TERMINATION CALIBRATION RECOVERS ANSWERS THAT REASONING MODELS COMPUTE BUT NEVER EXPRESS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large reasoning models trained with reinforcement learning from verifiable rewards (RLVR) achieve state-of-the-art performance, but they are evaluated under generous token limits while deployment imposes hard budgets. We show that under such budgets, these models fail primarily through *termination failure* rather than capability failure: nearly half of a distilled model's truncated generations already contain the correct answer—computed, but never expressed. Existing efficiency methods do not resolve this. Length-penalized rewards teach models to be shorter, yet their stopping points remain coupled to whatever budget is available; and we find the standard correctness-only RLVR recipe can actively worsen termination, lengthening typical traces and dropping tight-budget accuracy below that of the untrained base model. We introduce *termination calibration* through **R3**, a single trajectory-level reward granting full credit only when a correct answer terminates within a threshold that tracks the policy's own recent correct-solution lengths. Our design (i) requires no auxiliary judge, token-level labels, or curriculum, and (ii) includes a matched correctness-only control that isolates the causal mechanism. Across budgets and datasets, R3 matches or exceeds specialized efficiency methods at roughly half their tokens and twice the speed; its learned stopping rule is budget-invariant yet extends adaptively to problems harder than any seen in training, and the effect replicates across three backbones. Efficient reasoning, we argue, is less about learning to be brief than learning when to stop.

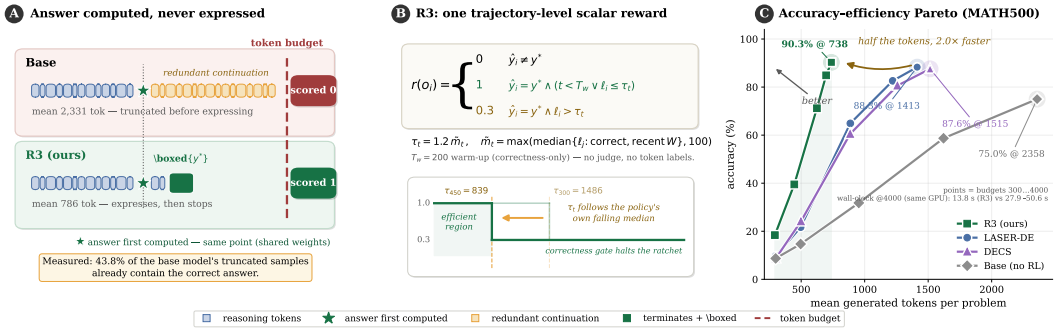


Figure 1: Reasoning models compute answers they never express; pricing termination recovers them. **(A)** Under a hard budget (MATH500, 4,000 tokens), the base model reaches its answer (★) but generates until truncation and is scored 0 — 43.8% of its truncated outputs already contain the correct answer (Appendix G.3) — while R3 expresses it and stops (786 vs. 2,331 mean tokens, sampled protocol). **(B)** The complete method: one trajectory-level reward; τ_t tracks the policy's own recent correct lengths (inset: 1,486→839 over steps 300–450). No judge, token labels, or curriculum. **(C)** Pareto on MATH500 (greedy, budgets 300–4,000): R3 reaches 90.3% at 738 tokens where the base needs 3.2× more for 75.0%; timing on identical hardware (Appendix 11 and 6).

1 INTRODUCTION

Reinforcement learning from verifiable rewards has made extended chain-of-thought generation the dominant recipe for reasoning, with models such as o1 (Jaech et al., 2024), DeepSeek-R1 (Guo

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.

et al., 2025) and Kimi k1.5 (Kimi Team, 2025) converting additional inference tokens into accuracy. The cost of this recipe is now well documented: reasoning-specialised models routinely generate far longer traces than their answers require, a phenomenon termed overthinking (Chen et al., 2024; Sui et al., 2025). Several families of remedies have emerged. Length-penalised training augments the correctness reward with a price on tokens — from fixed or shaped penalties (Kimi Team, 2025; Arora & Zanette, 2025; Yang et al., 2025) to difficulty-aware dynamic targets (Liu et al., 2026a) and per-problem histories of previously correct lengths (Huang et al., 2026a). Inference-time control instead constrains generation directly, by forcing budgets at decode time (Muennighoff et al., 2025) or by training models to obey a length constraint stated in the prompt (Aggarwal & Welleck, 2025). A third line routes between reasoning modes, deciding per problem whether and how long to think (Fang et al., 2026; Zhang et al., 2025; Liu et al., 2026b). At the frontier of the first family, DECS (Jiang et al., 2026) argues from theory that trajectory-level length rewards are structurally insufficient — penalising essential exploration while rewarding partial redundancy — and therefore introduces token-level credit assignment with an auxiliary judge model and curriculum scheduling.

All of these methods, however diverse their machinery, pursue the same objective under the same protocol: teach the model to produce *shorter* traces, and evaluate at limits generous enough that traces complete. Deployment inverts the protocol — latency and cost impose hard budgets — and under hard budgets we find the field has been optimising the wrong quantity. Scanning the truncated generations of a 7B distilled model on MATH-500, nearly half already contain the correct answer: computed, but never expressed (Appendix G.3 and G). The bottleneck is not the length of the reasoning but the absence of a decision to stop, and no existing reward, penalty, or controller trains that decision directly. This reframing raises the question the present work addresses: *What prevents reasoning models from expressing their capability under constrained token budgets, and can a single reward signal train it away?* (Extended related work in Appendix B.)

This study addresses the question through a three-phase design. We *diagnose* by evaluating 11 models spanning three DeepSeek-R1-distilled backbones (Qwen-7B, Llama-8B, Qwen-1.5B) (Guo et al., 2025) on GSM8K, MATH-500, and ARC-Challenge (Cobbe et al., 2021; Hendrycks et al., 2021; Clark et al., 2018) under five token budgets (300–4,000) with a uniform prompt, decoder and answer extractor — approximately 49,500 budgeted assessments — decomposing every accuracy figure into completion rate and per-completion accuracy. We *intervene* causally: R3, a single trajectory-level reward granting full credit only when a correct answer terminates within a threshold tracking the policy's own recent correct-solution lengths, is trained against a matched correctness-only control (R1) sharing data, steps and optimiser, so their difference isolates the termination signal; a within-run control (200 warm-up steps of correctness-only reward inside the R3 run itself) timestamps the mechanism. We *verify* that what is learned is calibration rather than compression, using budget-response curves, sampled pass@1–8 on MATH-500 and AIME 2024, and replication across all three backbones.

Four findings emerge. **First**, budget-constrained failure is termination failure: per-completion accuracy is nearly budget-invariant for every model wherever completions are plentiful, so the steep accuracy–budget curves of reasoning models are completion curves, not capability curves (Appendix D.5). **Second**, termination is trainable by one scalar reward: R3 raises MATH-500 accuracy at the 4,000-token budget from 75.0% to 90.3% while reducing mean generation from 2,358 to 738 tokens, with median length falling near-monotonically from the exact step the signal activates ($r = -0.973$; Appendix C, Table 11 and 6). **Third**, the standard correctness-only recipe moves termination the *wrong* way: it lengthens typical traces and reduces tight-budget accuracy below the untrained base model on the primary backbone (Table 1; median lengths in Table 43). **Fourth**, the learned rule is calibration rather than compression: token usage is budget-invariant yet extends with difficulty beyond the training distribution — median generation rises $333 \rightarrow 576 \rightarrow 1{,}175$ across GSM8K $\rightarrow$ MATH-500 $\rightarrow$ AIME — while pass@1–8 exceeds the base model on MATH-500 and matches it on AIME (Tables 43, 2, 22, 30). These patterns give rise to four corollaries:

Corollary 1. *Evaluation at generous limits systematically misorders methods for deployment: under tight budgets the correctness-only model trails its own base by up to 25 points, and a token-level method with judge and curriculum (Jiang et al., 2026) trails a simpler trajectory-level length reward (Liu et al., 2026a) at every budget* $\geq$1,000 *tokens (12 of 15 budget–dataset cells overall; Table 1 and 41).*

Corollary 2. *Stopping behaviour forms a coupling spectrum: when the budget doubles from 2,000 to 4,000 tokens, generation grows $+45.5\%$ (base), $+31.7\%$ (correctness-only), $+20.4\%$ DECS (Jiang et al., 2026), $+15.7\%$ LASER (Liu et al., 2026a) and $+6.0\%$ (R3) — separating methods that learn to be shorter from one that learns to stop (Table 1, Δtok; all models in Table 18).*

Corollary 3. *Policy-entropy decline under R3 ($0.57 \to 0.26$) reflects commitment to termination, not exploration collapse: pass@k exceeds the base model at every $k \in \{1, 2, 4, 8\}$ (Figure 3d; Table 22).*

Corollary 4. *Budget-coupling is a property of the reward, not the backbone: within every backbone triad on MATH-500, token growth under budget doubling orders Base > R1 > R3 ($+45.5$>$+31.7$>$+6.0$; $+58.5$>$+32.7$>$+28.1$; $+59.9$>$+57.0$>$+32.3$), and the 7B R3 arm holds the smallest coupling in the entire 11-model grid on both mathematics benchmarks — including an exact $+0.0\%$ on GSM8K (Table 18).*

2 METHOD

2.1 PRELIMINARIES

We consider reasoning models trained with reinforcement learning from verifiable rewards. Given a prompt x, the policy π_θ samples a group of G rollouts $\{o_i\}_{i=1}^{G}$; each rollout has an extracted answer $\hat{y}_i$, a ground truth y^*, and a length ℓ_i in tokens. We train with Group Relative Policy Optimization (GRPO) (Shao et al., 2024), which replaces PPO's value network with group-normalised advantages,

$$\hat{A}_i \;=\; \frac{r(o_i) - \mathrm{mean}\{r(o_j)\}_{j=1}^{G}}{\mathrm{std}\{r(o_j)\}_{j=1}^{G}}, \tag{1}$$

optimised under the standard clipped surrogate objective (Eq. 4, Appendix C). Everything this paper contributes enters through the scalar $r(\cdot)$; architecture, optimiser, and objective are unchanged. At deployment, generation is subject to a hard budget B: a rollout that has not terminated within B tokens is truncated and scored on whatever it has produced.

2.2 BUDGETED ACCURACY DECOMPOSES INTO COMPLETION AND CAPABILITY

Let $c(B)$ denote the fraction of rollouts that terminate within budget B, $A_{\mathrm{c}}(B)$ the accuracy of those completed rollouts, and $A_{\mathrm{t}}(B)$ the accuracy credited to truncated ones (an answer may still be extractable from an unfinished trace). Budgeted accuracy then obeys an exact identity:

$$\mathrm{Acc}(B) \;=\; c(B)\,A_{\mathrm{c}}(B) \;+\; \big(1 - c(B)\big)\,A_{\mathrm{t}}(B). \tag{2}$$

Eq. 2 is not a model of behaviour but an accounting of it, and it separates two hypotheses that the literature conflates. If budget failure is a *capability* problem, tighter budgets should depress $A_{\mathrm{c}}(B)$: the reasoning that fits inside B tokens is simply worse. If it is a *termination* problem, $A_{\mathrm{c}}(B)$ should be flat in B while $c(B)$ collapses — the model reasons as well as ever but rarely finishes — and the residual term should be non-trivial, because answers are being computed inside traces that never close. §3 measures every term and finds the termination signature throughout. The design principle follows directly: *raise $c(B)$ at every B simultaneously, without touching A_{c}*. A reward that prices tokens attacks length, which raises c only at the budgets the shortened traces happen to clear; a reward that prices the decision to stop attacks $1{-}c$ itself.

2.3 R3: A SELF-CALIBRATING TERMINATION REWARD

R3 implements that principle as a single trajectory-level scalar. Writing t for the training step,

$$r(o_i) = \begin{cases} 1, & \hat{y}_i = y^* \;\wedge\; \big(t < T_w \;\vee\; \ell_i \le \tau_t\big), \\ 0.3, & \hat{y}_i = y^* \;\wedge\; \ell_i > \tau_t, \\ 0, & \hat{y}_i \neq y^*, \end{cases} \tag{3a}$$

$$\tau_t = 1.2 \cdot \max\big(\,\mathrm{med}\{\ell_j : \hat{y}_j = y^*,\; j \in \mathcal{W}_t\},\; 100\big), \tag{3b}$$

where $\mathcal{W}_t$ holds the $W{=}5{,}000$ most recent correct rollouts and $T_w{=}200$ warm-up steps leave the termination term inactive (doubling as a control; §2.4).

Four properties of Eq. 3 do all of the work. **Self-calibration.** The threshold τ_t is not a hyperparameter expressing how long we believe solutions should be; it is a running median of how long the policy's own recent *correct* solutions actually are, with 20% headroom. As the policy discovers shorter correct solutions the median falls, the threshold follows, and full credit migrates toward earlier termination — a ratchet that only ever asks the model to stop slightly sooner than it already can (Fig. 1B, inset). **A brake on the ratchet.** A wrong answer earns zero regardless of length, so pressure toward brevity is exerted exclusively within the set of correct solutions; the moment shorter generation starts costing correctness, the median of correct lengths stops falling and the ratchet halts. **No cliff.** Verbose-but-correct rollouts retain partial credit rather than facing a discontinuity, preserving gradient for trajectories that solve the problem the long way — exploration is discounted, never extinguished. **Signal after mastery.** The structure interacts with group-relative estimation in a way that matters: under correctness-only reward, a group in which every rollout is correct is uninformative — all rewards equal, Eq. 1 assigns zero advantage, no gradient flows — precisely on the prompts the model has mastered. Under R3 the same group still carries signal, advantaging efficient-correct over verbose-correct rollouts, so the gradient continues to shape *when* the model stops after it has learned *what* to answer.

What the reward omits is as deliberate as what it contains: no auxiliary judge model, no token-level labels, no curriculum, no target length, no inference-time controller. The nearest neighbour, HAPO (Huang et al., 2026a), also adapts its target to the policy's own correct-solution history — but as a *per-problem minimum* driving a graded length price, so pressure ratchets toward the shortest length ever achieved and never relaxes; R3's policy-wide *median* can stop falling, which is the brake, and its credit is flat below the threshold, which is why no pressure exists inside the boundary. Recent analysis argues that trajectory-level *length* rewards are structurally insufficient, penalising essential exploration while rewarding partial redundancy, and concludes that token-level machinery is required (Jiang et al., 2026); R3 is trajectory-level but lies outside the length-penalty family that analysis concerns — credit is conditional and threshold-relative rather than a per-token price — a separation Proposition M.1 (Appendix M) makes formal — and whether one scalar suffices is precisely the empirical question of §3. Eq. 3 also makes a falsifiable prediction: once the termination term activates at $t = T_w$, median generation length should decline monotonically as the ratchet engages, plateauing only when correctness begins to bind (Proposition M.2). §3.2 tests this prediction directly. Algorithm 1 (Appendix C) summarises one training step.

2.4 Causal design

A reward that changes behaviour is not yet evidence of *which* of its terms changed it, so we train two controls. The *between-run* control, R1, is an identical GRPO run — same data, same 450 steps, same optimiser and rollout configuration — whose reward is correctness alone. Because our base models were produced by pure supervised distillation with no reinforcement-learning stage (Guo et al., 2025), R1 is itself a novel training condition, and the R3−R1 difference isolates the termination term from every effect of reinforcement-learning exposure. The *within-run* control is built into R3 itself: for $t < T_w$ the reward reduces exactly to R1's, so any behavioural discontinuity at step T_w=200 has a single available cause — the termination term switching on — inside one run with nothing else changed. We train R1 and R3 on each of three backbones — DeepSeek-R1-Distill Qwen-7B, Llama-8B, and Qwen-1.5B — on the GSM8K and MATH training splits, six runs in total; the untouched base checkpoints complete each backbone's triad (base, R1, R3). Hyperparameters and the reward implementation are listed in Table 4.

2.5 Evaluation protocol

Our central instrument is the *budgeted grid*: every model answers the same problems under hard limits $B \in \{300, 500, 1{,}000, 2{,}000, 4{,}000\}$ on the GSM8K test split, the MATH-500 test subset (Lightman et al., 2024) ("MATH" throughout), and ARC-Challenge as an out-of-domain generalisation check (300 problems each, fixed seed), with greedy decoding, zero-shot prompts identical across models — one template for open-ended problems, one for multiple choice — and one answer extractor shared by all models (templates in Appendix C; instruments in Appendix G). Uniformity is the point: a model that underperforms here does so under deployment's rules, not its authors' harness. Every cell is decomposed via Eq. 2, which is what separates a capability claim from a termination claim. Two published efficiency methods join the grid from their released checkpoints, LASER-DE (Liu et al., 2026a) and DECS (Jiang et al., 2026) — the external methods with released checkpoints on the

shared Qwen-7B backbone, keeping every grid comparison scale-matched. Training corpora differ — our models saw GSM8K and MATH; both baselines were trained on DeepScaleR — so we disclose overlap per benchmark (Appendix D.1) and note that ARC-Challenge appears in no method's corpus, making it the one fully corpus-fair comparison. Two stress tests use sampled decoding (temperature 0.6, top-p 0.95, n=8): pass@1–8 on MATH asks whether compression cost sampling diversity on the distribution where compression was trained, and the same protocol on AIME 2024 asks whether the learned stopping rule clips capability on problems harder than any seen in training; a third check, on HumanEval (Chen et al., 2021) at two greedy budgets (Appendix K), asks whether the same behaviour transfers to a structurally different task — the two benchmarks on which the metric discriminates, since GSM8K saturates at generous budgets and four-option multiple choice reaches pass@8 ceiling by chance. Wall-clock latency is reported only for the five Qwen-7B-backbone models, which ran on identical hardware; other backbones are compared in hardware-independent tokens (Appendix D.2). All claims concern the hard-budget regime ($B \leq 4{,}000$); unconstrained limits are a different regime, to which we return in §4.

