CAP: COMMITMENT-AWARE PENALTIES FOR TRAINING AND DEPLOYING REASONING MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Reasoning models often keep writing after they have internally settled on an answer, and much of this additional output does not change the final result. Reinforcement learning methods that penalize length address this by cutting text indiscriminately, at a cost that falls hardest on the problems that need long reasoning most. We therefore propose **CAP**, **C**ommitment-**A**ware **P**enalty: a lightweight probe reads the model's internal commitment state from hidden states during a forward pass that GRPO already performs, so the penalty applies only to text generated after commitment. We train this probe separately on each of six models from five architectures. We then test the resulting reward in two settings: (1) At training, a single full-scale run on Qwen3-8B shows no detectable accuracy difference from outcome-only GRPO on four sets not used for checkpoint selection, while generating 9–28% fewer tokens per set. A conventional length penalty compresses more aggressively but trails CAP by 8.7 points on MMLU-Pro and 6.3 on AIME 2025; only the MMLU-Pro gap survives multiple-comparison correction. (2) At deployment, the same probe can instead be used on base models as a stopping rule: evaluated on every trace, it removes over half the chain-of-thought tokens on MATH-500 for under 3 accuracy points, losing fewer points than fixed-fraction truncation at similar savings and far fewer than confidence-triggered stopping. This operating point, however, does not transfer to harder benchmarks without re-tuning.

1 INTRODUCTION

Long chains of thought (CoT) are now a major cost of deploying reasoning models, and it is unclear how much of this cost is useful. Test-time compute rivals training compute in importance (Snell et al., 2025), and much of it goes to reflection, the visible self-checking that reinforcement learning with verifiable rewards is credited with teaching (Guo et al., 2025). This spending raises two questions. The first is whether the reflection is worth its tokens, since within a single problem the longest sampled chains are often the least accurate (Hassid et al., 2025). The second is whether the text faithfully reports what the model is doing at all (Turpin et al., 2023; Chen et al., 2025b). Both questions matter beyond efficiency, because chains of thought are also the channel through which reasoning systems are monitored (Korbak et al., 2025).

Recent causal evidence on the first question indicates that most reflection is performed after the decision it claims to inform. Truncation attribution places the answer's stabilization point well before the end of a typical chain, with up to 87% of reasoning tokens following it in some settings (Scalena et al., 2026). The final answer can be decoded from intermediate activations long before it is written (Boppana et al., 2026). Perturbing text after this point rarely changes the outcome (Scalena et al., 2026), and step-level perturbation finds many steps to be decorative (Zhao et al., 2025).

This paper asks whether the commitment signal can improve training. The studies above have not tested this with matched controls, and the answer bears on how such interpretability signals should be used. We place a frozen commitment probe (Scalena et al., 2026) inside a GRPO pipeline (group relative policy optimization, the standard RL recipe for reasoning models). The probe reads sentence-final hidden states during a forward pass the algorithm already performs and labels each sentence of each rollout with the model's internal answer state. The reward then penalizes text generated after internal commitment, and we call this design CAP (Commitment-Aware Penalty).

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

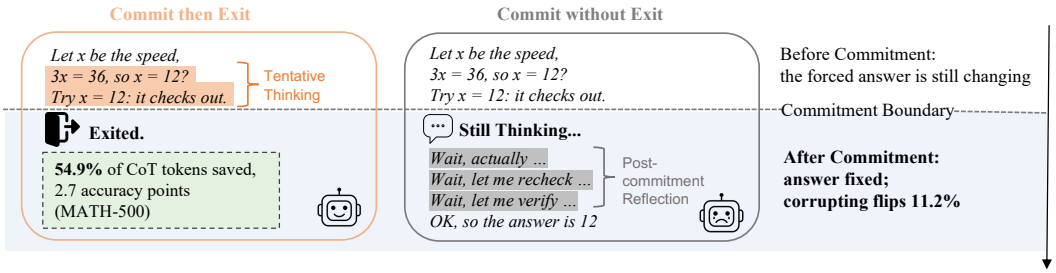


Figure 1: Schematic of commitment. Before the commitment boundary (dashed line), the model engages in tentative thinking. At the boundary the answer becomes fixed. Left (Commit then Exit): the model exits at the boundary, saving 54.9% of CoT tokens on MATH-500. Right (Commit without Exit): the model keeps generating post-commitment reflection (gray), all of which is redundant because the answer is already fixed. Values in the CoT example are illustrative.

