, `kl_loss_coef` $= 0.001$ and `stage_epochs` $= 1$ — three settings that all differ from the recipe of Section 6. They are consequently *not* comparable to any other table in this paper, and in particular the main-table Ri1A run is not a third point on this axis; we give them their own block for that reason. Read on their own terms, s=0 (threshold at $\bar{\delta}_i$) is ahead of s=1 (the recipe value, threshold raised to $\bar{\delta}_i(1 + c\ell_k)$) by 1.0 pp HARD-AVG — the opposite of the recipe's setting, with one seed, one pair and no interval. The recipe value of s is therefore a default rather than a tuned choice (Table 7).

Table 20: The only ablation we hold on the deferral machinery: `cl_drop_hard_scale` s at Qwen3-1.7B, both runs at $c\ell_k$=0.2, cap $= 0.5$, seed 42. **These two runs use lr $= 1 \times 10^{-6}$, `kl_loss_coef` $= 0.001$ and `stage_epochs` $= 1$, and are therefore not comparable with Table 1 or any other table here.** The deferral *cap* itself is never swept in this paper; both previously reported cap sweeps are withdrawn.

s	GSM8K	MATH-500	AMC23	AIME23/24/25	Minerva	Omni-MATH	HumanEval	MBPP	HARD-AVG
0	**0.8219**	**0.7153**	**0.4562**	**0.1083**	**0.2238**	**0.1737**	0.5404	**0.5880**	**0.2405**
1 (recipe value)	0.8205	0.7135	0.4344	0.0972	0.2224	0.1663	**0.5434**	0.5875	0.2301

Bold marks the better of the two per column. On this early-round configuration s=0 is ahead on the hard end by 1.0 pp, i.e. the *opposite* of the recipe's setting; with one seed, no interval, and a configuration that matches no other run in the paper, we draw no conclusion from it beyond recording that the setting is untested at the recipe's hyperparameters.

Deferral vs. discard. Replacing deferral with a hard band filter that also discards mastered problems, with both arms on the same single H200 GPU, lowers HARD-AVG by 2.15 pp (0.2327 vs 0.2542) at half the steps and costs 1.7 and 2.8 pp on GSM8K and MATH-500. Part of the mechanism is in the data: 30.7% of the pool is mastered at base, so discarding it truncates training to 214 steps and leaves the cosine learning-rate schedule badly under-decayed. The comparison is a single seed per arm with no interval.

Table 21: Deferral vs. discarding, Qwen3-1.7B, both arms on the same 2xH200 GPU. "Band" also discards mastered problems.

	Ri1A (defer)	Band
GSM8K	**0.8400**	0.8233
MATH-500	**0.7442**	0.7160
AMC23	**0.4781**	0.4500
AIME23/24/25	**0.1347**	0.0944
Minerva	**0.2325**	0.2252
Omni-MATH	**0.1713**	0.1613
HARD-AVG	**0.2542**	0.2327
Optim. steps	428	214

Re-anchoring vs. frozen difficulty. Setting `freeze_difficulty_at_base` replaces the stage-$i{+}1$ anchor — normally the pass rate the *current* policy achieves on what stage i trained on — with the base-policy score, switching off the scalar threshold update. The matched pair at 1.7B shows no difference (pooled $\Delta = +0.14$ pp, p=.56), as Corollary 4 requires where the whole curriculum effect vanishes; the 4B and 8B pairs favour re-anchoring by $+0.84$ and $+1.61$ pp with the gain falling entirely on the hard benchmarks, but those runs were evaluated off-site without per-problem records and carry no intervals. Both arms also rank the unseen pool by the same base-policy scores, so the ablation varies only the scalar threshold and never tests re-estimating the difficulty of problems not yet trained on. We therefore do *not* claim this ablation establishes Proposition 1.

C.10 THE CURRICULUM IN SFT, AND WHY IT IS NOT EVIDENCE OF TRANSFER

Table 6 applies the construction to supervised fine-tuning, reading competence from an inverse-perplexity weight rather than a pass rate, which selects 9,547 of 13,452 sequences (71%). The result

Table 22: Re-estimation minus frozen anchor (pp), everything else held fixed. This table does *not* establish Proposition 1: the 1.7B pair shows no measurable difference (p=.56 pooled), and the 4B/8B entries are point estimates without intervals. At 4B/8B the difference falls entirely on the hard benchmarks, as a threshold-only intervention should.

	hard-end mean	per benchmark				
Scale	3 bm	GSM8K	MATH-500	AIME	Minerva	Omni
1.7B	$+0.27$	$+0.24$	$+0.15$	$+0.97$	-0.78	$+0.63$
4B	$+0.84$	0.00	0.00	$+1.00$	$+0.64$	$+0.87$
8B	$+1.61$	0.00	0.00	$+2.00$	$+1.21$	$+1.62$

is positive on all three out-of-distribution benchmarks but cannot be attributed to the ordering effect, for two reasons we state rather than argue around. First, objective *and* corpus change together: these runs fine-tune on GSM8K, not the Hendrycks-MATH pool the RL results use, so a gain is equally consistent with competence-weighted data selection helping SFT for reasons unrelated to the geodesic account. Second, the SFT arm is a single run with no intervals. We therefore treat Table 6 as a demonstration that the same competence-weighting machinery *runs* outside RL, not as evidence that the geometric ordering effect transfers to it; Table 7 records the claim as not established.

C.11 Training dynamics

Two caveats bear on Figure 3. Because the panels are not compute-aligned, an equal-token crop of the first 40M tokens would show Ri1A still pinned near 1.0: with $c\ell_k$=0.2 and `stage_epochs` = 2, the easiest fifth is trained twice before the next stage opens. And only four curves appear because the AdaCuRL 4B run logged no training metrics.

C.12 The 1.7B Minerva/Omni-MATH regression, examined

The main text (Section 9) flags a regression against plain DAPO on Minerva and Omni-MATH at 1.7B that Corollary 4 does not cover: the corollary bounds how a curriculum gap *shrinks* with displacement, not whether it can change sign, so this result sits outside what the geometry predicts. From Table 13, DAPO leads Ri1A by $+0.012$ on Minerva (0.243 vs. 0.231) and $+0.011$ on Omni-MATH (0.184 vs. 0.173), both outside the 0.002–0.005 seed spread. Re-anchoring does not account for both: at 1.7B Ri1A's *own* frozen-anchor variant (Table 22) is ahead on Minerva (-0.78) but behind on Omni-MATH ($+0.63$), so switching it off would recover at most one regression. Two candidate mechanisms remain untested. First, Minerva and Omni-MATH have the least topical overlap with MATH's competition-style problems, so a curriculum that spends more budget near the base model's ability frontier could be trading breadth for depth at this scale. Second, 1.7B is the one Qwen3 scale where Ri1A's schedule completes rather than stopping at the five-hour cap, so it alone reaches the final stage, where the hardest deferred problems are admitted and any left over are discarded. Distinguishing these needs per-problem records on out-of-distribution items that this paper lacks, so we report the regression as an open anomaly outside the geometry's account.