3 Results

3.1 Budget failure is termination failure

Unless noted otherwise, this section reports the five models sharing the DeepSeek-R1-Distill-Qwen-7B backbone; the other backbones appear in §3.4 and Appendix I. Eq. 2 turns our question into three measurable quantities; Figure 2 measures all three.

Per-completion accuracy is budget-invariant; completion rate is not. Across the grid, $c(B)$ is the quantity that moves (Fig. 2a): every method's truncation falls with budget, and R3's is below one-fifth at 1,000 tokens — a level LASER-DE first reaches at 2,000, R1 and DECS only at 4,000, and the base model at no budget in the grid. Per-completion accuracy barely moves: wherever completions are plentiful it converges to 90–93% for base and R3 alike (Appendix D.5). Fig. 2b draws the consequence: at the most generous budget, overall accuracy spans fifteen points across five models whose per-completion accuracies sit within a 2.3-point band. Budgeted evaluation is measuring whether models finish, not how well they reason when they do; R3's two markers nearly coincide because it has no completion gap left. **Truncated traces already contain the answers.** If

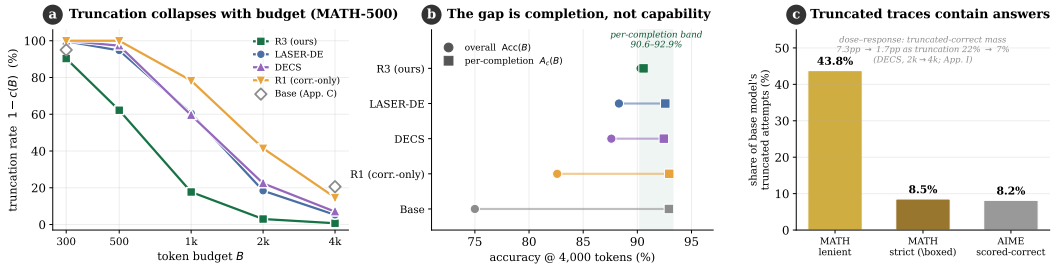


Figure 2: Budget failure is termination failure. (a) Truncation $1-c(B)$ is the budget-varying term of Eq. 2; R3 drives it to near zero earliest (full grid App. D). **(b)** At B=4,000, overall accuracy spans 15 points across the five Qwen-7B-backbone models while per-completion accuracy sits in a 2.3-point band — R3's markers nearly coincide. **(c)** Base-model truncated attempts already containing the answer (definitions in Appendix G.3); dose–response in Appendix N.

the deficit is termination, the missing accuracy should be recoverable inside the truncated traces — and it is (Fig. 2c): 43.8% of the base model's truncated MATH-500 generations contain the correct answer (a lenient upper bound), 8.5% in explicit `\boxed{}` form, and 8.2% of its truncated AIME attempts score correct under the shared extractor (Appendix G.3). The phenomenon tracks its cause: truncated-yet-correct mass shrinks in lock-step with truncation itself, from 7.3 points to 1.7 as DECS's truncation falls between the two largest budgets (Appendix N). Answer stability agrees from the inside: final answers fix by 1,000 tokens in 62% of problems while generation runs to a mean of 2,358 (Table 44). Verbatim traces — one per evidence tier, and one R3 failure — appear in Appendix H.

Budget-constrained failure is a failure of the decision to stop: reasoning that fits the budget is as accurate as it ever is, and the answers denied by truncation were already computed. Whether a reward can train that decision is the subject of §3.2.

Table 1: Correctness-only RL worsens tight budgets; adding termination calibration leads at every condition. Accuracy (%) under hard budgets (greedy decoding, uniform prompts and extractor, n=300; §2.5), with mean tokens, budget-coupling (Δtok, 2k→4k), and MATH latency. All rows share the Qwen-7B backbone; replication in Appendix I. **Bold** = column-best; R3 leads in all 30 budget–dataset cells (Tables 5–7). R1 shares R3's data, steps and optimiser — at tight budgets it falls *below the untrained base*. LASER-DE and DECS evaluated from released checkpoints; corpus disclosure Appendix D.1; latency scope §2.5.

	MATH			GSM8K			ARC-Challenge			Efficiency (MATH)		
Model	300	1k	4k	300	1k	4k	300	1k	4k	Δtok	tok@4k	time@4k
Base	8.7	31.8	75.0	41.0	77.3	86.0	26.0	49.0	63.3	+45.5%	2358	50.6 s
R1 (correctness-only)	7.0	41.1	82.6	16.3	86.3	93.0	4.0	79.0	87.0	+31.7%	2034	38.7 s
LASER-DE	9.7	64.9	88.3	34.0	90.7	94.0	12.7	74.0	83.3	+15.7%	1413	27.9 s
DECS	8.7	60.5	87.6	33.7	88.3	89.7	26.3	57.0	70.7	+20.4%	1515	29.5 s
R3 (ours)	**18.4**	**71.2**	**90.3**	**48.7**	**92.0**	**94.3**	**58.0**	**79.7**	**87.7**	**+6.0%**	**738**	**13.8 s**

3.2 One reward trains it away — causally

The design of §2.4 lets us do better than show that R3 works: Table 1 and Figure 3 identify *which term* of the reward acts, *when* it acts, and through *what mechanism* the standard recipe fails.

The termination term, isolated. R1 and R3 share data, steps, and optimiser; their difference is the termination signal alone. On MATH-500 that difference is worth $+11.4$, $+27.5$, $+30.1$, $+18.1$ and $+7.7$ points across the five budgets (Table 1; Appendix Table 6) — largest at 1,000 tokens, precisely where Fig. 2a locates the completion cliff, and smallest where truncation has nearly vanished. The effect is proportional to the failure it was designed to remove, which is what a causal account requires.

Why the standard recipe hurts: it moves the floor, not the tail. R1 is no neutral control — at tight budgets it falls *below the untrained base* (16.3% vs. 41.0% on GSM8K at 300 tokens; 4.0% vs. 26.0% on ARC). The mechanism is visible in the length distribution. Correctness-only RL *raises the median* trace (736 vs. 618 tokens, GSM8K at 4,000) while trimming the extreme tail (mean 904 vs. 1,218; Tables 43 and 8): it homogenises generation around a longer typical solution, so the short traces that once cleared tight budgets vanish — truncation at 300 tokens goes from 80.7% to 100% (Table 10). The same signature appears *inside the R3 run itself*: during its own correctness-only warm-up, mean length climbed from ∼1,300 toward 2,200 (Fig. 3a, grey) while the median held flat — two hundred steps of the field's default reward were already making deployment under constraint worse, before the termination term existed to reverse it. No mean-length metric would catch this failure; the mean *falls* while the floor rises. **Why R3 works: the ratchet binds, the gate brakes.** At step 200 the threshold — computed throughout warm-up but inert — begins to bind at 1,631 tokens (Fig. 3b). What follows is the mechanism of Eq. 3 operating in the open: the policy compresses beneath the threshold, the median of correct solutions falls, the threshold follows it down to 839, and median length declines near-monotonically to 699 ($r = -0.973$) — the trajectory the reward's structure predicted in §2.3. The median's near-perfect correlation against the mean's weaker one (-0.55) says the *typical* solution is what the gradient reshapes first, with the heavy tail lagging. At the rollout level the reward's target population migrates — full-credit correct rollouts rise from 12 of 20 (step 300) to 19 of 21 (step 450): one remaining path to full reward, short *and* correct (Appendix C). The brake holds meanwhile: reward recovers within-phase from the re-scoring dip ($0.52 \rightarrow 0.64$, phase means), entropy settles at 0.26, and ceiling-clipping falls to 2.4% from a post-activation peak of 25.7%.

Against specialised methods, in their own currency. R3 exceeds LASER-DE and DECS in all 30 budget–dataset comparisons (Table 1), at roughly half their generation (738 vs. 1,413–1,515 mean tokens at 4,000) and about twice their speed (13.8 vs. 27.9–29.5 s per problem; scope per §2.5). At the most generous budget the margin peaks where no method holds a corpus advantage: ARC-Challenge, $+17.0$ points over DECS. Under the uniform budgeted protocol the baselines themselves reorder — DECS trails the simpler LASER-DE at every budget $\geq$1,000 (12 of 15 cells overall) — a regime observation, not a quality claim, developed in §4.[1]

[1] Both baselines evaluated from released checkpoints; corpus disclosure and full comparison in Appendices D and N.

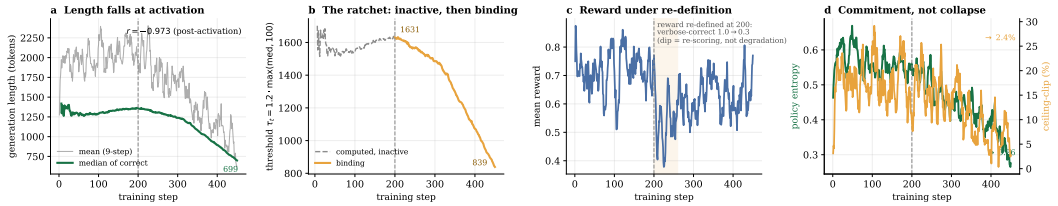


Figure 3: Termination behaviour changes at the step the signal activates — the within-run control. One R3 run's training log; dashed line = activation (T_w=200). **(a)** Median correct-solution length is flat through warm-up (while the *mean* climbs), then falls near-monotonically to 699 ($r = -0.973$). **(b)** The threshold, computed throughout, binds only at activation: 1,631 → 839. **(c)** The reward *definition* changes at step 200 (verbose-correct 1.0→0.3); the dip is re-scoring, not degradation, and recovers within-phase. **(d)** Entropy declines to 0.26 and ceiling-clipping to 2.4%: commitment, not collapse — tested in §3.4.

A single trajectory-level reward, causally isolated against the correctness-only recipe it extends, converts termination failure into expressed accuracy — and the recipe it extends, left alone, makes that failure worse. What kind of stopping rule was learned — compression or calibration — is the question §3.3 answers.

3.3 Calibration, not compression

Any length pressure produces shorter traces; the question is what *rule* was learned. A compressed model has a new, smaller characteristic length. A calibrated one has no characteristic length at all — it stops when its answer is ready, wherever that falls. Three signatures separate the two, and R3 exhibits all three.

The stopping point ignores the budget. On GSM8K, R3's mean generation is 353 tokens whether the ceiling is 1,000, 2,000 or 4,000; its median is 333 from 500 tokens upward (Table 43; means in Table 8). Every other model's stopping point chases the ceiling: when the budget doubles from 2,000 to 4,000 on MATH-500, generation grows +45.5% (Base), +31.7% (R1), +20.4% (DECS) and +15.7% (LASER-DE) — against R3's +6.0% (Table 1, Δtok). This *coupling spectrum* is Corollary 2's content, and it extends beyond our grid: in its native 16k-token regime DECS's own reported average is 3,968 tokens (Appendix N). The spectrum also explains the baseline inversion of §3.2: at the two most generous budgets, DECS generates slightly *more* than LASER-DE for slightly less accuracy, because being trained shorter is not the same as knowing when to stop.

The stopping point tracks the problem. Within MATH-500, R3 spends 424 tokens on the easiest stratum and 785 on the hardest — a 1.9× range (stratification in Appendix O). LASER-DE and DECS show the same 1.9× ratio, but from double the operating point (939→1,816 and 1,017→1,958): difficulty-adaptivity in slope is common; R3's distinction is the intercept it adapts *from*. Across datasets the rule extrapolates — median generation rises 333 → 576 → 1,175 over GSM8K → MATH-500 → AIME — a 3.5× span ending on problems harder than anything in the training corpus. A compressed model would carry its training-set length to AIME; a calibrated one earns a longer one there, and does.

What little growth remains is composition, not budget-seeking. R3's MATH-500 mean does drift from 624 to 738 as budgets rise — but its *median* is fixed at 576 from 1,000 tokens up, while truncation falls 17.7% → 0.7% (Tables 11, 43, and 13). The mean moves because previously-truncated hard problems now complete at their natural length, not because typical solutions inflate to fill the room. And the rule is a property of the policy, not the decoder: under temperature-0.6 sampling the median is 622 against greedy's 576 (Table 28).

The learned rule is calibration: budget-invariant where capability suffices, longer exactly where problems demand it, and stable across decoding. One suspicion remains — that this discipline was purchased with the model's depth or diversity — and §3.4 is its direct test.

3.4 Nothing measurable is lost

The discipline of §3.3 invites the objection DECS formalises (Jiang et al., 2026): length pressure collapses exploration by penalising the very tokens search requires — and R3's own telemetry supplies the prima facie evidence, entropy falling 0.57 → 0.26 (Fig. 3d). If that is collapse, it must

Table 2: Parity where it must hold, gains where it can. Sampled evaluation (temperature 0.6, top-p 0.95, n=8; 4,000-token budget) on the DeepSeek-R1-Qwen-7B backbone. MATH-500 tests in-distribution diversity; AIME 2024 tests out-of-distribution ceiling (n=30: the paired bootstrap CI on Δavg@1 spans zero, so we claim parity, never superiority). Full statistics in Appendices E and F.

	MATH-500 (in-distribution)				AIME 2024 (OOD)			
Model	pass@1	pass@8	tok	trunc	avg@1	pass@8	med. tok	trunc
Base	76.4	90.0	2,331	22.5%	23.3	43.3	4,000	81.7%
R3 (ours)	80.6	93.0	786	1.3%	20.4	43.3	1,175	20.0%

surface as degraded multi-sample performance: first in distribution, then as a clipped ceiling on problems needing long thought. We test both.

In distribution, diversity survives sampling. On MATH-500 under temperature-0.6 sampling, R3's pass@k exceeds the base model's at every k from 1 to 8 (80.6/86.9/90.5/93.0 vs. 76.4/82.9/86.9/90.0; Fig. 6a), with paired bootstrap intervals excluding zero throughout (Appendix E). The gap has a mechanism, and it is the familiar one: restrict both models to problems where all eight samples terminate, and pass@8 converges to $\sim$95% for each — the base model's deficit under sampling is completion failure again, now paying per sample. The efficiency ledger compounds it: 6,767 tokens per solved problem against 20,727. Entropy 0.26, then, was never the signature of a collapsed policy; it was commitment to termination — Corollary 3, discharged by direct measurement. A static text-diversity proxy flags *every* model in the grid — R3 least (-22.3% vs. the base's own -46.7%; Table 27) — which is why functional pass@k, not textual similarity, is ground truth here.

Out of distribution, the ceiling is intact. AIME 2024 poses problems harder than anything in the training corpus, and n=30 makes greedy scoring $\pm$6-point noise — so we sample (Table 2). The result is parity with the books balanced: pass@8 identical at 43.3%, the paired outcome exactly symmetric (10 problems solved by both, 3 uniquely each), and Δavg@1 of -2.9 points whose CI $[-11.7, +5.8]$ spans zero — we therefore claim *matches*, never exceeds. What parity costs is the finding: R3 reaches it at a median of 1,175 tokens against the base model's ceiling-pinned 4,000 (81.7% truncation — on AIME the base model has effectively no stopping rule at all; R3's own residual is disclosed in Table 33: 20% truncation remains, and one of its 48 truncated attempts still contains a scored answer — calibration collapses the phenomenon, it does not eliminate it). And the classic failure mode of length-penalised models — abandoning hard problems early to harvest the brevity reward — is absent by direct measurement: R3 spends 1,752 tokens on problems it never solves, essentially identical to the 1,704 on those it solves (Fig. 6b). The stopping rule is calibrated to readiness, not to difficulty-avoidance.[2]

The ordering is a property of the reward, not the backbone. Retraining the full triad on DeepSeek-R1-Llama-8B and DeepSeek-R1-Qwen-1.5B reproduces R3>R1>Base at the generous budget in eight of nine backbone–dataset cells — cleanly on both mathematics benchmarks — with one exception on ARC-Challenge for Llama-8B, where R1 (90.0) edges R3 (89.7) by a single problem and both clear the base (Appendix I). The starkest single cell in the grid is this backbone's: on ARC-Challenge at 300 tokens the base scores 3.3% — below four-option chance, because almost nothing it starts is finished — correctness-only training moves it to 4.0%, and R3 to 83.7% (Table 36): termination, not capability, at its most extreme. Llama-8B_R3 also marks the frontier's edge, over-compressing on MATH even as its raw accuracy leads the base at every budget (Appendix I) — the calibration factor wants per-backbone tuning, taken up in §4.

Nothing measurable was traded: diversity is higher where compression was trained, the ceiling holds where problems are hardest, and effort goes where difficulty demands it. With the cure characterised and its costs audited, what remains is to situate the result.

The axis also composes: applied to the strongest token-level baseline's own checkpoint, the same reward flattens its median generation length across every budget, though a tail of hard MATH problems remains only partially decoupled (Appendix J).