Because a shaped reward's gain is hard to attribute, we compare CAP with matched controls on the same base model. The controls are outcome-only reinforcement learning from verifiable rewards (RLVR), a length penalty, a confidence-based twin of our reward, and an advantage filter in the style of ProFIL (Parekh & Goyal, 2026). All arms share data, rollout geometry, and step budget. Seeds are replicated at diagnostic scale and at a step-25 full-scale snapshot, while the step-50 comparison is a single trajectory per arm.

We make three contributions.

- **We introduce a commitment probe recipe.** A lightweight probe, trained separately on each of six models from five architectures, reads the internal commitment state. On Qwen3-14B the median boundary lies a quarter of the way into MATH-500 chains and later on harder sets. Corrupting half the numerals after the boundary leaves the answer intact in 89% of runs (93% on Phi, 57% on Llama; Appendix G).
- **We propose CAP, a commitment-aware penalty.** It penalizes only the post-commitment fraction of each rollout. The frozen probe reads the commitment state inside a forward pass that GRPO already performs, so the penalty adds no generation cost.
- **We evaluate it in training and at deployment.** In a single full-scale run, CAP shows no detectable accuracy difference from outcome-only GRPO (four-set difference $+0.1$ points, 95% CI $[-1.8, +2.0]$) while generating 9 to 28 percent fewer tokens per set (Table 2). The λ=0.8 length penalty compresses more but trails CAP by 8.7 points on MMLU-Pro ($p<10^{-3}$, significant after Holm correction) and by 6.3 on AIME 2025 (p=0.08, not significant). At inference on base models, the same probe recipe removes 54.9% of chain-of-thought tokens on MATH-500 for 2.7 accuracy points over all traces, whereas fixed-fraction truncation costs 5.2 points and confidence-triggered stopping costs 28.8 at similar savings (Tables 5 and 14). This operating point needs re-tuning on harder sets.

Further comparisons rule out simpler explanations. Blind compression loses significantly on MMLU-Pro (Table 2), and a reward of the same shape built from surface confidence does not beat the outcome-only control at diagnostic scale (Table 3). Point estimates for CAP are higher on Phi, where post-boundary text is inert (+1.3 average, one run, no interval), but not on Llama, where it is not (Appendix H).

2 Related Work

Evidence that late reasoning text rarely changes the answer. Causal and behavioral studies agree that much chain-of-thought content, and especially content after the commitment boundary, has little effect on the answer. Sentence-level truncation, step perturbation, and attention suppression show this (Scalena et al., 2026; Bogdan et al., 2025; Zhao et al., 2025), and the final answer is decodable from intermediate activations well before it is written (Boppana et al., 2026). Among behavioral studies, reflection markers appear in base models before any RL (Wei et al., 2025; Liu et al., 2025b;

Method	AIME 2025 ↑	GPQA-D ↑	ZebraLogic ↑	MMLU-Pro ↑	Average ↑
Qwen3-8B (initial)	0.501	0.573	0.580	0.565	0.555
DAPO (Yu et al., 2025)	0.540	0.578	**0.873**	**0.592**	0.646
Dr. GRPO (Liu et al., 2025b)	0.522	0.572	0.853	0.569	0.629
k1.5 length reward (Kimi Team, 2025)	0.549	0.575	0.845	0.560	0.632
ALP (Xiang et al., 2025)	0.552	0.580	0.865	0.537	0.634
Length penalty	0.551	0.586	0.865	0.548	0.637
CAP-T (ours)	**0.573**	**0.591**	0.853	0.573	**0.648**

Table 1: Diagnostic screen: 30 GRPO steps on Qwen3-8B, identical data, one seed per arm (the first seed where a second exists); baselines are our reimplementations. AIME 2025 uses 32 samples per problem, other cells eight; MATH-500 is the tuning set and is excluded. CAP-T is the targeted form (Section 3.3). Row differences under about two points are within evaluation noise.

Yang et al., 2025b), and few-shot prompting cuts reflection while raising accuracy (Ge et al., 2025). Thinking tokens can hurt under tight budgets (Ding et al., 2025), a flawed first step substantially lowers final accuracy (Liao et al., 2025), and reflective text is cheap to elicit by steering or logit arithmetic (Yan et al., 2026; Zhang et al., 2026). Our probe reads internal state rather than text, following the faithfulness and probing literatures. Verbalized reasoning misreports its own basis (Turpin et al., 2023; Lanham et al., 2023; Chen et al., 2025b), while hidden activations and self-evaluation heads expose what a model knows about its answer (Kadavath et al., 2022; Azaria & Mitchell, 2023; Burns et al., 2023). We reuse the probe of Scalena et al. (2026) and keep it frozen and outside the loss, which avoids the direct optimization pressure on the chain that Korbak et al. (2025) warn can erode monitorability. The causal measurements are held out for evaluation.

Internal signals as training rewards. Several methods already reward internal or causal signals, but none runs the controls that decide our question. ProFIL (Parekh & Goyal, 2026), the closest prior use of a commitment probe in RL, zeroes the advantage of rollouts whose probe score crosses a threshold. In our setting this design loses accuracy where dense shaping gains it (Table 3), plausibly because correct rollouts with long tails also lose their gradient. ProFIL reports the opposite on its own tasks, and reconciling the two settings remains open. PAIR (Kim et al., 2026) feeds hidden-state scores to agents as dense reward, Yu et al. (2026) reward behaviorally estimated causal importance, and AtManRL (Höth et al., 2026) derives rewards from attention saliency. Process reward models score step correctness from human labels (Lightman et al., 2024), automated search (Wang et al., 2024a; Luo et al., 2024), implicit credit (Cui et al., 2026; Yuan et al., 2025), or progress estimates (Setlur et al., 2025). Our signal is instead a label-free readout of belief change, and a step can be correct yet inert. Trained and prompted self-correction (Shinn et al., 2023; Madaan et al., 2023; Qu et al., 2024; Kumar et al., 2025) revises answers across turns instead of tracking belief within a chain.

Efficient reasoning and test-time control. Efficient-reasoning methods shorten chains either during training or at inference, and they form our baseline family. In training, length-targeted RL (Aggarwal & Welleck, 2025; Kimi Team, 2025; Luo et al., 2026; Hou et al., 2026; Xiang et al., 2025), adaptive thinking (Zhang et al., 2025b), budget prompting (Xu et al., 2025; Han et al., 2025), reflection-aware RL (Deng et al., 2026), and certainty-based suppression (Huang et al., 2026) shorten chains by construction. Sui et al. (2025) survey this family, which is motivated by overthinking on easy problems (Chen et al., 2025a; Arora & Zanette, 2025), and GRPO variants address its length bias (Liu et al., 2025b; Yu et al., 2025) and training stability (Yu et al., 2025; Yue et al., 2025; Liu et al., 2025a). Our results qualify these methods without dismissing them. Blind pressure does most of the causal targeting on easy distributions (Figure 3b) but at full scale loses accuracy on MMLU-Pro and, in point estimate, on AIME 2025 (Table 2). At inference, test-time compute allocation (Snell et al., 2025), budget forcing (Muennighoff et al., 2025), and confidence-triggered early exit of the chain (Yang et al., 2026; Fu et al., 2025) or, at the layer level, of the forward pass (Schuster et al., 2022) control the same budget. Hassid et al. (2025) find that shorter sampled chains are more accurate than the longest ones for the same question and let the first finished chains vote. The closest prior method probes hidden states for intermediate-answer correctness and stops when verification succeeds (Zhang et al., 2025a). Our probe instead reads commitment, is applied across datasets without retraining (one lightweight probe per model), and also serves as the training signal.