[2]The base model's 92.6% per-completed-attempt accuracy under sampling is a selection artifact of its 77% completion rate and should not be read as superior conditional capability (Appendix Table 24).

4 DISCUSSION

Under deployment token budgets, reasoning failure is termination failure — not capability failure — and termination is trainable by a single self-calibrated reward across every condition examined here. The finding positions within a landscape of efficiency mechanisms spanning four axes. Length-penalised training governs *how much* a model generates (Arora & Zanette, 2025; Liu et al., 2026a; Huang et al., 2026a); inference-time control governs how much it is *allowed* to generate (Muennighoff et al., 2025; Aggarwal & Welleck, 2025); routing governs *whether and in which mode* it reasons (Fang et al., 2026; Zhang et al., 2025; Liu et al., 2026b). The present work isolates a fourth axis the others leave implicit: *when generation is done* — and supplies its measurement instrument, the coupling spectrum, on which the field's methods spread from $+45.5\%$ to $+6.0\%$ token growth per budget doubling (Corollary 2). This axis reorders the others under constraint: rankings taken at generous limits do not survive deployment budgets, because learning to be brief and learning when to stop are different objectives, and only the second is budget-robust.

Our relation to token-level machinery deserves precision, because theory and our measurements only appear to disagree. DECS's Theorem 2 shows that a sequence-level length penalty, under group-relative training, leaves the first redundant token unpenalised — the policy cannot learn to stop exactly at the reasoning boundary — motivating token-level credit with judge and curriculum (Jiang et al., 2026). R3 is sequence-level but outside the quantified family — credit is conditional and threshold-relative, a price on the *decision*, not on tokens — so the necessity claim does not bind it; Proposition M.1 (Appendix M) shows the threshold-conditional form expresses the boundary structure unavailable inside that family, and empirically one scalar sufficed in this regime (Appendix N). The constraint was never reward granularity; it was the length-penalty family itself. Two concessions scope this: at unconstrained budgets the verbose models recover (DECS averages $\sim$3,968 tokens natively and holds a per-completion edge, 92.4 vs. 90.6), and the regime choice alone inverts published orderings — evidence the field measures one regime and deploys into another, not a deficiency of either method (Appendix N).

One question this work raises is already answered: because R3 is a drop-in scalar, we tested whether it composes with token-level credit by applying it to the released DECS checkpoint, and it does — one reward term flattens DECS's budget-coupling to exactly 0.0% median-based on every dataset — though a matched control shows most of that coupling gain is generic RLVR exposure, not the reward term specifically — while accuracy gains at tight budgets are attributable to the termination term alone (Appendix J); on HumanEval Chen et al. (2021), evaluated on both checkpoints unchanged, the same accuracy, truncation, and speed advantage generalises beyond mathematical reasoning, though the budget-invariance signature does not (Appendix K). Two directions remain open. Per-backbone tuning of the calibration factor and window would turn Llama-8B's over-compression frontier into a dial on the accuracy–latency trade-off; and the latent-answer phenomenon poses a representation-level question our training-process evidence cannot answer: where, internally, an answer becomes ready before generation stops.

Several limitations constrain these findings. The compression–capability frontier is real but smaller than an unmatched comparison suggests: on the 183 problems both Llama-8B arms complete, the gap is 3.8, not 8.5, points (85.8 vs. 89.9; the unmatched 81.4 mixes in the selection effect Appendix G.3's AIME footnote already discounts) — yet raw accuracy still leads at every budget, indicating the $1.2\times$ factor and window W require per-backbone tuning (sensitivity check in Appendix L); the remaining reward constants (the 0.3 partial credit, 100-token floor, and warm-up T_w) were fixed a priori and not swept. The within-run control timestamps the mechanism and the three-backbone replication reproduces the ordering. Individual margins over the strongest baseline at $B{=}4{,}000$ (0.3–4.4 points at $n{=}300$) lie within sampling noise, so the dominance claim rests on the uniform sign across all 30 cells and the large tight-budget separations, not on any single generous-budget cell; and the compression trajectory had not fully plateaued at the fixed 450-step budget matched to R1. AIME's $n{=}30$ yields wide intervals (Δavg@1 CI $[-11.7, +5.8]$), supporting parity and nothing stronger. Finally, all results concern distilled 1.5–8B models under budgets $\leq$4,000 with math-centric training corpora; frontier-scale models and the unconstrained regime remain untested. The composition experiment's matched R1 control (Appendix J) isolates the termination term for the accuracy claim, but coupling improvement under R1 shows generic RLVR exposure — not the reward specifically — drives most of DECS's own coupling reduction.

Reproducibility Statement

All training code (R1 and R3, both reward implementations and the GRPO configuration), the full evaluation harness (prompt templates, answer extraction, truncation classification), and the calibration-factor sensitivity analysis are included in the anonymous supplementary material. Hyperparameters for every training run are listed in Table 4 (Appendix C); the reward implementation is stated exactly in Eq. 3 and summarised in Algorithm 1. The evaluation protocol — prompts, decoder settings, and the shared answer extractor — is specified in full in Appendix C and Appendix G, and applied identically across all models to keep comparisons uniform. Training and evaluation data are public benchmarks (GSM8K, MATH-500, ARC-Challenge, AIME 2024, HumanEval); corpus overlap between our training data and the external baselines' training data is disclosed in Appendix D.1. All theoretical claims (Propositions M.1 and M.2) are stated with complete proofs in Appendix M. Trained model checkpoints, Code, All Models diagnostic and evaluation raw dataset will be released publicly and linked to the paper directly upon acceptance; anonymized checkpoint identifiers are omitted from the supplementary material to preserve double-blind review.

References

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. In *Conference on Language Modeling (COLM)*, 2025. arXiv:2503.04697.

Daman Arora and Andrea Zanette. Training language models to reason efficiently. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=AiZxn84Wdo.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, et al. Do NOT think that much for 2+3=? On the overthinking of o1-like LLMs. *arXiv preprint arXiv:2412.21187*, 2024.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

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

Muzhi Dai, Chenxu Yang, and Qingyi Si. S-grpo: Early exit via reinforcement learning in reasoning models. *Advances in Neural Information Processing Systems*, 38:48178–48204, 2026.

Razvan-Gabriel Dumitru, Darius Peteleaza, Vikas Yadav, and Liangming Pan. Conciserl: Conciseness-guided reinforcement learning for efficient reasoning models. In *EMNLP (Findings)*, pp. 17099–17123, 2025.

Gongfan Fang, Xinyin Ma, and Xinchao Wang. Thinkless: Llm learns when to think. *Advances in neural information processing systems*, 38:151268–151295, 2026.

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.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. Token-budget-aware llm reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 24842–24855, 2025.

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 *NeurIPS Datasets and Benchmarks Track*, 2021.

Bairu Hou, Yang Zhang, Jiabao Ji, Yujian Liu, Kaizhi Qian, Jacob Andreas, and Shiyu Chang. ThinkPrune: Pruning long chain-of-thought of LLMs via reinforcement learning. *arXiv preprint arXiv:2504.01296*, 2025.

Chengyu Huang, Zhengxin Zhang, and Claire Cardie. Hapo: Training language models to reason concisely via history-aware policy optimization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 31122–31130, 2026a.

Zixuan Huang, Xin Xia, Yuxi Ren, Jianbin Zheng, Xuanda Wang, Zhixia Zhang, Hongyan Xie, Songshi Liang, Zehao Chen, Xuefeng Xiao, Fuzhen Zhuang, Jianxin Li, deqing wang, and Yikun Ban. Does your reasoning model implicitly know when to stop thinking? In *Forty-third International Conference on Machine Learning*, 2026b. URL https://openreview.net/forum?id=xF6klfHhO8.

Aaron Jaech, Adam Kalai, Adam Lerer, et al. OpenAI o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Shuyang Jiang, Yusheng Liao, Ya Zhang, Yanfeng Wang, and Yu Wang. Overthinking reduction with decoupled rewards and curriculum data scheduling. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=kdeiRledV6.

Yu Kang, Xianghui Sun, Liangyu Chen, and Wei Zou. C3ot: Generating shorter chain-of-thought without compromising effectiveness. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pp. 24312–24320, 2025.

Kimi Team. Kimi k1.5: Scaling reinforcement learning with LLMs. *arXiv preprint arXiv:2501.12599*, 2025.

Zheng Li, Qingxiu Dong, Jingyuan Ma, Di Zhang, Kai Jia, and Zhifang Sui. Selfbudgeter: Adaptive token allocation for efficient llm reasoning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 21135–21156, 2026.

Tian Liang, Wenxiang Jiao, Zhiwei He, Jiahao Xu, Haitao Mi, and Dong Yu. Deepcompress: A dual reward strategy for dynamically exploring and compressing reasoning chains. In *International Conference on Learning Representations*, volume 2026, pp. 87309–87323, 2026.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, Jan Kautz, and Pavlo Molchanov. DLER: Doing length penalty right — incentivizing more intelligence per token via reinforcement learning. *arXiv preprint arXiv:2510.15110*, 2025.

Wei Liu, Ruochen Zhou, Yiyun Deng, Yuzhen Huang, Junteng Liu, Yuntian Deng, Yizhe Zhang, and Junxian He. Learn to reason efficiently with adaptive length-based reward shaping. In *International Conference on Learning Representations*, volume 2026, pp. 43279–43308, 2026a.

Xiang Liu, Xuming Hu, Xiaowen Chu, and Eunsol Choi. Diffadapt: Difficulty-adaptive reasoning for token-efficient llm inference. In *International Conference on Learning Representations*, volume 2026, pp. 31640–31666, 2026b.

Haotian Luo, Haiying He, Yibo Wang, Shiwei Liu, Wei Li, Xiaochun Cao, Dacheng Tao, Naiqiang Tan, and Li Shen. O1-pruner: Length-harmonizing fine-tuning for o1-like reasoning pruning. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 14242–14257, 2026.

Xinyin Ma, Guangnian Wan, Runpeng Yu, Gongfan Fang, and Xinchao Wang. Cot-valve: Length-compressible chain-of-thought tuning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6025–6035, 2025.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20286–20332, 2025.

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. *arXiv preprint arXiv:2402.03300*, 2024.

Yi Shen, Jian Zhang, Jieyun Huang, Shuming Shi, Wenjing Zhang, Jiangze Yan, Ning Wang, Kai Wang, Zhaoxiang Liu, and Shiguo Lian. Dast: Difficulty-adaptive slow-thinking for large reasoning models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pp. 2322–2331, 2025.

Jinyan Su, Jennifer Healey, Preslav Nakov, and Claire Cardie. Between underthinking and overthinking: An empirical study of reasoning length and correctness in LLMs. *arXiv preprint arXiv:2505.00127*, 2025.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Hanjie Chen, et al. Stop overthinking: A survey on efficient reasoning for large language models. *arXiv preprint arXiv:2503.16419*, 2025.

Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. Tokenskip: Controllable chain-of-thought compression in llms. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 3351–3363, 2025.

Yang Xiao, Jiashuo Wang, Ruifeng Yuan, Chunpu Xu, Kaishuai Xu, Wenjie Li, and Pengfei Liu. Limopro: Reasoning refinement for efficient and effective test-time scaling. *Advances in Neural Information Processing Systems*, 38:157001–157026, 2026.

Silei Xu, Wenhao Xie, Lingxiao Zhao, and Pengcheng He. Chain of draft: Thinking faster by writing less. *arXiv preprint arXiv:2502.18600*, 2025.

Shiming Yang, Yuxuan Tong, Xinyao Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=OLodUbcWjB.

Jingyang Yi, Jiazheng Wang, and Sida Li. Shorterbetter: Guiding reasoning models to find optimal inference length for efficient reasoning. *Advances in Neural Information Processing Systems*, 38: 39011–39043, 2026.

Chuhuai Yue, Chengqi Dong, Yinan Gao, Hang He, Jiajun Chai, Wei Lin, and Guojun Yin. Promoting efficient reasoning with verifiable stepwise reward. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 34530–34538, 2026.

Zhipeng Zeng, Xiaoyang Huang, Bowen Li, Hao Zhang, and Zhijie Deng. Done is better than perfect: Unlocking efficient reasoning by structured multi-turn decomposition. In *2nd AI for Math Workshop @ ICML 2025*, 2025.

Jiajie Zhang, Nianyi Lin, Lei Hou, Ling Feng, and Juanzi Li. Adaptthink: Reasoning models can learn when to think. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 3716–3730, 2025.

APPENDIX

APPENDIX CONTENTS

A AI Use Statement **14**

B Extended related work **14**

C Training details **16**

D The full budgeted grid **18**

D.1 Corpus disclosure 18

D.2 Protocol notes and known measurement caveats 18

D.3 Full tables 18

D.4 Coupling derivations 24

D.5 Truncation decomposition 24

E Sampled evaluation: pass@k statistics **25**

F AIME 2024: protocol, statistics, and derivations **27**

G Measurement stack **29**

G.1 Truncation classification 29

G.2 Answer extraction 29

G.3 Latent-answer scans 30

G.4 Answer-stability metric 30

H Qualitative traces **30**

I Cross-backbone replication and the calibration frontier **32**

J Composition with token-level credit **33**

K Generalization to code: HumanEval **34**

L Calibration-factor sensitivity **35**

M Expressibility and idealised dynamics of the termination reward **35**

N Relation to DECS: scope, arithmetic, and regime **37**

O Adaptive allocation and stability **38**

A AI USE STATEMENT

Large language models (LLMs) were employed exclusively to improve the quality of written English and for stylistic refinement, with attentive supervision by the authors. No scientific content, experimental design, data analysis, or any intellectual contribution were generated through LLMs, nor did such assistance influence any reported results.

B EXTENDED RELATED WORK

The overthinking phenomenon. That reasoning-specialised models generate far beyond what their answers require was documented almost immediately after the o1-class recipe emerged (Chen et al., 2024), and has since been surveyed systematically (Sui et al., 2025). Analyses of the long-chain-of-thought recipe itself trace how reinforcement learning shapes trace length and where additional length stops paying off (Yang et al., 2025), while Su et al. (2025) show the dual failure — under- and over-thinking — is difficulty-dependent. These works characterise the pathology at generous limits; none asks what the pathology costs when generation is cut off, which is where §3.1 begins.

Length-shaped reinforcement learning. The largest family prices length inside the reward. Fixed or scheduled penalties appear in Kimi k1.5 (Kimi Team, 2025) and in dedicated recipes that trade a controlled amount of accuracy for brevity (Arora & Zanette, 2025); cosine-shaped length rewards (Yang et al., 2025) smooth the price. Difficulty-aware variants set dynamic targets — LASER-DE conditions the target on problem difficulty (Liu et al., 2026a), DAST budgets by difficulty (Shen et al., 2025), and DeepCompress flips the sign, rewarding *longer* traces on problems the model currently finds hard (Liang et al., 2026). HAPO conditions a graded (cosine-shaped) length price on the per-problem *minimum* of previously correct lengths (Huang et al., 2026a) — closest to R3 in spirit, and worth differentiating precisely: HAPO's price is strictly decreasing on correct rollouts, it prices the length of incorrect ones, its per-problem state requires repeated encounters with each problem (multi-epoch training), and its own ablation shows that substituting a central-tendency statistic into that graded price *worsens* the accuracy trade. R3 differs in shape, not statistic: credit is flat below a threshold on a single policy-wide window, a wrong rollout's length receives no gradient at all, and no problem need be seen twice (§2.3); ShorterBetter targets the shortest sampled correct length (Yi et al., 2026); O1-Pruner constrains an RL-style fine-tune to a pre-sampled accuracy baseline (Luo et al., 2026); ConciseRL replaces the length term with an LLM-judged conciseness score (Dumitru et al., 2025). Two lines are closest to our regime in spirit. ThinkPrune trains *under* a hard limit — unfinished rollouts score zero — and tightens the limit iteratively (Hou et al., 2025); DLER argues the simple truncation penalty fails only because of RL optimisation pathologies, and repairs the optimiser rather than the reward (Liu et al., 2025). Both, however, optimise and evaluate against their own training limits; neither evaluates against *deployment* budgets the model was never trained at, nor decomposes failure into completion and capability. At the frontier of this family, DECS argues trajectory-level length rewards are structurally insufficient and moves to token-level credit with a judge and curriculum (Jiang et al., 2026); our theoretical and empirical relation to that argument is developed in §4 and Appendix N.

Supervised compression. A parallel family compresses via data rather than reward: TokenSkip prunes low-salience tokens and fine-tunes on the compressed traces with a controllable ratio (Xia et al., 2025); CoT-Valve constructs an elastic parameter direction so one model emits chains of tunable length (Ma et al., 2025); C3oT trains on paired long–short chains conditioned on a compression instruction (Kang et al., 2025); LIMOPro refines distilled traces so that test-time scaling starts from more efficient exemplars (Xiao et al., 2026). These methods inherit the lengths present in their training data; whether the resulting stopping behaviour is budget-invariant is not measured.