Method	AIME 2025 ↑	GPQA-D ↑	ZebraLogic ↑	MMLU-Pro ↑	Average ↑	Tokens (k) ↓
Qwen3-8B (initial)	0.463	0.573	0.580	0.565	0.545	8.2
GRPO (Shao et al., 2024)	**0.554**	0.581	0.870	**0.575**	0.645	7.8
Length penalty	0.483	0.581	**0.920**	0.469	0.613	4.8
CAP-T	0.492	0.574	0.905	0.438	0.602	5.2
CAP (ours)	0.546	**0.591**	0.893	0.556	0.646	6.5

Table 2: Full scale, one run per arm: step 50 at batch 128 (6,400 problems consumed), 16k budget, eight samples per problem, on four sets not used for checkpoint selection (three of them informed reward design at diagnostic scale). Tokens is the mean response length. Our row is the fraction form with the revision term; CAP-T is the Table 1 variant at full scale.

3 METHODOLOGY

3.1 PROBLEM FORMULATION

We first define commitment and the training and deployment objectives. A policy π_θ maps a prompt x to a CoT $C = (C_1, \ldots, C_n)$ of sentences and a final answer a, scored by a verifier $O(x, a) \in \{0, 1\}$. While generating, the model traverses internal answer states $s_i \in \{\text{NO}, \text{MID}, \text{FINAL}\}$, where s_i is the state after sentence C_i. The commitment boundary is $b = \min\{i : s_i = s_{i+1} = \text{FINAL}\}$ (set to n when no such pair exists), and the suffix $C_{b+1:n}$ is the post-commitment span. Training seeks

$$\max_\theta \ \mathbb{E}[O(x, a)] - \lambda \, \mathbb{E}[g(C_{b+1:n})] \,, \tag{1}$$

for a penalty g on the post-commitment span. The states s are estimated by a probe p_ψ over hidden states, whose parameters ψ stay frozen throughout training. Deployment seeks the shortest prefix of C that preserves O and stops generation once p_ψ signals commitment.

3.2 COMMITMENT MEASUREMENT

Following Scalena et al. (2026), we label commitment by forced decoding and train a probe to predict the labels. Each CoT is split into sentences $C_1, \ldots, C_n$. After each prefix $C_{1:i}$ an answer is force-decoded greedily from the prefix followed by a fixed answer template, which yields a per-sentence state y_i. The state is NO when no answer can be extracted or when the first answer token has probability below α=0.5; MID when an answer is extracted but is not equivalent to the model's own full-chain answer; and FINAL when it is equivalent. Two answers are equivalent if they match after normalizing whitespace and LaTeX formatting or, failing that, if math-verify judges them symbolically equal. On multiple-choice sets the option letters must match. The reference is the model's final answer, not the ground truth, so the label tracks commitment rather than correctness. To predict y_i online, a three-way attention probe with a sliding causal window is trained on sentence-final residual states. Layer and window are chosen on held-out validation traces, with accuracy reported on a disjoint test split (Section 4.2). Each model gets its own probe, trained once on that model's base rollouts and then frozen, so it never co-adapts with the policy. This one-time cost is about a tenth of one full-scale RL run (Appendix B). It reads only a correlate of commitment, so Section 4.2 tests separately, by truncation and perturbation, whether its boundary marks a fixed answer. The reward inherits that status only to the extent those tests transfer, which Section 4.5 measures per family.

3.3 REWARD DESIGN

The reward penalizes the post-commitment fraction of each rollout and can add a revision bonus. Let $s_{1:n}$ denote probe states for one rollout and R its set of reflective sentences. During training a trigger lexicon identifies reflective sentences, while a classifier distilled from 815k LLM-labeled sentences (F1 0.778) is used only for offline analysis (Appendix B). The reward terms are

$$\phi = \tfrac{n-b}{n}, \quad b = \min\{i : s_i = s_{i+1} = \text{FINAL}\}, \tag{2}$$

$$R_{\text{total}} = R_{\text{outcome}} + \lambda_1 r_{\text{rev}} - \lambda_2 \phi - \lambda_3 \mathbb{1}[\text{no answer}], \tag{3}$$

where ϕ is the post-commitment fraction of the rollout (zero when no boundary is found) and r_{rev} is the fraction of reflective sentences that change the probe state. The diagnostic-scale arm uses

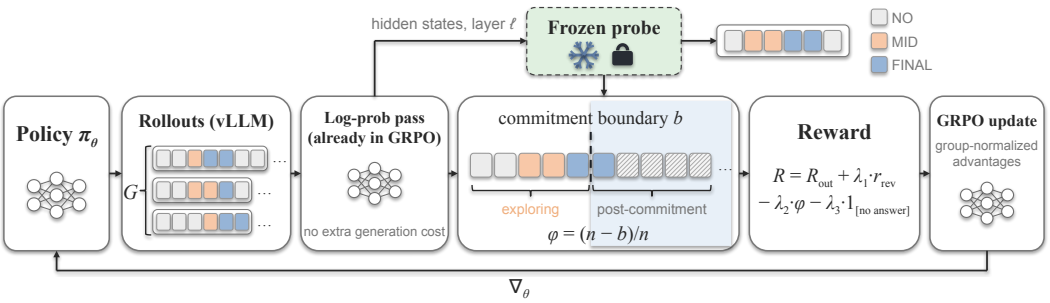


Figure 2: Training loop. The frozen probe reads hidden states in the log-probability pass and adds the commitment penalty to the reward.

the pure penalty $(\lambda_1, \lambda_2) = (0, 0.8)$. The full-scale arm was launched with the joint configuration $(0.5, 0.4)$ before the ablation showed the pure form to be stronger, so full-scale numbers carry a small revision bonus. At diagnostic scale the bonus lowered the gain (Table 3), but at full scale a pure-penalty arm shows no detectable difference from the joint form at step 25 (Appendix I).

Each control keeps this shape and replaces the probe signal. The length control uses normalized response length. The confidence control thresholds the sentence's geometric-mean token probability (exponentiated mean log-probability) to produce states of the same type. The filter control adds no dense term and instead sets the reward of rollouts with $\phi > 0.7$ to their GRPO group mean, which zeroes their advantage as in ProFIL.

The fraction form of ϕ has two structural defects, both exposed by the experiments. First, a policy can lower the fraction by deferring commitment instead of cutting text. Second, its pressure diminishes as the fraction shrinks, because a shorter tail changes the reward less, whereas a length penalty keeps pushing. This defect later turns out to be useful. The targeted variant CAP-T removes the second defect. It penalizes the absolute post-commitment span under the length penalty's normalization, $R = R_{\text{outcome}} - \lambda \min(c_{\text{post}}/24\text{k}, 1)$, where c_{post} counts characters after the boundary, so its penalty matches the blind control in size but applies only after the answer has been fixed. CAP-T still rewards deferring commitment, which the drift audit monitors (Appendix J).

3.4 Training System

The probe runs inside a standard GRPO loop and adds its penalty to the reward before advantages are computed (Figure 2). All Qwen arms train Qwen3-8B (Yang et al., 2025a) with GRPO (Shao et al., 2024) in verl (Sheng et al., 2025) on 23k verifiable problems (20k DeepScaleR (Luo et al., 2025) and 3k synthetic finite-automaton tasks with program-checked answers). The probe reads packed hidden states from the actor's old-log-probability pass and scores sentence-final positions on CPU (a 64-dimensional head, negligible next to generation; Appendix A). The evaluation signals (truncation and perturbation) never use the probe, and we audit its drift after training (Appendix J). Arms train on nodes of eight A100 or A800 GPUs with vLLM (Kwon et al., 2023) rollouts, and evaluations run on single GPUs at a 16k generation budget.