Trained early exit. A further line trains models to exit mid-reasoning rather than to be shorter overall: S-GRPO truncates a full rollout at several serial positions, forces an answer at each, and rewards earlier-correct exits on a decaying schedule (Dai et al., 2026); VSRM assigns verifiable stepwise rewards at segment boundaries (Yue et al., 2026); MinD cold-starts explicit segment separators via SFT before RL (Zeng et al., 2025). These are the nearest neighbours to training a stopping decision, which makes the differences the point: their exit incentive is a graded price on *position* — earlier is always better — rather than credit conditioned on a threshold set by the policy's

Table 3: Efficient-reasoning methods by mechanism and regime. "Stage": where the intervention lives. "Extra": auxiliary machinery beyond a scalar reward (judge or reward model, token labels, probes, curricula, mode data, decode-time controllers). Final column: does the method's evaluation include *enforced* generation cutoffs at deployment scale ($\leq$1,000 tokens)? ✓ = yes; (✓) = own-model or own-limit only (footnotes); ✗ = no. Assessments reflect the cited papers' reported protocols.

Method	Core mechanism	Stage	Extra machinery	Enforced $\leq$1k?
LENGTH-SHAPED RL				
Kimi k1.5 (Kimi Team, 2025)	length penalty in RL	train	—	✗
Arora & Zanette (Arora & Zanette, 2025)	accuracy-constrained length RL	train	—	✗
LASER-DE (Liu et al., 2026a)	difficulty-aware dynamic target	train	—	✗
DAST (Shen et al., 2025)	difficulty-adaptive budget reward	train	—	✗
DeepCompress (Liang et al., 2026)	dual reward: short-easy, long-hard	train	online difficulty split	✗
HAPO (Huang et al., 2026a)	per-problem correct-length history	train	history state	✗
ShorterBetter (Yi et al., 2026)	shortest-correct-sample target	train	—	✗
O1-Pruner (Luo et al., 2026)	length-harmonised fine-tuning	train	pre-sampling	✗
ConciseRL (Dumitru et al., 2025)	LLM-judged conciseness reward	train	judge model	✗
ThinkPrune (Hou et al., 2025)	zero reward past limit, iterated	train	limit schedule	(✓)[a]
DLER (Liu et al., 2025)	truncation penalty, optimiser repairs	train	—	✗[b]
DECS (Jiang et al., 2026)	token-level credit assignment	train	judge + curriculum	(✓)[c]
SUPERVISED COMPRESSION				
TokenSkip (Xia et al., 2025)	token-pruned SFT, ratio-conditioned	train	pruning labels	✗
CoT-Valve (Ma et al., 2025)	elastic length parameter direction	train	paired-length data	✗
C3oT (Kang et al., 2025)	long–short conditioned SFT	train	compressor	✗
LIMOPro (Xiao et al., 2026)	refined traces for test-time scaling	train	refinement pipeline	✗
TRAINED EARLY EXIT				
S-GRPO (Dai et al., 2026)	serial-position decaying exit reward	train	multi-exit rollouts	✗
VSRM (Yue et al., 2026)	verifiable stepwise exit reward	train	segment scoring	✗
MinD (Zeng et al., 2025)	separator-structured multi-turn CoT	train	SFT cold-start	✗
PROMPT / INFERENCE-TIME CONTROL				
TALE (Han et al., 2025)	estimated budget in prompt	inference[d]	budget estimator	✗[e]
Chain-of-Draft (Xu et al., 2025)	minimalist-draft prompting	inference	—	✗
s1 (Muennighoff et al., 2025)	decode-time budget forcing	inference	controller	(✓)[f]
L1 (Aggarwal & Welleck, 2025)	prompt-target length RL	train+inf.	—	(✓)[e]
MODE ROUTING				
Thinkless (Fang et al., 2026)	`<short>`/`<think>` via DeGRPO	train	control tokens	✗
AdaptThink (Zhang et al., 2025)	constrained NoThinking selection	train	mode data	✗
SelfBudgeter (Li et al., 2026)	self-predicted budget + guided GRPO	train	budget head/format	✗[e]
DiffAdapt (Liu et al., 2026b)	probe-routed strategy selection	inference	probe + thresholds	✗
SAMPLING-LEVEL EXPLORATION				
SAGE-RL (Huang et al., 2026b)	confidence-guided rollout mixing	train	search procedure	(✗)[g]
R3 (ours)	**termination-calibrated reward**	**train**	**—**	✓

[a] Trains under iterated hard limits (4k→3k→2k); evaluation is unconstrained apart from a budget-forcing analysis; no cross-method deployment grid. [b] Trains under truncation limits; no cross-method comparison under enforced deployment budgets. [c] Sweeps generation limits from 2,048 upward on its own model only. [d] A post-training variant internalises the estimate. [e] Prompted or self-predicted budgets are targets, not enforced cutoffs; overshoot is scored, not truncated. [f] Forced termination on its own model for scaling curves; not a method comparison under fixed budgets. [g] Evaluated only at a generous 32,768-token ceiling; no enforced or truncating budget of any kind is tested.

own competence, so no brake exists when brevity begins to cost correctness; each requires auxiliary scaffolding (segmentation, multi-exit rollouts, process-level scoring) where R3 is one scalar; and none is evaluated under enforced deployment budgets, so whether the learned exits are budget-invariant — the signature separating calibration from compression (§3.3) — is unmeasured.

Prompt- and inference-time control. A fourth family leaves weights alone or adds decode-time machinery. TALE estimates a per-problem token budget and injects it into the prompt (with a post-training variant that internalises the estimate) (Han et al., 2025); Chain-of-Draft prompts for minimalist intermediate steps, reaching as little as 7.6% of chain-of-thought tokens on some tasks (Xu et al., 2025). s1's budget forcing terminates or extends thinking at decode time — appending "Wait" to lengthen — yielding controllable test-time scaling (Muennighoff et al., 2025).

L1 trains with length-conditioned RL so that a target stated in the prompt is obeyed at inference (Aggarwal & Welleck, 2025). All of these control *how much room* the model gets or is asked to use; none

trains the model's own decision that its answer is ready, and soft prompt-targets are not enforced cutoffs — a model can and does overshoot them, which is precisely the regime our grid measures.

Mode routing. The fifth family decides per problem whether to think at all. Thinkless learns a <short>/<think> control token under a decoupled GRPO objective (Fang et al., 2026); AdaptThink optimises a constrained objective that prefers a NoThinking mode when accuracy permits (Zhang et al., 2025); SelfBudgeter first predicts its own budget, then is trained with budget-guided GRPO to respect it (Li et al., 2026); DiffAdapt routes among Easy/Normal/Hard inference strategies via a probe on hidden states (Liu et al., 2026b). Routing is binary or few-way and often needs auxiliary infrastructure (probes, thresholds, mode data); R3's stopping rule is continuous, internal to the policy, and needs nothing at inference.

The gap, made explicit. Table 3 organises these methods by mechanism, stage, and auxiliary requirements. The final column records whether the method's own evaluation includes hard, *enforced* generation cutoffs at deployment scale ($\leq$1,000 tokens) — the regime of §2.5. Partial entries exist: ThinkPrune and s1 evaluate at their own trained or forced limits, DECS sweeps its own model from 2,048 upward, and L1 obeys prompted targets that are not enforced. To our knowledge, no prior work evaluates *multiple methods under one uniform enforced-budget protocol*, decomposes budgeted accuracy into completion and per-completion terms (Eq. 2), or measures the latent-answer mass inside truncated traces — the three instruments on which this paper's diagnosis rests.

Sampling-level exploration. A fifth axis leaves both reward and decoding constraints untouched and instead reshapes which rollouts training sees: SAGE (Huang et al., 2026b) finds that reasoning models assign high cumulative confidence to short, correct reasoning paths that greedy or random decoding rarely surfaces, and SAGE-RL mixes confidence-guided rollouts into standard GRPO/GSPO groups so training itself is nudged toward them — empirical support, via a different mechanism and metric (their RFCS vs. our latent-answer scans, Appendix G.3), for the diagnosis §3.1 makes: models compute answers earlier than they express them. Unlike R3, this operates entirely through rollout selection, modifying neither the reward nor a decoding constraint; their evaluation is also unconstrained (32,768-token ceiling), so whether SAGE-RL's discovered stopping behavior survives an enforced budget — the regime §2.5 tests — is not addressed by their results.

C Training Details

All runs use TRL's `GRPOTrainer` (TRL 0.16.1, transformers 4.51.3) with vLLM rollouts on two H200 GPUs. The three backbones are DeepSeek-R1-Distill Qwen-7B, Llama-8B, and Qwen-1.5B; R1 and R3 runs are identical except for the reward. Training data is 7,500 GSM8K + 7,500 MATH training problems (multiple-choice MATH items filtered), shuffled with seed 42 and consumed in under one epoch.

Table 4: Hyperparameters used for R1 and R3.

● General Training Setup			
Learning rate	1×10^{-6}	Steps	450
Group size G	8	Batch $\times$ grad. accum.	4×8 (256 seq/step)
Max completion length	4,096 tok.	Max prompt length	512 tok.
Temperature / top-p	1.0 / 1.0	KL coef. β / clip ϵ	0 / 0.2
● R3-Specific Settings			
Window W	5,000 rollouts	Warm-up T_w	200 steps
Threshold factor	1.2	Median floor	100 tok.
Partial credit	0.3	Precision	bf16, grad. ckpt.

Objective. GRPO optimises the standard clipped surrogate with group-normalised advantages (Eq. 1); with KL coefficient β=0 our objective is

$$\mathcal{J}(\theta) = \mathbb{E}\Big[\tfrac{1}{G}\sum_{i=1}^{G}\tfrac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\big(\rho_{i,t}\hat{A}_i,\ \mathrm{clip}(\rho_{i,t},1{-}\epsilon,1{+}\epsilon)\hat{A}_i\big)\Big], \quad \rho_{i,t} = \tfrac{\pi_\theta(o_{i,t}|x,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|x,o_{i,<t})}, \tag{4}$$

with ϵ=0.2. Everything this paper changes enters through $r(\cdot)$ in Eq. 3; the objective, optimiser and architecture are TRL defaults.

Two warm-ups, one active. The trainer's learning-rate warm-up is *zero* (`warmup_steps=0`): the LR is constant from step 1. The warm-up of §2.4 is a property of the *reward*: for $t < T_w$=200, Eq. 3a reduces exactly to correctness-only. Any behavioural discontinuity at step 200 therefore cannot be an optimiser-schedule artefact — nothing in the optimiser changes at that step.

Reward implementation. Lengths are counted with the base model's tokenizer. Answers are extracted from the last balanced `\boxed{}` with a numeric fallback, and verified at tolerance 10^{-2}. The window $\mathcal{W}$ receives only correct-rollout lengths; the running median (floored at 100) is recomputed each step, and the reward state — including the full window — is checkpointed alongside model weights, enabling exact resumption of the ratchet across restarts. Algorithm 1 summarises one training step.

Algorithm 1 One R3 training step

Require: policy π_θ; window $\mathcal{W}$ of last W correct lengths; warm-up T_w
1: sample $\{o_i\}_{i=1}^{G} \sim \pi_\theta(\cdot \mid x)$; extract $\hat{y}_i$, ℓ_i
2: $\tau_t \leftarrow 1.2 \cdot \max(\mathrm{med}(\mathcal{W}),\ 100)$
3: **for** $i = 1 \ldots G$ **do**
4: $\quad r_i \leftarrow$ Eq. 3a
5: $\quad$ **if** $\hat{y}_i = y^*$ **then** push ℓ_i into $\mathcal{W}$ ▷ only correct lengths calibrate τ
6: $\quad$ **end if**
7: **end for**
8: GRPO update with $\{r_i\}$ via Eq. 1

Reward-population migration. From the persisted reward state we log, at each checkpoint, how many of the group's correct rollouts fall under the active threshold. At step 300, 12 of 20 correct rollouts earn full credit ($\ell \le \tau_t$); at step 450, 19 of 21 do — the migration quoted in §3.2. Telemetry definitions: "policy entropy" is TRL's mean per-token entropy of the sampled completions; "ceiling-clipping" is the fraction of rollouts reaching `max_completion_length` = 4,096 (the *training* ceiling, distinct from evaluation budgets B); end-of-training telemetry values quoted in the main text are final-10-step means.

Compute. Each training arm runs 450 steps at 256 sequences per step ($4 \times 8 \times G$=8), i.e. 115,200 rollouts per run at up to 4,096 completion tokens, on two H200s with vLLM rollouts (bf16, gradient checkpointing). Six arms were trained in total (R1 and R3 on each of three backbones).

Prompts. Training uses the open-ended template only. Evaluation uses it for GSM8K, MATH, and AIME, and a multiple-choice variant for ARC-Challenge; both are zero-shot (no in-context examples), fixed, and identical across all models. Reproduced verbatim from the harness:

OPEN-ENDED TEMPLATE (training; GSM8K, MATH, and AIME evaluation)

```
Problem: {question}
Solution (put your FINAL answer in \boxed{} at the END):
```

MULTIPLE-CHOICE TEMPLATE (ARC-Challenge evaluation)

```
Question: {question}
Choose the correct option (A, B, C, or D). put your FINAL
answer in \boxed{} at the END):
```

D THE FULL BUDGETED GRID

This appendix is the evidence vault behind §3: every cell of the 11 models × 3 datasets × 5 budgets grid, in every metric the main text cites. Figure 4 shows the accuracy grid at a glance; the tables give the numbers.

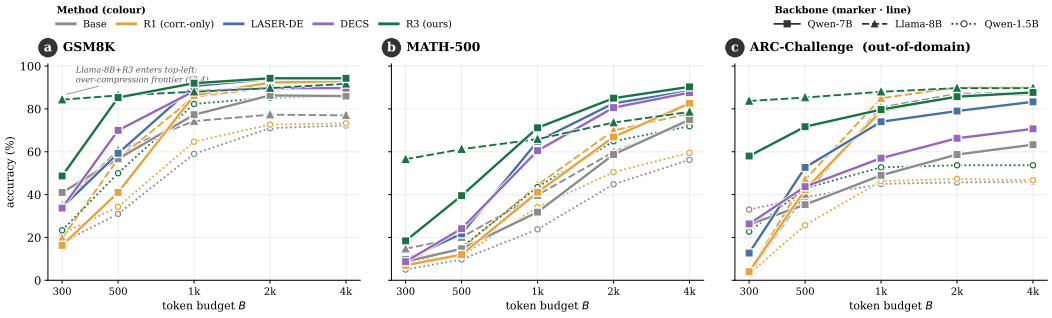


Figure 4: The full grid at a glance: accuracy vs. budget for all 11 models. Solid lines: the Qwen-7B cohort of the main text. Dashed and dotted green: R3 on Llama-8B and Qwen-1.5B — the ordering R3>R1>Base replicates in 8 of 9 backbone–dataset cells (Appendix I), and Llama-8B+R3's high, shallow curve entering at the top-left of each panel is the over-compression frontier discussed in §3.4. Faint grey: remaining base/R1 arms (dashed: Llama-8B; dotted: Qwen-1.5B; values in the tables below).

D.1 CORPUS DISCLOSURE

Training corpora differ across the grid, and we disclose rather than adjudicate. Our R1/R3 arms are trained on the GSM8K and MATH *training* splits only (7,500 each; evaluation draws MATH problems from the MATH-500 test subset (Lightman et al., 2024), disjoint from the training split by construction); the base models were produced by supervised distillation on an undisclosed corpus (Guo et al., 2025); LASER-DE and DECS checkpoints were trained on DeepScaleR-derived data per their releases (Liu et al., 2026a; Jiang et al., 2026), whose overlap with MATH we do not attempt to enumerate. Three consequences for reading the grid: (i) GSM8K/MATH comparisons between our arms and the external baselines mix method and corpus effects; (ii) the R3−R1 contrast is corpus-clean by construction, since both arms share identical data; (iii) ARC-Challenge appears in *no* model's training corpus, making it the one fully corpus-fair column — which is why the main text's largest margin claim is staked there.

D.2 PROTOCOL NOTES AND KNOWN MEASUREMENT CAVEATS

Timing scope. Generation time is logged per run; absolute values are comparable only within a dataset and only for models evaluated in the same session on identical hardware. All cross-model latency claims in the main text are therefore restricted to MATH-500 on the five Qwen-7B-backbone models, which ran under one session (§2.5); the base model's GSM8K timing, logged in an earlier session, is excluded from any latency comparison. Token counts, by contrast, are hardware-independent and comparable everywhere. **Truncation detection.** Truncation is classified by a balanced-`\boxed{}` depth scan plus a `</think>`-with-content check (Appendix G). One legacy artefact is disclosed: the base model's original MATH run predates the final detector and stores the harness fallback flag; re-scanning its raw generations with the final detector reproduces the reported rates, and the affected cells are marked in the source data. **Sample.** Every cell is the same fixed 300 problems per dataset — for MATH, sampled once with the fixed seed from MATH-500 — greedy decoding, one shared prompt and extractor.

D.3 FULL TABLES

Seventeen tables cover the full grid, one metric at a time, each as model × budget with one table per dataset. In order: *accuracy* (Tables 5, 6, 7), *mean generated tokens* (Tables 8, 11, 15), *median generated tokens*, *truncation rate*, and *per-problem generation time*, each over the same three datasets; the coupling summary (Table 18, derived in Appendix D.4); and the Eq. 2 decomposition (Tables 19–21, read in Appendix D.5). Rows are grouped by backbone; the R3 arms are shaded. Every value is generated directly from the per-run result files by the release pipeline.

Table 5: Accuracy (%) on GSM8K, all 11 models, greedy decoding, uniform prompt and extractor (n=300).

Model	300	500	1,000	2,000	4,000
Qwen-7B (Base)	41.0	56.7	77.3	86.3	86.0
Qwen-7B+R1	16.3	41.0	86.3	92.3	93.0
LASER-DE (7B)	34.0	59.3	90.7	94.0	94.0
DECS (7B)	33.7	70.0	88.3	89.7	89.7
Qwen-7B+R3 (ours)	48.7	85.3	92.0	94.3	94.3
Llama-8B (Base)	35.0	60.7	74.3	77.3	77.0
Llama-8B+R1	19.7	56.7	85.7	90.0	91.3
Llama-8B+R3 (ours)	84.3	86.3	88.0	89.7	91.7
Qwen-1.5B (Base)	17.7	31.0	59.0	71.0	72.3
Qwen-1.5B+R1	21.7	34.3	64.7	72.7	73.3
Qwen-1.5B+R3 (ours)	23.3	50.0	82.3	85.3	85.7

Table 6: Accuracy (%) on MATH, all 11 models, greedy decoding, uniform prompt and extractor (n=300).

Model	300	500	1,000	2,000	4,000
Qwen-7B (Base)	8.7	14.7	31.8	58.7	75.0
Qwen-7B+R1	7.0	12.0	41.1	66.9	82.6
LASER-DE (7B)	9.7	21.7	64.9	82.6	88.3
DECS (7B)	8.7	24.1	60.5	80.6	87.6
Qwen-7B+R3 (ours)	18.4	39.5	71.2	85.0	90.3
Llama-8B (Base)	14.7	20.1	39.8	59.9	73.9
Llama-8B+R1	9.7	14.7	44.5	69.9	77.9
Llama-8B+R3 (ours)	56.5	61.2	65.9	73.6	78.6
Qwen-1.5B (Base)	5.0	9.7	23.8	44.8	56.2
Qwen-1.5B+R1	6.7	11.4	34.1	50.5	59.5
Qwen-1.5B+R3 (ours)	7.0	15.4	43.5	64.9	71.9

Table 7: Accuracy (%) on ARC-Challenge, all 11 models, greedy decoding, uniform prompt and extractor (n=300).

Model	300	500	1,000	2,000	4,000
Qwen-7B (Base)	26.0	35.3	49.0	58.7	63.3
Qwen-7B+R1	4.0	42.3	79.0	86.0	87.0
LASER-DE (7B)	12.7	52.7	74.0	79.0	83.3
DECS (7B)	26.3	43.7	57.0	66.3	70.7
Qwen-7B+R3 (ours)	58.0	71.7	79.7	85.7	87.7
Llama-8B (Base)	3.3	40.3	80.7	87.0	87.7
Llama-8B+R1	4.0	47.0	85.0	90.0	90.0
Llama-8B+R3 (ours)	83.7	85.3	88.0	89.7	89.7
Qwen-1.5B (Base)	33.0	38.7	45.0	45.7	46.0
Qwen-1.5B+R1	3.0	25.7	46.3	47.3	46.7
Qwen-1.5B+R3 (ours)	22.7	43.0	52.7	53.7	53.7

Table 8: Mean generated tokens on GSM8K.

Model	300	500	1,000	2,000	4,000
Qwen-7B (Base)	280	416	632	869	1218
Qwen-7B+R1	300	494	750	846	904
LASER-DE (7B)	296	454	589	644	668
DECS (7B)	297	452	582	626	669
Qwen-7B+R3 (ours)	277	343	353	353	353
Llama-8B (Base)	292	407	498	572	641
Llama-8B+R1	300	493	751	854	963
Llama-8B+R3 (ours)	107	111	119	133	159
Qwen-1.5B (Base)	299	488	903	1402	2178
Qwen-1.5B+R1	300	497	894	1332	1999
Qwen-1.5B+R3 (ours)	300	494	778	905	1021

Table 9: Median generated tokens on GSM8K.

Model	300	500	1,000	2,000	4,000
Qwen-7B (Base)	300	500	618	618	618
Qwen-7B+R1	300	500	736	736	736
LASER-DE (7B)	300	500	542	542	542
DECS (7B)	300	500	536	536	536
Qwen-7B+R3 (ours)	300	333	333	333	333
Llama-8B (Base)	300	431	431	431	431
Llama-8B+R1	300	500	746	746	746
Llama-8B+R3 (ours)	94	94	94	94	94
Qwen-1.5B (Base)	300	500	1000	1402	1402
Qwen-1.5B+R1	300	500	1000	1254	1262
Qwen-1.5B+R3 (ours)	300	500	782	782	782

Table 10: Truncation rate $1-c(B)$ (%) on GSM8K.

Model	300	500	1,000	2,000	4,000
Qwen-7B (Base)	80.7	58.3	31.3	19.3	16.7
Qwen-7B+R1	100.0	89.7	23.7	3.7	2.3
LASER-DE (7B)	93.7	62.3	11.0	2.7	1.0
DECS (7B)	94.3	61.7	11.3	2.7	1.3
Qwen-7B+R3 (ours)	61.7	11.3	0.0	0.0	0.0
Llama-8B (Base)	82.7	34.7	10.0	5.3	3.3
Llama-8B+R1	99.7	88.3	22.0	6.7	4.7
Llama-8B+R3 (ours)	2.3	1.7	1.3	1.3	1.0
Qwen-1.5B (Base)	98.7	91.0	69.3	40.3	38.7
Qwen-1.5B+R1	100.0	96.3	62.0	34.7	33.3
Qwen-1.5B+R3 (ours)	100.0	91.3	28.3	7.3	5.3

Table 11: Mean generated tokens on MATH.

Model	300	500	1,000	2,000	4,000
Qwen-7B (Base)	299	496	954	1621	2358
Qwen-7B+R1	300	500	962	1545	2034
LASER-DE (7B)	300	496	887	1221	1413
DECS (7B)	300	498	886	1258	1515
Qwen-7B+R3 (ours)	293	446	624	696	738
Llama-8B (Base)	298	482	872	1485	2353
Llama-8B+R1	300	500	953	1506	1998
Llama-8B+R3 (ours)	193	243	324	420	538
Qwen-1.5B (Base)	300	498	985	1785	2854
Qwen-1.5B+R1	300	500	981	1720	2700
Qwen-1.5B+R3 (ours)	300	500	967	1516	2005

Table 12: Median generated tokens on MATH.

Model	300	500	1,000	2,000	4,000
Qwen-7B (Base)	300	500	1000	2000	2127
Qwen-7B+R1	300	500	1000	1668	1668
LASER-DE (7B)	300	500	1000	1118	1118
DECS (7B)	300	500	1000	1144	1144
Qwen-7B+R3 (ours)	300	500	576	576	576
Llama-8B (Base)	300	500	1000	2000	2166
Llama-8B+R1	300	500	1000	1596	1617
Llama-8B+R3 (ours)	183	183	183	183	183
Qwen-1.5B (Base)	300	500	1000	2000	3324
Qwen-1.5B+R1	300	500	1000	2000	2725
Qwen-1.5B+R3 (ours)	300	500	1000	1601	1601

Table 13: Truncation rate $1-c(B)$ (%) on MATH. Base row: the legacy run stores harness fallback flags (Appendix D.2); the 300- and 4,000-token endpoints shown are re-scanned with the final classifier, interior budgets in the supplementary decomposition files.

Model	300	500	1,000	2,000	4,000
Qwen-7B (Base)	95.0	—	—	—	20.6
Qwen-7B+R1	100.0	100.0	78.3	41.5	14.7
LASER-DE (7B)	99.7	94.7	60.5	18.4	5.3
DECS (7B)	100.0	97.3	59.5	22.4	7.0
Qwen-7B+R3 (ours)	90.3	62.2	17.7	3.0	0.7
Llama-8B (Base)	97.0	87.6	70.9	53.5	36.8
Llama-8B+R1	100.0	99.3	77.3	36.1	17.7
Llama-8B+R3 (ours)	32.8	21.4	13.4	7.0	4.7
Qwen-1.5B (Base)	99.3	98.7	94.7	67.2	45.8
Qwen-1.5B+R1	100.0	99.7	91.0	61.9	43.1
Qwen-1.5B+R3 (ours)	100.0	99.7	80.6	37.8	18.4

Table 14: Mean generation time per problem (s) on MATH. Timing scope per §2.5 and D.2.

Model	300	500	1,000	2,000	4,000
Qwen-7B (Base)	6.5	10.6	28.7	35.2	50.6
Qwen-7B+R1	5.5	9.6	18.4	29.5	38.7
LASER-DE (7B)	4.5	7.6	13.5	18.8	27.9
DECS (7B)	8.0	14.1	25.9	24.1	29.5
Qwen-7B+R3 (ours)	5.7	9.1	12.5	14.3	13.8
Llama-8B (Base)	6.2	10.2	18.7	31.7	37.2
Llama-8B+R1	6.5	11.6	22.3	37.3	73.8
Llama-8B+R3 (ours)	4.0	5.2	7.0	9.0	11.7
Qwen-1.5B (Base)	2.8	4.8	9.2	16.6	26.4
Qwen-1.5B+R1	5.2	8.6	16.9	26.6	49.1
Qwen-1.5B+R3 (ours)	3.0	5.0	9.7	15.2	20.1

Table 15: Mean generated tokens on ARC-Challenge.

Model	300	500	1,000	2,000	4,000
Qwen-7B (Base)	273	416	687	1094	1770
Qwen-7B+R1	300	470	641	763	887
LASER-DE (7B)	294	433	579	716	858
DECS (7B)	264	386	582	863	1226
Qwen-7B+R3 (ours)	190	233	260	287	331
Llama-8B (Base)	300	470	636	731	774
Llama-8B+R1	300	468	629	703	748
Llama-8B+R3 (ours)	111	119	131	147	173
Qwen-1.5B (Base)	182	238	330	462	709
Qwen-1.5B+R1	299	491	812	1285	2222
Qwen-1.5B+R3 (ours)	240	329	420	515	695

Table 16: Median generated tokens on ARC-Challenge.

Model	300	500	1,000	2,000	4,000
Qwen-7B (Base)	300	500	866	866	866
Qwen-7B+R1	300	500	575	575	575
LASER-DE (7B)	300	490	490	490	490
DECS (7B)	300	500	514	514	514
Qwen-7B+R3 (ours)	205	205	205	205	205
Llama-8B (Base)	300	500	582	582	582
Llama-8B+R1	300	500	562	562	562
Llama-8B+R3 (ours)	86	86	86	86	86
Qwen-1.5B (Base)	172	172	172	172	172
Qwen-1.5B+R1	300	500	997	997	1096
Qwen-1.5B+R3 (ours)	300	364	364	364	364

Table 17: Truncation rate $1-c(B)$ (%) on ARC-Challenge.

Model	300	500	1,000	2,000	4,000
Qwen-7B (Base)	78.3	65.0	48.3	38.0	31.0
Qwen-7B+R1	98.7	68.0	18.0	8.7	4.7
LASER-DE (7B)	90.3	49.7	20.7	11.3	4.3
DECS (7B)	71.3	51.0	33.7	23.0	14.3
Qwen-7B+R3 (ours)	33.3	12.3	3.3	2.3	1.3
Llama-8B (Base)	99.0	67.3	15.7	4.7	1.0
Llama-8B+R1	98.7	64.0	13.0	3.7	2.0
Llama-8B+R3 (ours)	5.7	3.7	2.0	1.3	1.3
Qwen-1.5B (Base)	31.3	24.3	14.3	12.7	12.3
Qwen-1.5B+R1	99.3	90.3	50.3	46.3	46.7
Qwen-1.5B+R3 (ours)	60.3	32.7	10.7	9.0	9.0

Table 18: Budget-coupling Δtok: percentage growth in mean generated tokens when the budget doubles 2k $\rightarrow$ 4k (derivation of Corollary 2 and Table 1's Δ column, extended to all models and datasets).

Model	GSM8K	MATH	ARC-C
Qwen-7B (Base)	+40.2%	+45.5%	+61.8%
Qwen-7B+R1	+6.9%	+31.7%	+16.3%
LASER-DE (7B)	+3.7%	+15.7%	+19.8%
DECS (7B)	+6.9%	+20.4%	+42.1%
Qwen-7B+R3 (ours)	+0.0%	+6.0%	+15.3%
Llama-8B (Base)	+12.1%	+58.5%	+5.9%
Llama-8B+R1	+12.8%	+32.7%	+6.4%
Llama-8B+R3 (ours)	+19.5%	+28.1%	+17.7%
Qwen-1.5B (Base)	+55.3%	+59.9%	+53.5%
Qwen-1.5B+R1	+50.1%	+57.0%	+72.9%
Qwen-1.5B+R3 (ours)	+12.8%	+32.3%	+35.0%

Table 19: Decomposition summary (Eq. 2) on GSM8K: per-completion accuracy A_c at the 4,000-token budget, its mean and standard deviation *across all five budgets* (a small SD is budget-invariance), truncated-credited accuracy A_t at 4,000, and the residual truncation rate.

Model	A_c@4k	mean A_c	SD A_c	A_t@4k	$1-c$@4k
Qwen-7B (Base)	94.0	94.7	1.2	46.0	16.7
Qwen-7B+R1	94.9	96.1	1.1	14.3	2.3
LASER-DE (7B)	94.9	97.3	2.2	0.0	1.0
DECS (7B)	90.5	93.0	1.8	25.0	1.3
Qwen-7B+R3 (ours)	94.3	94.0	1.7	—	0.0
Llama-8B (Base)	79.0	83.3	7.3	20.0	3.3
Llama-8B+R1	95.1	97.4	2.4	14.3	4.7
Llama-8B+R3 (ours)	91.9	88.9	2.3	66.7	1.0
Qwen-1.5B (Base)	91.8	87.8	7.3	41.4	38.7
Qwen-1.5B+R1	91.5	95.0	4.1	37.0	33.3
Qwen-1.5B+R3 (ours)	90.1	94.0	4.5	6.2	5.3

Table 20: Decomposition summary (Eq. 2) on MATH: per-completion accuracy A_c at the 4,000-token budget, its mean and standard deviation *across all five budgets* (a small SD is budget-invariance), truncated-credited accuracy A_t at 4,000, and the residual truncation rate.

Model	A_c@4k	mean A_c	SD A_c	A_t@4k	$1-c$@4k
Qwen-7B (Base)	92.9	81.3	15.0	6.1	20.6
Qwen-7B+R1	92.9	94.6	1.4	22.7	14.7
LASER-DE (7B)	92.6	96.9	3.3	12.5	5.4
DECS (7B)	92.4	95.9	3.2	23.8	7.0
Qwen-7B+R3 (ours)	90.6	87.3	2.8	50.0	0.7
Llama-8B (Base)	89.9	92.4	5.0	46.4	36.8
Llama-8B+R1	91.5	95.4	3.8	15.1	17.7
Llama-8B+R3 (ours)	81.4	78.8	2.6	21.4	4.7
Qwen-1.5B (Base)	90.1	80.1	18.4	16.1	45.8
Qwen-1.5B+R1	90.0	94.3	4.2	19.4	43.1
Qwen-1.5B+R3 (ours)	86.1	92.0	6.1	9.1	18.4

Table 21: Decomposition summary (Eq. 2) on ARC-Challenge: per-completion accuracy A_c at the 4,000-token budget, its mean and standard deviation *across all five budgets* (a small SD is budget-invariance), truncated-credited accuracy A_t at 4,000, and the residual truncation rate. ‡For the two re-evaluated rows (Qwen-7B+R3, Llama-8B+R3), A_c mean/SD is over the three largest budgets (Appendix D.2).

Model	A_c@4k	mean A_c	SD A_c	A_t@4k	$1-c$@4k
Qwen-7B (Base)	79.7	78.8	2.5	26.9	31.0
Qwen-7B+R1	89.5	93.3	4.3	35.7	4.7
LASER-DE (7B)	84.3	87.3	3.8	61.5	4.3
DECS (7B)	79.0	76.6	3.5	20.9	14.3
Qwen-7B+R3 (ours)	88.5	85.9	3.6‡	25.0	1.3
Llama-8B (Base)	88.6	92.9	4.6	0.0	1.0
Llama-8B+R1	91.2	89.1	8.2	33.3	2.0
Llama-8B+R3 (ours)	90.2	89.8	0.6‡	50.0	1.3
Qwen-1.5B (Base)	49.8	48.6	1.6	18.9	12.3
Qwen-1.5B+R1	71.2	70.0	12.9	18.6	46.7
Qwen-1.5B+R3 (ours)	57.9	55.8	4.0	11.1	9.0

D.4 Coupling derivations

Table 18 extends Table 1's Δtok column to all models and datasets: each entry is $100 \cdot (\bar{\ell}_{4k} - \bar{\ell}_{2k})/\bar{\ell}_{2k}$ computed from the mean-token tables above. The Corollary 2 spectrum is the MATH-500 column of its first five rows. Two patterns extend it — together, Corollary 4's content. Within *every* backbone triad on MATH-500, coupling orders Base $>$ R1 $>$ R3 ($+45.5 > +31.7 > +6.0$; $+58.5 > +32.7 > +28.1$; $+59.9 > +57.0 > +32.3$) — the reward, not the backbone, sets the coupling. And Qwen-7B+R3 holds the smallest coupling in the entire 11-model grid on both mathematics benchmarks, including an exact $+0.0\%$ on GSM8K: the budget doubled and mean generation did not move.