4 Experiments

4.1 Setup

We train at two scales, a diagnostic scale that screens reward designs cheaply and a full scale that decides between them. Diagnosis experiments use Qwen3-14B and Qwen3-8B on MATH-500 (Hendrycks et al., 2021; Lightman et al., 2024), AIME 2025, GPQA-Diamond (Rein et al., 2024), and ZebraLogic (Lin et al., 2025). At diagnostic scale, reward-ablation arms train on a fixed 512-problem subset for 30 GRPO steps (batch 60, response budget 8k, lr 10^{-6}), which is past the separation point of every effect we report. The fraction-form probe arm, the targeted form, the confidence twin, the GRPO control and the length control each have two seeds. At full scale, arms draw from the full 23k pool with a 16k budget, and 50 steps at batch 128 consume 6,400 problems.

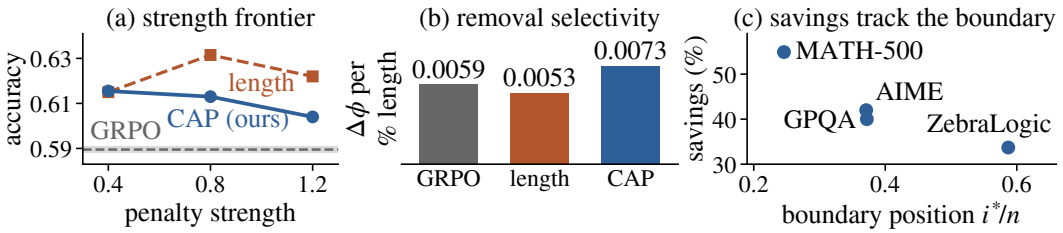


Figure 3: (a) Strength frontier: MATH-500 validation accuracy of CAP and the length penalty across penalty strengths 0.4–1.2 at diagnostic scale, with the two-seed GRPO band in gray. (b) Removal selectivity: how much of the post-commitment fraction ϕ (the share of the chain after commitment, defined in Section 3.3) each arm deletes per percent of response length removed; higher means the pressure lands more precisely on the tail. (c) Savings track the boundary: probe-gated early-exit token savings on four datasets against each dataset's median commitment boundary position i^*/n (four datasets, so illustrative rather than a fit).

Our evaluation separates the set used for selection from sets added after training. Validation accuracy is greedy pass@1 on MATH-500 unless stated otherwise, and MATH-500 also selects penalty strengths at diagnostic scale, an exposure shared equally by every arm. AIME 2024, AIME 2026, and MMLU-Pro (Wang et al., 2024b) are added after all training and selection and are touched by none of it. Hard-set retention in the sweeps uses AIME pass rate over eight samples, and the baseline table rescores AIME 2025 with 32. All generation uses Qwen3's default thinking mode, and "initial" denotes the released checkpoint before our training.

4.2 Diagnosis

Probe accuracy. We first check that the probe locates the boundary, which it does accurately on both Qwen models. On Qwen3-8B the selected configuration (layer 32, window 64) reaches 94.3% boundary detection (a hit is a prediction within two sentences of the true boundary) and 96.1% precision on the FINAL class. On Qwen3-14B the best configuration is layer 19 with window 64 at 94.0% and 95.2%. An answer-string rule (FINAL once the answer has appeared) matches the probe's accuracy but finds the boundary on 79% of chains and fires early on 21% (94% and 6% for the probe; Appendix B).