D.5 Truncation decomposition

Tables 19–21 report Eq. 2's terms for all 11 models: per-completion accuracy A_c at the generous budget, its mean and standard deviation across all five budgets — a small SD is budget-invariance stated as one number — the truncated-credited term A_t, and the residual truncation rate. Two reading notes. First, the five Qwen-7B-cohort models' A_c@4k spans exactly the 90.6–92.9 band drawn in Fig. 2b, while their overall accuracy spans fifteen points: the gap is completion, not capability, now stated for every model rather than five. Second, SDs for high-truncation models are inflated by low-n cells at tight budgets, where few completions exist to score.

E SAMPLED EVALUATION: PASS@k STATISTICS

This appendix documents the sampled MATH evaluation behind §3.4 and Corollary 3: both models, 299 held-out MATH-500 problems, n=8 samples per problem (temperature 0.6, top-p 0.95), the 4,000-token budget, one shared prompt and extractor (§2.5). Confidence intervals are percentile bootstrap over problems (10,000 resamples); paired intervals resample problems and difference within the resample, so problem difficulty is held fixed.

Table 22 is the headline: R3 exceeds the base model at every k, and every *paired* interval excludes zero; Figure 5 gives the same result at the per-problem grain. Table 23 locates the mechanism — restrict both models to problems whose samples all terminate and they converge — with one honest caveat carried in the caption: that restriction is itself selective for the base model (160 of 299 problems survive at k=8) and nearly vacuous for R3 (280), which is the phenomenon being measured, seen from the other side. Table 24 disarms the inverse artifact: conditioned on completion, the base model's per-attempt accuracy looks *higher*, because conditioning quietly discards its hardest 22% of attempts; this is the number footnoted in §3.4 and it should not be read as conditional superiority. Table 25 reports the exact paired test at k=8 for completeness: the discordant split favours R3 (14:5) but p=0.064 does not clear 0.05 at n=299 — the interval evidence of Table 22 is the primary claim, and we do not lean on the binary test.

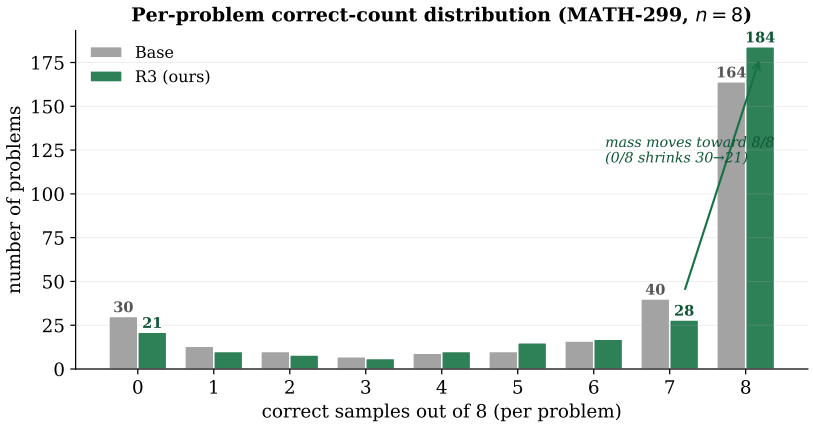


Figure 5: The microstructure of preserved diversity. Per-problem correct-count distribution over the 8 samples. R3 shifts mass from partial credit into 8/8 (164→184) while *also* shrinking the never-solved bin (30→21): compression consolidated solutions without abandoning hard problems — the per-problem grain of Corollary 3.

Table 26 prices the comparison in the sampling regime's own currency: eight attempts per problem cost R3 a third of the base model's tokens while solving more problems, and its median attempt (622 tokens) is where the base model's is 2,089. Table 27 then closes the loop on the static text-diversity proxy raised in §3.4: computed across all 11 models, the proxy flags every model in the grid — including untrained bases — which is precisely why we treat functional pass@k, not lexical similarity, as ground truth; within the proxy's own terms, the R3 arms degrade least or near-least on every backbone. Finally, Table 28 records decoding invariance: the learned stopping point moves by 46 median tokens between greedy and temperature-0.6 decoding — the rule lives in the policy, not the sampler.

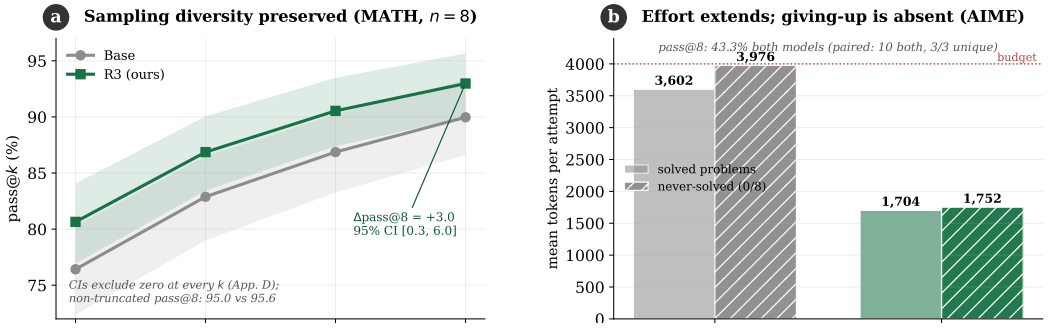


Figure 6: Compression costs nothing measurable. (a) pass@k on MATH-500 (temperature 0.6, n=8, 4,000-token budget; shaded 95% bootstrap CIs): R3 is above the base model at every k, with paired CIs excluding zero (Table 22); restricted to fully non-truncated problems, pass@8 converges (95.0 vs. 95.6) — the base deficit is completion, not exploration. **(b)** AIME 2024 effort: R3 spends as many tokens on problems it never solves as on those it solves (1,752 vs. 1,704), at less than half the base model's ceiling-pinned generation; pass@8 is identical and the paired outcome is symmetric (10 both, 3/3 unique, Appendix F).

Table 22: pass@k on MATH-299 (temperature 0.6, top-p 0.95, n=8, 4,000-token budget) with 95% bootstrap CIs, and the *paired* per-problem difference. Every paired interval excludes zero.

k	Base	**R3 (ours)**	Δ (paired)
1	76.4 [72.4, 80.3]	**80.6** [76.9, 84.1]	+4.2 [1.9, 6.7]
2	82.9 [79.0, 86.5]	**86.9** [83.4, 90.1]	+4.0 [1.6, 6.4]
4	86.9 [83.3, 90.2]	**90.5** [87.4, 93.5]	+3.7 [1.3, 6.3]
8	90.0 [86.6, 93.3]	**93.0** [90.0, 95.7]	+3.0 [0.3, 6.0]

Table 23: pass@k restricted to problems whose first k samples all terminate (n used in parentheses). The models converge — the sampled-regime gap of Table 22 is completion failure. Note the restriction is increasingly selective for the base model (160 of 299 problems at k=8) and barely selective for R3 (280).

k	Base (n)	R3 (n)
1	90.8 (273)	80.9 (299)
2	93.5 (265)	87.2 (299)
4	95.3 (246)	90.9 (299)
8	95.6 (160)	95.0 (280)

Table 24: Per-attempt accuracy conditioned on completion, under sampling. The base model's higher conditional accuracy is a *selection artifact*: conditioning discards its hardest 22% of attempts (the truncated ones), while R3's 98.7% completion leaves almost nothing out. This is the artifact footnoted in §3.4.

Model	Acc. | completed (%)	completed attempts	completion (%)
Base	92.6	1,853	77.5
R3 (ours)	81.6	2,362	98.7

Table 25: Paired problem-level outcome at k=8 on MATH-299 with an exact McNemar test. The discordant split favours R3 (14:5) but does not reach $p<0.05$ at this n; the interval-based evidence of Table 22 is the primary claim, and this test is reported for completeness.

Outcome	problems
solved by both	264
R3 only	14
Base only	5
neither	16
exact McNemar p	0.064

Table 26: The sampling-efficiency ledger on MATH-299 (8×299 attempts each). R3 solves *more* problems on *one third* of the token bill.

Model	total tokens	solved@8	tok/solved	mean/att.	med./att.	trunc. (%)
Base	5,575,518	269	20,727	2,331	2,089	22.5
R3 (ours)	1,881,234	278	6,767	786	622	1.3

Table 27: Static text-diversity proxy on MATH (% change in a lexical-diversity statistic across the budget sweep. The proxy flags *every* model — including untrained bases — and flags the R3 arms least or near-least on each backbone; functional pass@k (Table 22) is the ground truth this proxy approximates.

Model	Δ diversity (%)
Qwen-1.5B+R1	-56.7
Qwen-1.5B (Base)	-54.6
Qwen-7B (Base)	-46.7
Llama-8B (Base)	-46.2
Qwen-1.5B+R3 (ours)	-45.5
Llama-8B+R1	-43.4
Qwen-7B+R1	-43.4
DECS	-37.2
LASER-DE	-34.9
Qwen-7B+R3 (ours)	-22.3
Llama-8B+R3 (ours)	-6.4

Table 28: Decoding invariance of the stopping rule: R3's median MATH generation at the 4,000-token budget under both decoders. The rule is a property of the policy, not the decoder.

Decoder	median tokens
greedy (temperature 0)	576
sampled (temperature 0.6, $8\times$)	622

F AIME 2024: PROTOCOL, STATISTICS, AND DERIVATIONS

AIME 2024 is the out-of-distribution stress test of §3.4: 30 problems, harder than anything in the training corpus, evaluated under the same sampled protocol as Appendix E (temperature 0.6, top-p 0.95, n=8, 4,000-token budget, shared prompt and extractor) — 240 attempts per model. We report sampled statistics only: at n=30, greedy scoring moves $\pm$6–7 points per problem flipped and is noise; and we run no enlarged-budget condition, since budgets beyond 4,000 belong to the unconstrained regime this paper explicitly scopes out (§2.5, §4).

Bootstrap procedure. The interval on Δavg@1 (Table 30) is a paired percentile bootstrap at the problem level: each problem's score is its mean correctness over its 8 samples; a resample draws 30 problems with replacement and differences the two models' means *within* the resample, so problem identity — and therefore difficulty — is held fixed; 10,000 resamples, seed 42, percentiles 2.5/97.5. The resulting $[-11.7, +5.8]$ spans zero, and its width is the honest cost of n=30: the data support parity and nothing stronger, in either direction.

What the paired table shows. Table 29 lists all 30 problems. Three readings matter. The outcome is symmetric — 10 problems solved by both, 3 uniquely by each, 14 by neither — so pass@8 ties at 43.3% not by aggregation coincidence but problem-by-problem. The base model's token column is pinned at the ceiling almost everywhere, solved or not: on AIME it has effectively no stopping rule (81.7% truncation, median exactly 4,000). And R3's effort tracks the problem, not the reward: Table 31 shows 1,752 mean tokens on problems it never solves against 1,704 on those it solves — the difficulty-avoidance failure mode of length-penalised training, tested directly and absent.

Two derivations quoted in the main text. Table 32 is the AIME edition of the conditional-accuracy artifact (companion to Table 24): the base model's 90.9% accuracy-given-completion rests on 44 of

240 attempts — the few problems easy enough to finish under the ceiling — and measures selection, not capability. Table 33 derives the 8.2% of Fig. 2c: 16 of the base model's 196 truncated attempts are scored *correct* by the shared extractor — the answer computed, visible in the unfinished trace, and denied only a termination. On AIME, where truncation is near-total, even the strictest form of the latent-answer phenomenon survives out of distribution.

Table 29: Per-problem paired outcomes on AIME 2024 (n=8 samples each; problems indexed in a fixed shuffled order). c: correct samples of 8; tok: mean generated tokens. †marks the six problems solved by exactly one model (three each). The base model's token column is pinned at the 4,000 ceiling almost everywhere; R3's tracks difficulty.

#	c_{B}	c_{R3}	$\mathrm{tok}_{\mathrm{B}}$	$\mathrm{tok}_{\mathrm{R3}}$	#	c_{B}	c_{R3}	$\mathrm{tok}_{\mathrm{B}}$	$\mathrm{tok}_{\mathrm{R3}}$
1	3	1	3,921	1,183	16	5	1	3,600	938
2†	0	1	4,000	1,441	17	8	5	3,149	1,598
3	0	0	4,000	3,628	18	0	0	4,000	2,226
4	0	0	4,000	2,239	19	0	0	4,000	1,570
5	1	4	3,939	2,438	20†	0	3	4,000	2,264
6	0	0	4,000	1,392	21	8	8	3,008	1,354
7	0	0	4,000	1,282	22	2	2	4,000	1,670
8†	0	1	4,000	3,799	23†	4	0	3,717	620
9	0	0	4,000	1,729	24	0	0	4,000	4,000
10	7	7	3,387	1,364	25	8	8	3,052	704
11	0	0	3,590	926	26	0	0	4,000	1,281
12	0	0	4,000	1,287	27	0	0	4,000	1,426
13	7	2	3,183	1,350	28†	1	0	4,000	1,506
14	0	0	4,000	617	29	0	0	4,000	2,890
15†	1	0	4,000	1,165	30	1	6	3,878	2,042

Table 30: AIME 2024 headline statistics with the paired bootstrap interval on Δavg@1 (procedure in text). The interval spans zero: we claim parity, never superiority.

Metric	Base	R3 (ours)	Δ (R3−Base)
avg@1 (%)	23.3	20.4	−2.9 [-11.7, +5.8]
pass@8 (%)	43.3	43.3	0.0
paired outcome	10 both, 3 R3-only, 3 Base-only, 14 neither		
median tokens	4,000	1,175	
truncation (%)	81.7	20.0	

Table 31: Mean tokens per attempt on problems eventually solved ($\geq$1 of 8) vs. never solved (0 of 8); attempt counts in parentheses. R3's near-identical effort on never-solved problems is the direct measurement that difficulty-avoidance ("giving up early to harvest the brevity reward") is absent.

Model	solved problems	never-solved
Base	3,602 (104)	3,976 (136)
R3 (ours)	1,704 (104)	1,752 (136)

Table 32: The conditional-accuracy selection artifact, AIME edition (companion to Table 24). The base model completes only 44 of 240 attempts; conditioning on completion selects the few problems easy enough to finish under the ceiling and should not be read as conditional capability.

Model	acc. | completed (%)	completed attempts	completion (%)
Base	90.9	44	18.3
R3 (ours)	25.0	192	80.0

Table 33: Latent answers inside truncated AIME attempts: truncated attempts whose visible output is scored correct by the shared extractor — the answer was computed and even legible, but the trace never terminated. This is the 8.2% quoted in §3.1 and Fig. 2c.

Model	truncated attempts	scored correct	rate (%)
Base	196	16	8.2
R3 (ours)	48	1	2.1

G MEASUREMENT STACK

Every number in this paper passes through the instruments defined here. The design principle is stated in §2.5 and bears repeating because it does work: the prompts (one per task format), decoder configuration, and answer extractor are shared by *all* models — ours, external baselines, all backbones — so that a model which underperforms does so under deployment's rules, not its authors' harness. The prompt template is listed in Appendix C; this appendix specifies the three instruments built on top of it: the truncation classifier, the answer-extraction ladder, and the latent-answer scans. Pseudocode below is normative.

G.1 TRUNCATION CLASSIFICATION

A generation is *naturally terminated* if either (i) it contains a complete, brace-balanced `\boxed{}` expression, or (ii) it closes its reasoning block (`</think>`) and produces non-empty content after it. Otherwise, a generation that reaches the budget B is *truncated*. Criterion (i) is a depth-counting scan rather than a regular expression because boxed answers routinely nest braces (`\boxed{\frac{a}{b}}`):

Algorithm 2 Truncation classifier

Require: generation g, budget B, generated length ℓ

if HASCOMPLETEBOXED(g) **then return** TERMINATED
if `</think>` $\in g$ **and** len(content after) > 0 **then return** TERMINATED
if $\ell \geq B$ **then return** TRUNCATED
return TERMINATED ▷ EOS before budget

function HASCOMPLETEBOXED(g)
    for each occurrence of `\boxed{` in g **do**
        scan forward maintaining brace depth d (starting d=1)
        if d returns to 0 before end-of-text **then return** TRUE
    end for
    return FALSE
end function

An earlier heuristic — treating a terminal period as evidence of completion — was removed after it misclassified truncations that happened to cut at sentence boundaries; the classifier above contains no punctuation heuristics. **Legacy note** (disclosure in Appendix D.2): the base model's original MATH-500 run predates this classifier and stores the harness fallback flag ($\ell \geq B \Rightarrow$ truncated); re-scanning its stored generations with Algorithm 2 reproduces the reported rates, and affected cells are marked in the source data.

G.2 ANSWER EXTRACTION

One extractor serves training rewards and all evaluations. It is an ordered ladder; the first strategy that yields a candidate wins:

1. the content of the *last* brace-balanced `\boxed{}` in the generation;
2. for multiple-choice (ARC-Challenge): an answer-option letter in the terminal answer position;
3. numeric fallback: the final number in the generation.

Candidates are normalised before comparison (whitespace, commas, and formatting stripped); numeric answers verify at absolute tolerance 10^{-2}, multiple-choice by exact letter match. The identical ladder scores complete and truncated generations alike — which is what makes the A_t term of Eq. 2 measurable: a truncated trace is scored on whatever it produced, exactly as deployment would score it.

G.3 Latent-answer scans

The scans quantify answers that were *computed but never expressed*. All three operate only on generations classified truncated by Algorithm 2:

Lenient. The ground-truth answer, after the same normalisation the grader applies, occurs anywhere in the truncated text. This is an upper bound on recoverable answers: it credits an answer that appears mid-reasoning even if never committed. On the base model's truncated MATH-500 generations this rate is 43.8% (Fig. 2c).

Strict. The truncated text already contains a brace-balanced `\boxed{}` whose content verifies against ground truth — the answer formally committed, the trace merely failing to stop afterwards. Rate: 8.5% on the same population. Note the two definitions are not nested readings of one event: strict requires the commitment ritual, lenient only the content.

Scored-correct (AIME). The full extraction ladder of §G.2 applied to the truncated text returns a verified answer — operationally identical to how deployment would score the trace. Rate: 8.2% (16 of 196 truncated attempts; Table 33). A dose–response check — truncated-correct mass shrinking in lock-step with truncation itself — appears in Appendix N.

G.4 Answer-stability metric

The stability statistics of Table 44 use a prefix scan: for a stored generation, the extractor of §G.2 runs on successive prefixes; the *stabilisation point* is the earliest prefix length after which the extracted answer never changes through the end of the generation. "Stable by 1,000" means the stabilisation point is $\leq$1,000 tokens; "correct at stable" scores the answer extracted at that point. The metric is diagnostic — it reads a generation post hoc and adds nothing at inference time.

H Qualitative traces

Verbatim excerpts, chosen to make the statistics of §3.1 concrete: one specimen per evidence tier of Appendix G.3, plus one R3 failure. Specimen 1 is drawn from the greedy grid of §2.5; Specimens 2–3 from the sampled AIME protocol of Appendix F (temperature 0.6); all at the 4,000-token budget. Ellipses mark elided text; Unicode sub/superscripts and division glyphs are transliterated to ASCII; model-emitted LaTeX markup is reproduced verbatim.

Answer reached; scored zero. MATH problem 72 (answer .38). The base model states the correct value in prose within the first 4% of its trace, then second-guesses and re-verifies until truncation — never emitting a `\boxed{}` — and is scored 0: the lenient tier of Fig. 2c, enacted. R3 boxes the same answer in 333 tokens.

BASE (truncated at 4,096-token ceiling)

```
Thus, log_2(x^4 y^3 z^2) = log_2(2^(-25/8)) = -25/8.
Taking the absolute value: | -25/8 | = 25/8.
So, the value is 25/8, which is already in lowest terms [...]
Thus, m = 25, n = 8, so m + n = 25 + 8 = 33.
**Final Answer**
\boxed{33}
</think>
Given the system of equations involving logarithms: [...]

(the trace's final characters):
[...] Adding the exponents:
-\frac{7}{6} - \frac{9}{8} - \frac{5}{6} = -\frac{28}{24} - \frac
    {27}{24} - \   [truncated]
```

R3 (same problem; terminates at 1,094 tokens)

```
[...] = 2^{-28/24 - 27/24 - 20/24} = 2^{-75/24} = 2^{-25/8}
Now, compute the logarithm:
\log_2(x^4 y^3 z^2) = \log_2(2^{-25/8}) = -\dfrac{25}{8}
Take absolute value:
\left|\log_2(x^4 y^3 z^2)\right| = \dfrac{25}{8}
**Final Answer:**
\boxed{33}
```

Answer committed; generation does not stop. AIME problem 16 (answer 33). The base model completes its derivation, boxes the answer, and closes its reasoning block — all within the first $\sim$80% of its trace — then begins a full formal re-derivation of the same solution and is still rewriting, roughly 800 tokens later, when the ceiling cuts it off mid-formula. This attempt is one of the 16 scored-correct truncations of Table 33: termination fails even where scoring survives.

BASE (truncated at the 4,000-token ceiling; the commitment)

```
  Thus, log_2(x^4 y^3 z^2) = log_2(2^(-25/8)) = -25/8.
  Taking the absolute value: | -25/8 | = 25/8.
  So, the value is 25/8, which is already in lowest terms [...]
  Thus, m = 25, n = 8, so m + n = 25 + 8 = 33.
  **Final Answer**
  \boxed{33}
  </think>
  Given the system of equations involving logarithms: [...]

BASE (the trace's final characters):
  [...] Adding the exponents:
  -\frac{7}{6} - \frac{9}{8} - \frac{5}{6} = -\frac{28}{24} - \frac
      {27}{24} - \   [truncated]
```

R3 (same problem; terminates at 1,094 tokens)

```
  [...] = 2^{-28/24 - 27/24 - 20/24} = 2^{-75/24} = 2^{-25/8}
  Now, compute the logarithm:
  \log_2(x^4 y^3 z^2) = \log_2(2^{-25/8}) = -\dfrac{25}{8}
  Take absolute value:
  \left|\log_2(x^4 y^3 z^2)\right| = \dfrac{25}{8}
  **Final Answer:**
  \boxed{33}
```

A failure, for balance. AIME problem 18 (answer 315). R3 answers incorrectly — and the trace shows what its failures look like: a clean stop at a committed, incorrect closed form, after 310 tokens. The failure is capability, not termination; the R3 failures we inspected stop at committed answers rather than exhibiting degenerate stopping (empty reasoning blocks, answer-then-continuation).

R3 (complete ending)

```
[...] The formula to calculate the number of rectangles in a
regular n-gon is given by:
\frac{n(n-2)(n-4)}{24}
for even n.
Apply the Formula: Substituting n = 12:
\frac{12 \times 10 \times 8}{24} = 40
**Final Answer:**
\boxed{40}
```

I CROSS-BACKBONE REPLICATION AND THE CALIBRATION FRONTIER

The full 11-model grids — accuracy, tokens, truncation, timing, and the Eq. 2 decomposition — live in Appendix D; this appendix reads them. Three questions: does the causal ordering replicate, what exactly did each backbone learn, and what does the one over-compressed arm teach about the calibration factor.

Replication, with its exception. Table 34 tests R3>R1>Base at the generous budget in every backbone–dataset cell. It holds in eight of nine — all six mathematics cells cleanly — failing once, on ARC-Challenge for Llama-8B, where R1 (90.0) edges R3 (89.7) by a single problem out of 300 while both clear the base (87.7). The margin is within evaluation noise, and we count it against ourselves under strict ordering. It is also where the mechanism predicts the smallest effect: the base model completes nearly everything at this budget (1.0% truncation), so there is little completion failure left for termination calibration to convert — R3's edge over the base shrinks to $+2.0$ points where its mathematics edges run to double digits. Coupling replicates without exception: within every backbone triad on MATH, Δtok orders Base>R1>R3 (Appendix D.4).

What each backbone learned. Table 35 states calibration as four signatures — small coupling, a budget-invariant median, early truncation collapse, preserved per-completion accuracy — and the three R3 arms separate cleanly on them. Qwen-7B realises all four. Llama-8B realises the *rule* perfectly and the *level* wrongly: its median is 183/183/183 across budgets — budget-invariance as exact as the 7B arm's — but at an operating point so low that per-completion accuracy concedes 8.5 points on MATH. Qwen-1.5B is the mirror failure: it learned the direction (its coupling is its triad's smallest) but not the rule — the median still chases the ceiling ($1{,}000 \rightarrow 1{,}601$) and truncation at 1,000 tokens remains 80.6%. Over-compression is a miscalibrated *intercept*; partial calibration is an incomplete *ratchet*. The distinction matters because the remedies differ: the former is a per-backbone constant ($1.2\times$, W), the latter plausibly wants more steps or capacity.

The frontier, in detail. Table 36 prices Llama-8B+R3's trade explicitly. It leads its base in raw accuracy at every budget on every dataset — including 84.3% on GSM8K within 300 tokens where the base needs substantially more, converging at a fraction of the base's tokens — while conceding a small per-completion accuracy cost on MATH — 3.8 points on the 183 problems both arms complete (85.8 vs. 89.9), smaller than the 8.5-point gap an unmatched comparison would suggest. Nothing about the concession is hidden by aggregate numbers: the arm is strictly better to deploy under any budget in this grid *and* measurably past its backbone's compression frontier, which is exactly the pair of facts that makes the calibration factor a dial rather than a constant (§4).

Table 34: Replication of the causal ordering at the generous budget: accuracy (%) at B=4,000 for every backbone triad. R3>R1>Base holds in eight of nine backbone–dataset cells — cleanly in all six mathematics cells — with one exception on ARC-Challenge for Llama-8B, where R1 edges R3 by a single problem and both clear the base. Full budget curves in Appendix D.

Dataset	Backbone	Base	R1	R3	R3−Base	order
GSM8K	Qwen-7B	86.0	93.0	94.3	+8.3	✓
	Llama-8B	77.0	91.3	91.7	+14.7	✓
	Qwen-1.5B	72.3	73.3	85.7	+13.4	✓
MATH	Qwen-7B	75.0	82.6	90.3	+15.3	✓
	Llama-8B	73.9	77.9	78.6	+4.7	✓
	Qwen-1.5B	56.2	59.5	71.9	+15.7	✓
ARC-Challenge	Qwen-7B	63.3	87.0	87.7	+24.4	✓
	Llama-8B	87.7	90.0	89.7	−2.0	✗
	Qwen-1.5B	46.0	46.7	53.7	+7.7	✓

Table 35: Calibration signatures of the three R3 arms on MATH (plus GSM8K accuracy at 300 tokens). Full calibration shows small coupling, a budget-invariant median, early truncation collapse, and preserved per-completion accuracy; the three arms realise it to different degrees — the frontier discussed in §3.4 and §4.

Arm	Δtok$_{2k\to4k}$	med. tok 1k/2k/4k	trunc@1k (%)	A_c@4k	SD A_c	GSM8K@300
Qwen-7B+R3	+6.0%	576/576/576	17.7	90.6	2.8	48.7
Llama-8B+R3	+28.1%	183/183/183	13.4	81.4	2.6	84.3
Qwen-1.5B+R3	+32.3%	1000/1601/1601	80.6	86.1	6.1	23.3

Table 36: The over-compression frontier in detail: Llama-8B base / Llama-8B+R3, per dataset. R3 leads raw accuracy at every budget — including a 10× compression on GSM8K at 300 tokens — while *conceding* per-completion accuracy on MATH: the calibration factor pushed past this backbone's frontier (§4).

Dataset	acc@300 (B/R3)	acc@4k (B/R3)	tok@4k (B/R3)	A_c@4k (B/R3)
GSM8K	35.0 / 84.3	77.0 / 91.7	641 / 159	79.0 / 91.9
MATH	14.7 / 56.5	73.9 / 78.6	2,353 / 538	89.9 / 81.4
ARC-Challenge	3.3 / 83.7	87.7 / 89.7	774 / 173	88.6 / 90.2

J Composition with token-level credit

Whether the termination axis is orthogonal to token-level credit assignment is directly testable: we apply the R3 and R1 reward to the released DECS checkpoint under the recipe of §2 unchanged — same data, steps, hyperparameters, and seed; only the initialisation differs — and evaluate the result under the full grid of §2.5. Because the composition arm trains on GSM8K and MATH while the DECS checkpoint was trained on DeepScaleR, accuracy movements on the mathematics benchmarks could in principle reflect fresh in-domain exposure rather than the termination term; a matched R1 (correctness-only) control on the same checkpoint rules this out for accuracy specifically (below), while showing coupling improvement is substantially a generic RLVR effect rather than reward-specific.

Table 37: Termination calibration (R3) raises accuracy and cuts tokens at every budget. DECS from its released checkpoint vs. the same checkpoint after R1 (correctness-only) or R3 training; greedy grid, n=300. Mean tokens reported at the tightest and most generous budgets; **bold** = column-best. Δtok (2k→4k mean-token growth) discussed in text.

	DECS		DECS+R1		DECS+R3	
Dataset	acc@300 / tok	acc@4k / tok	acc@300 / tok	acc@4k / tok	acc@300 / tok	acc@4k / tok
GSM8K	33.7% / 296.6	89.7% / 669.5	19.0% / 299.9	**91.7%** / 790.5	**60.0%** / **274.3**	**91.7%** / **363.5**
MATH	8.7% / **300.0**	87.6% / 1515.1	6.4% / **300.0**	86.3% / 1566.0	**20.7%** / 296.0	**88.3%** / **956.9**
ARC-Challenge	**26.3%** / 264.1	70.7% / 1225.5	8.0% / **298.8**	84.0% / 597.4	23.3% / 292.5	**84.7%** / **464.8**

Table 38: Median generated tokens per budget: DECS from its released checkpoint vs. the same checkpoint after R1 or R3 training.

	DECS					DECS+R1					DECS+R3				
Dataset	300	500	1k	2k	4k	300	500	1k	2k	4k	300	500	1k	2k	4k
GSM8K	300	500	537	536	536	300	500	700	700	700	300	**320**	**320**	**320**	**320**
MATH	300	500	1000	1140	1144	300	500	1000	1260	1246	300	500	**612**	**612**	**612**
ARC-Challenge	300	500	514	514	514	300	489	487	487	487	300	**388.5**	**388.5**	**388.5**	**388.5**

The clearest signature is the median (Table 38): DECS+R3's median generated length is flat from 500 tokens onward on GSM8K and ARC-Challenge, and from 1,000 tokens onward on MATH — the same budget-invariance signature R3 exhibits on its own backbone (Table 43). DECS+R1's medians reveal what the coupling percentages alone conceal: on GSM8K and MATH its median plateaus *above* DECS's own baseline (536→700; 1,144→1,260) rather than below it — a low percentage growth rate at a worse absolute operating point, not calibration. This is the same signature §3.2 identifies on the primary backbone: correctness-only training raises the typical trace length while flattening the tail's relative growth. Only on ARC does R1 modestly improve over DECS (514→487), still well short of R3's 388.5. The coupling-percentage convergence between R1 and R3 reported above is therefore partly an artefact of what percentages hide: R3's medians are lower in absolute terms on every dataset, which the percentage alone does not show. A matched R1 (correctness-only) control on the same DECS checkpoint (Table 37) separates two effects the corpus-mixing caveat above cannot on its own. **Coupling improves substantially under R1 alone**: on ARC-Challenge, R1 very nearly matches R3's flattening ($+1.7\%$ vs. $+1.9\%$, both from DECS's own $+42.0\%$), and on GSM8K both are small ($+2.3\%$ vs. $+0.0\%$, from $+6.9\%$); only on MATH does R3 hold a clear if modest edge ($+15.2\%$ vs. $+17.4\%$, from $+20.8\%$). Generic RLVR exposure, not the termination term specifically, accounts for most of DECS's own coupling reduction. **Accuracy does not follow the same pattern.** R1 sits at or below DECS's own baseline at every tight budget (ARC@300: $26.3\% \rightarrow 8.0\%$; MATH@300: $8.7\% \rightarrow 6.4\%$) — the same tight-budget cost correctness-only training exacts on the base model (§3.2), now reproduced on a second initialisation — while R3 lifts accuracy substantially at the same budgets (ARC@300: 23.3%; GSM8K@300: 60.0%), and does so at fewer tokens throughout (Table 37): at GSM8K's generous budget R3 matches R1's accuracy using 363.5 mean tokens against R1's 790.5. The R1 control therefore isolates the termination term specifically for the *accuracy* claim: DECS+R3's accuracy gains are attributable to the added reward term, not to further RLVR exposure on this corpus. A per-problem check locates the residual coupling gap on MATH: the share of problems still growing by more than 500 tokens between the 300 and 4,000-token budgets falls from 74.2% (DECS) to 36.8% (DECS+R3), with DECS+R1 in between — the mechanism transfers to the typical problem cleanly and to the hardest tail only partially. On ARC, the corpus-fair benchmark, R3's result is mixed by budget: a small regression at B=300 (26.3%→23.3%) alongside the grid's largest single accuracy gain at B=4,000 (70.7%→84.7%) — we report both rather than the latter alone.

K Generalization to code: HumanEval

All training data and every result in the main text concern mathematical reasoning. To test whether termination calibration generalises to a structurally different task, we evaluate the released DECS checkpoint against Qwen-7B+R3 and base Qwen-7B on HumanEval (Chen et al., 2021) (164 problems, pass@1, execution-verified) at two token budgets, on both checkpoints *unchanged* — neither model was trained or fine-tuned on code.

Table 39: HumanEval pass@1, mean/median tokens, and truncation rate at two budgets; generation time at 4,000 (Base timing not logged in this session).

	2,000-token budget			4,000-token budget			
Model	acc	mean/med. tok	trunc	acc	mean/med. tok	trunc	time
Base	50.61%	1696.1/2000	53.66%	58.54%	2491.9/2157	29.27%	—
DECS	54.27%	1479.0/1528.5	34.15%	57.93%	1874.4/1528.5	13.41%	36.16 s
R3 (ours)	**64.63%**	1439.0/1412.0	**29.27%**	**67.07%**	1821.8/1412.0	**7.32%**	**27.44 s**

Base's own signature confirms the pattern seen on mathematical reasoning: at B=2,000, its median equals the budget itself (2,000) — not a stopping decision but the ceiling, reflecting 53.66% truncation, the highest of any model at any budget tested. R3 leads both baselines at every budget on accuracy, truncation, and speed — including a 14.0-point lead over base at B=2,000 (64.63% vs. 50.61%) despite base needing more tokens to get there (1696.1 mean vs. R3's 1439.0). Both R3 and DECS's medians are genuine stopping points well below the ceiling; base's is not. R3 and DECS's medians are each flat 2,000→4,000 (0.0% growth); R3's edge over DECS here is capability and completion, not budget-invariance — the coupling gap that separates the two methods on math does not appear on code.