Where commitment falls. On MATH-500 the boundary comes early, and the text after it rarely affects the answer. We analyze the 1,303 of 3,976 Qwen3-14B traces on 497 problems that the attribution pipeline accepts, namely those that fit its memory budget and whose chain changes the answer relative to no chain (Appendix F shows how such filtering can bias statistics). On them the relative boundary position b/n (written i^*/n in figures) has median 0.246. Cutting the chain at the boundary reproduces the final answer, which holds by construction of the label and serves as a sanity check. Corrupting half of the numeric literals after the boundary preserves the answer in 88.8% of runs. The same corruption before the boundary, read at the boundary, preserves it in 48.3%, a contrast that reflects where the answer is read rather than the text itself. Read after the boundary, either corruption alone is nearly harmless (99.7% and 96.4%), since the tail restates the answer verbatim in 91% of traces (Appendix G). The boundary thus marks where the answer is fixed and begins to be restated. Its position depends on task structure as well as on the model, since within Qwen3-14B the median moves from 0.246 on MATH-500 to 0.371 on AIME, 0.372 on GPQA, and 0.587 on ZebraLogic, which refines the model-centric account in Scalena et al. (2026).

4.3 Main Results

We first screen reward recipes at diagnostic scale and then compare the selected designs at full scale. Table 1 compares CAP-T with four published reward recipes on a shared backbone, data and step budget (DAPO and Dr. GRPO also change the objective, a change that could be composed with our reward). CAP-T has the highest four-set average (0.648 against DAPO's 0.646) and the highest AIME 2025 and GPQA-Diamond scores. With one seed per arm, however, every recipe lies within

Arm	MATH-500 ↑	vs GRPO ↑
CAP (fraction form)	**0.616±0.003**	**+2.6**
w/o probe signal (confidence twin)	0.583±0.003	−0.7
w/o probe signal (answer-text rule)	0.593±0.003	+0.3
w/o dense term (advantage filter)	0.532	−5.8
w/ revision bonus	0.598	+0.8
revision bonus only	0.579	−1.1
targeted form	0.627±0.003	+3.7

Table 3: Component ablation: 512 problems, 30 steps, greedy MATH-500 at the 8k budget; ± is the half-range over two seeds. GRPO scores 0.590±0.002 and the tuned length penalty 0.632 (Figure 3a).

about two points of the others, a band that sampling noise can produce. For this reason we decide reward designs at full scale, where CAP-T does not hold up (Table 2).

At full scale, CAP shows no detectable accuracy difference from outcome-only GRPO. Table 2 compares CAP with GRPO and with the λ=0.8 length penalty selected at diagnostic scale, now trained on the full pool with a 16k budget instead of the 512-problem subset. Every arm is reported at step 50, which is both its last checkpoint and its best under greedy MATH-500 validation. A paired problem-level bootstrap (10,000 resamples) puts the four-set difference between CAP and GRPO at $+0.1$ points with a 95% interval of $[-1.8, +2.0]$, and the three-set difference without MMLU-Pro at $+0.8\,[-1.4, +3.1]$. Per set, CAP has a one-point edge on GPQA and two on ZebraLogic against deficits of 0.8 on AIME 2025 and 1.9 on MMLU-Pro, all within noise. On the untouched AIME 2024 and 2026, GRPO scores 0.671 and 0.558 against 0.667 and 0.583 for CAP. GRPO leads by 2.7 points on the tuning-exposed MATH-500 (0.730 against 0.703; Appendix E).

The length penalty, in contrast, does not match GRPO in accuracy. It wins ZebraLogic by 5.0 points but, in point estimate, gives back most of the RL gain on AIME 2025 (0.483 against a base of 0.463; not significant) and falls below the base on MMLU-Pro (0.469 against 0.565). Its four-set average therefore trails GRPO by 3.2 points $[+0.5, +5.9]$ and CAP by 3.3 $[+0.9, +5.6]$. Under a Holm correction across the eight per-set comparisons of CAP against GRPO and against the length arm, only the MMLU-Pro gap over the length arm survives. Broader draws of the two narrow sets, 1,000 MMLU-Pro test questions and 200 ZebraLogic puzzles, reproduce the ordering (Appendix E).