L CALIBRATION-FACTOR SENSITIVITY

The reward's calibration factor (1.2, §2.3) is a design choice, not a derived constant (§4). We check its local stability without retraining: for each correct rollout in a sample of logged training batches (steps 100, 200, 300, 450; n=150 correct rollouts), we recompute the reward tier — full credit if generated length is at most $f\times$ the window median recorded live at that step, partial credit otherwise — under $f \in \{1.0, 1.1, 1.2, 1.3, 1.5, 2.0\}$, using the same median each rollout was actually scored against.

Table 40: Reward-tier distribution and adjacent flip rate under counterfactual calibration factors.

f	full-credit share	flip vs. previous f
1.0	72.0%	—
1.1	75.3%	3.3%
1.2 (paper)	78.0%	2.7%
1.3	80.0%	2.0%
1.5	85.3%	5.3%
2.0	93.3%	8.0%

Between f=1.1 and f=1.3, 4.7% of assignments flip: the chosen value sits in a locally stable region rather than a knife-edge. Across the full sweep (f=1.0 to 2.0), 21.3% flip, and by f=2.0 93.3% of correct rollouts already receive full credit — the threshold has stopped discriminating length at all. This is a signal-level check on what the reward *would have rewarded*, not a behavioural ablation: it does not show what a model *trained* under a different factor would learn to do. That experiment is left to future work.

M EXPRESSIBILITY AND IDEALISED DYNAMICS OF THE TERMINATION REWARD

This appendix makes two informal claims of the main text formal: that R3 lies outside the reward family whose insufficiency Jiang et al. (2026) prove (§2.3, §4), and that the ratchet's predicted dynamics — monotone decline, halting where correctness binds — follow from the reward's structure (§2.3). Proposition M.1 is exact. Proposition M.2 is an idealisation, stated as such; its role is to derive the predictions that the within-run control of §3.2 then tests empirically.

Setting. Fix a prompt and let $\mathcal{O}$ be its set of rollouts; each $o \in \mathcal{O}$ carries a correctness $c(o) \in \{0, 1\}$ and a length $\ell(o) \in \mathbb{N}_{>0}$. A *trajectory-level reward* is any $R(o) = \rho(c(o), \ell(o))$. Write $\mathcal{C} = \{o : c(o) = 1\}$ and $\mathcal{C}_\tau = \{o \in \mathcal{C} : \ell(o) \le \tau\}$. Throughout, $\mathcal{C}_\tau \neq \emptyset$ and $\mathcal{O} \setminus \mathcal{C} \neq \emptyset$.

Definition M.1 (Boundary expression)**.** *R expresses the stopping boundary τ on $\mathcal{O}$ if $\arg\max_{o\in\mathcal{O}} R(o) = \mathcal{C}_\tau$: every correct rollout terminating within τ is optimal, and nothing else is.*

Definition M.2 (Correctness dominance)**.** *R is* correctness-dominant *on $\mathcal{O}$ if $\inf_{o\in\mathcal{C}} R(o) > \sup_{o\notin\mathcal{C}} R(o)$: every correct rollout outranks every incorrect one, regardless of lengths.*

Proposition M.1 (Separation)**.** *Fix τ such that $\mathcal{C}_\tau$ contains rollouts at two or more distinct lengths.* (i) *No reward of the linear length-penalised family $R_\gamma(o) = c(o) - \gamma\,\ell(o)$, $\gamma > 0$ — and more generally no reward with $\rho(1, \cdot)$ strictly decreasing — expresses τ. Moreover, every R_γ fails correctness dominance on any $\mathcal{O}$ containing an incorrect rollout shorter than some correct rollout by more than $1/\gamma$.* (ii) *Every member of the threshold-conditional family of Eq. 3a at fixed threshold τ — $\rho(1, \ell) = 1$*

for $\ell \le \tau$, $\rho(1,\ell) = c_p$ for $\ell > \tau$, $\rho(0,\ell) = 0$ — both expresses τ and is correctness-dominant, for every τ and every partial credit $c_p \in (0,1)$.

Proof. (i) If $\rho(1,\cdot)$ is strictly decreasing, its maximum over $\mathcal{C}$ is attained only at the minimal correct length $\ell_{\min}$. Hence $\arg\max_{\mathcal{O}} R \cap \mathcal{C} \subseteq \{o \in \mathcal{C} : \ell(o) = \ell_{\min}\}$, while any remaining maximisers are incorrect. Since $\mathcal{C}_\tau$ contains a correct rollout o' with $\ell(o') > \min_{\mathcal{C}_\tau} \ell$, that o' is not a maximiser, so the maximiser set cannot equal $\mathcal{C}_\tau$. For dominance: with $o_0 \notin \mathcal{C}$, $o_1 \in \mathcal{C}$, $R_\gamma(o_0) - R_\gamma(o_1) = \gamma\big(\ell(o_1) - \ell(o_0)\big) - 1 > 0$ whenever $\ell(o_1) - \ell(o_0) > 1/\gamma$. *(ii)* The reward takes values in $\{1, c_p, 0\}$ with $1 > c_p > 0$; the value 1 is attained exactly on $\mathcal{C}_\tau \neq \emptyset$, so the maximiser set is $\mathcal{C}_\tau$; and $\inf_{\mathcal{C}} R = c_p > 0 = \sup_{\mathcal{O}\setminus\mathcal{C}} R$. □

Remark M.1 (What the learning signal consumes). GRPO's group-normalised advantage (Eq. 1) is invariant to positive affine transformations of rewards within a group; the properties above are ordering and maximiser-set facts — exactly the invariant content. In particular, part (ii) holds *uniformly* in $c_p \in (0,1)$: the partial credit is ordinal. Its endpoints are the two failure modes the main text discusses: $c_p = 1$ collapses to correctness-only (the maximiser set becomes all of $\mathcal{C}$; no boundary — this is R1), and $c_p = 0$ breaks dominance (verbose-correct ties incorrect — the cliff of §2.3).

Remark M.2 (Scope). Part (i) concerns the same length-penalised family that Jiang et al. (2026)'s Theorem 2 analyses — and, via the monotonicity argument, any reward strictly decreasing in length on correct rollouts (fixed and cosine-shaped penalties) — but at a different level: their theorem addresses the *training signal* (the induced token advantage leaves the boundary-adjacent token unpenalised), ours the *objective* (no such reward makes a stopping boundary optimal). The statements are complementary; neither implies the other. Clipped variants (a graded price with a reward floor, as in HAPO (Huang et al., 2026a)) fall to the same argument wherever rollouts lie below the clip point: the maximiser over correct rollouts is the minimum length present, never an interval — though the floor does restore correctness dominance, a property that design shares with Eq. 3a. The prediction has been tested directly: composing R3 with a representative of the length-penalty family it is not subject to (Appendix J) recovers, on the median problem, the budget-invariance that family's own theory says it cannot express alone — with a residual tail showing the recovery is not yet complete. Rewards that are not length-monotone on correct rollouts — difficulty-conditioned dynamic targets — lie outside both that theorem and this proposition; the separation claims nothing about them. Expressibility concerns the objective, not the optimiser: Proposition M.1 states which stopping structures a reward can make optimal; whether training reaches them is the subject of Proposition M.2 (idealised) and §3 (empirical).

Idealised dynamics. Let F_t denote the length distribution conditional on correctness under the policy at step t (with $P_t(c{=}1) > 0$), m_t its median, and $\tau_t = 1.2 \cdot \max(m_t, 100)$ as in Eq. 3b. We idealise: *(I1)* the window median equals m_t (large-W limit); *(I2)* each post-activation update makes the correct-length distribution stochastically no larger, $F_{t+1}(x) \ge F_t(x)$ for all x — motivated by the reward's structure: within any mixed group, correct rollouts with $\ell \le \tau_t$ carry positive advantage and verbose-correct rollouts negative, so the expected update transports correct mass toward the full-credit set; *(I3)* the update strictly decreases m_t whenever a correctness-preserving decrease is locally available, and leaves it fixed otherwise.

Proposition M.2 (Ratchet monotonicity and the brake as a fixed point)**.** *Under (I1)–(I3), for $t \ge T_w$:* (a) *τ_t is non-increasing and bounded below by* 120*, hence converges to some $\tau_\infty \ge 120$.* (b) *The ratchet is stationary at t if and only if no correctness-preserving reduction of the correct-length median is available; consequently, at any stationary point with $\tau_\infty > 120$, further compression of typical correct length is achievable only by reducing correctness. Moreover, since $\rho(0,\ell) \equiv 0$, the dynamics exert no direct length pressure on incorrect rollouts at any step: all compression pressure acts within the correct set.*

Proof. (a) Stochastic ordering of distributions orders all quantiles: $F_{t+1} \ge F_t$ pointwise implies $m_{t+1} = \inf\{x : F_{t+1}(x) \ge \frac{1}{2}\} \le m_t$. The map $m \mapsto 1.2 \cdot \max(m, 100)$ is non-decreasing, so τ_t is non-increasing; it is bounded below by $1.2 \cdot 100 = 120$ and therefore converges. (b) is (I3) read at a stationary point, together with inspection of Eq. 3a: incorrect rollouts receive reward 0 independent of length, so no gradient component prices their length. □

Corollary M.1 (The objective is budget-free)**.** *No element of Eq. 3, the window, or (I1)–(I3) references a deployment budget B; τ_∞ is therefore a functional of the policy class and training distribution*

alone. Any dependence of a trained policy's stopping length on the deployment budget cannot originate in the objective — the exclusion that the coupling spectrum of §3.3 measures.

Remark M.3 (Status and empirical test). Proposition M.2 is an idealisation of expected updates — stochastic training can violate (I2) on individual steps even where it holds in expectation — and its value is that it derives, rather than asserts, the two predictions of §2.3: median length declines monotonically from the step the termination term activates (before T_w the reward reduces to correctness-only, so (I2) is not induced and no decline is predicted), and the decline plateaus exactly where correctness binds. The within-run control (Fig. 3a–b) is the direct test of both — the realised decline is near-monotone, with 35 of 249 post-activation step transitions ticking upward, none by more than six tokens against a 660-token decline — and Corollary M.1's exclusion is tested by Table 1's Δtok column.

N Relation to DECS: scope, arithmetic, and regime

Evaluation fairness, stated up front. DECS enters our grid from its released checkpoint, evaluated under the uniform prompt, decoder and extractor of §2.5 — deployment's rules, not its authors' harness. Numbers under a method's native harness can differ (LASER-DE, for instance, reports 92.2% on MATH-500 at ∼1,836 average tokens in its own setup (Liu et al., 2026a)); nothing below is a claim about either method's best-case configuration. It is a claim about what a fixed serving stack observes.

Theorem 2, and why R3 sits outside it. DECS's Theorem 2 analyses the training signal a sequence-level length penalty ($r - \gamma L$) induces under group-relative optimisation, and proves that the first redundant token — adjacent to the necessary-reasoning boundary — receives no penalisation, so the policy cannot learn to stop exactly there; token-level credit with an auxiliary judge and curriculum is their remedy (Jiang et al., 2026). The theorem is correct and its scope is the point. Eq. 3 exits the quantified family in three ways: its credit carries *no per-token price* — reward is constant in ℓ on either side of τ_t, so there is no γ; the length term is *conditional on correctness* — a wrong rollout's length receives no gradient pressure at all, where the $r - \gamma L$ family prices every token of every rollout; and the threshold is *self-referential* — τ_t tracks the policy's own recent correct lengths, making the reward nonstationary in a way a fixed γ cannot represent. A necessity theorem binds its family; R3 is sequence-level but not in the family, so the empirical question — does one scalar suffice? — was open, and §3 answers it for the deployment regime.

The dose–response arithmetic. §3.1 claims truncated-yet-correct mass tracks truncation itself. The identity makes this two lines: the mass is $(1 - c(B)) \cdot A_t(B)$. For DECS on MATH at B=4,000: $7.0\% \times 23.8 = 1.7$ pp — exact, from Table 20's terms. At B=2,000 the per-budget decomposition gives 7.3 pp; an independent check that assumes only A_c budget-invariance (SD 3.2 for DECS) bounds the same quantity at ≤8.9 pp from the aggregate tables alone. Truncation fell $22.4\% \to 7.0\%$ between the two budgets and the recoverable mass fell with it, $7.3 \to 1.7$ — the phenomenon shrinks in lock-step with its cause, which is what Fig. 2c's annotation states.

The fifteen cells. Table 41 lists every budget–dataset cell for the two external baselines. LASER-DE leads 12 of 15, including all nine at budgets ≥1,000; DECS's three wins sit exclusively at the tightest budgets (GSM8K@500, MATH@500, ARC-Challenge@300), where truncation dominates both models and outcomes turn on which fragments happen to score. Corollary 1's reading follows: this is a *regime* observation, not a quality ranking — a method carrying heavier machinery is not thereby better where deployment actually measures.

The unconstrained regime, conceded with numbers. In its native evaluation regime DECS reports an average generation of ∼3,968 tokens (Jiang et al., 2026), roughly 2.6× its behaviour under our 4,000-token grid — give a length-trained model a taller ceiling and it grows into it (§3.3). And conditioned on completion, DECS holds a per-completion edge over R3 at our most generous budget: 92.4 vs. 90.6. We report this without netting it away, and we bound what conditioning does: DECS's conditional set omits its 7.0% hardest attempts while R3's omits 0.7%, so the +1.8-point edge sits inside a 6.3-point difference in what is being averaged over. A per-problem matched-subset comparison would require per-sample records for the external checkpoint, which our aggregate

exports do not retain; the bound above is the honest substitute. The synthesis is §4's: two regimes, two orderings, and the field has been measuring in one and deploying into the other.

Table 41: LASER-DE vs. DECS accuracy (%) in all 15 budget–dataset cells under the uniform protocol. **Bold**: the three cells where DECS leads — all at the two tightest budgets, where DECS's slightly longer generations (Appendix D) occasionally clear where LASER-DE's truncate. LASER-DE leads the remaining 12, including all nine at budgets $\geq$1,000.

Dataset	Model	300	500	1,000	2,000	4,000
GSM8K	LASER-DE	34.0	**59.3**	90.7	94.0	94.0
	DECS	33.7	70.0	88.3	89.7	89.7
MATH	LASER-DE	9.7	**21.7**	64.9	82.6	88.3
	DECS	8.7	24.1	60.5	80.6	87.6
ARC-C	LASER-DE	**12.7**	52.7	74.0	79.0	83.3
	DECS	26.3	43.7	57.0	66.3	70.7

O ADAPTIVE ALLOCATION AND STABILITY

This appendix collects three single-purpose tables serving §3.3 and §3.1. The stratification table prices difficulty-adaptivity *within* MATH-500: all trained methods share a $\sim$1.9$\times$ easy-to-hard ratio, so adaptivity in slope is common — the operating point it adapts from is what separates them. Table 43 reports median generated tokens per budget, the signature statistic of §3.3: a median that stops tracking the ceiling is a learned stopping point (R3 from 1,000 tokens on MATH, from 500 on GSM8K), while a median pinned to the budget is compression at best. Table 44 reports answer stability under the prefix-scan metric of Appendix G.4: base models decide early and generate on — the within-trace view of the termination failure that §3.1 measures across traces. Strata and metric definitions: difficulty strata are MATH-500 levels (easiest vs. hardest occupied stratum); all cells are the greedy grid at n=300 per §2.5.

Table 42: Within-MATH difficulty stratification at the 4,000-token budget (mean tokens on easiest vs. hardest stratum). Adaptivity ratios are similar across methods; operating points are not.

Model	Easy	Hard	Ratio
Base	1,771	2,432	1.4$\times$
R1 (correctness-only)	1,514	2,495	1.6$\times$
LASER-DE	939	1,816	1.9$\times$
DECS	1,017	1,958	1.9$\times$
R3 (ours)	**424**	**785**	**1.9**$\times$

Table 43: **Median generated tokens per budget** (greedy grid, §2.5). A budget-invariant median is the signature of a learned stopping point; a ceiling-tracking median is the signature of compression. R3's median is fixed from 1,000 tokens on MATH and from 500 on GSM8K; every other model's chases the ceiling.

	MATH					GSM8K				
Model	300	500	1k	2k	4k	300	500	1k	2k	4k
Base	300	500	1000	2000	2127	300	500	618	618	618
R1	300	500	1000	1668	1668	300	500	736	736	736
LASER-DE	300	500	1000	1118	1118	300	500	542	542	542
DECS	300	500	1000	1144	1144	300	500	536	536	536
R3 (ours)	300	500	**576**	**576**	**576**	300	**333**	**333**	**333**	**333**

Table 44: Answer stability on MATH: share of problems whose final answer is fixed by 1,000 generated tokens, and accuracy at the stabilisation point. Base models decide early and generate on.

Model	Stable by 1k (%)	Correct at stable (%)
Base	61.9	66.9
R1 (correctness-only)	56.9	82.6
LASER-DE	76.6	88.3
DECS	71.9	87.6
R3 (ours)	**78.3**	**90.3**