The two penalties also differ in how many tokens they remove and where. Relative to GRPO, CAP generates 14% fewer tokens on AIME 2025, 23% on GPQA, 9% on ZebraLogic and 28% on MMLU-Pro (30% on MATH-500; Appendix E). The length arm generates 30% to 56% fewer on the same four sets and pays for the extra compression on AIME 2025 and MMLU-Pro. When problems are stratified by GRPO's own solve rate (Appendix E), CAP cuts 18% of tokens on hard and medium problems and 23% on easy ones. The length arm cuts 38 to 46% in every stratum and loses 5.6 points of accuracy on medium problems, where CAP gains 1.8. The pressure of CAP is lighter, slightly more so on harder problems, as designed (though this does not measure exploration).

The targeted variant CAP-T, which won the diagnostic screen, does not survive scaling. By step 50 it compresses like the length arm and loses on the same two sets (AIME 2025 0.492, MMLU-Pro 0.438). In this single run, then, the fraction-form penalty with a small revision bonus (λ_1=0.5) reaches the observed accuracy of outcome-only training with fewer tokens. A pure-penalty arm matches it at step 25 but was not run to step 50. Appendix I reports three completeness checks at a matched step-25 snapshot (a second CAP seed, a pure-penalty arm, and a milder length penalty).

4.4 Reward Ablations and Mechanism

The component ablation in Table 3 tests which part of the reward produces its gain. The probe-ϕ arm gains 2.5 and 2.7 points over GRPO in its two seeds. Replacing the probe's boundary with the answer-string rule of Section 4.2, with the reward otherwise identical, removes the gain (0.590 and 0.595 over two seeds). A signal that matches the probe on three-way accuracy (Section 4.2) thus does not reproduce its effect as a reward. The confidence twin performs like GRPO, and the ProFIL-style filter loses accuracy from its own starting point. Adding r_{rev} weakens the gain, and

Model	Arch	Layer	3-way Acc ↑	Macro-F1 ↑
Qwen3-4B	qwen3	22	0.755	0.568
Qwen3-8B	qwen3	32	0.804	0.557
Llama-8B (DeepSeek-R1 distill)	llama	20	0.758	0.644
Phi-4-reasoning-plus	phi3	20	**0.880**	0.623
GPT-OSS-20B (OpenAI, 2025)	gpt-oss MoE	12	0.778	0.637
GLM-Z1-9B	glm4	25	0.828	**0.656**

Table 4: Commitment probes, one trained per model: best layer by validation macro-F1, accuracy on held-out traces. Phi-4-reasoning-plus (Abdin et al., 2025) is called Phi-14B, and the DeepSeek-R1 Llama distill Llama-8B, elsewhere. The Qwen3-14B probe used for diagnosis and early exit is in Section 3.2.

r_{rev} alone falls below the control. This fits its definition, since revision counts any state change, which can move toward a wrong answer as easily as toward a correct one.

At diagnostic scale, however, a naive length penalty is the strongest single arm on MATH-500 validation. With the same tuning budget, a sweep over strengths 0.4 to 1.2 with the best selected on validation, the probe arm plateaus while the length arm peaks higher (Figure 3a), and both reach identical 8k-budget AIME retention. Two further conditioned forms in Table 7, a probe-free correctness-conditioned length penalty (0.639) and a difficulty-scaled targeted form (0.637), sit inside the same noise band as the unconditioned recipes of Table 1 (0.629 to 0.646). This scale thus does not separate conditioning on commitment from conditioning on correctness or difficulty.

Selectivity helps explain why blind pressure gets so far (Figure 3b). Per percent of length removed, the probe reward deletes post-commitment content at rate 0.0073, against 0.0053 for the length arm and 0.0059 for RLVR drift. The probe is the most selective of the three, but the margin is small, plausibly because post-commitment tokens already carry the lowest advantage under every objective we test. Compression removes the post-commitment tail first without being told where it is, because with the median boundary a quarter of the way into MATH-500 chains most text lies after it.

Training also degrades the frozen probe, so during training ϕ is a lagged estimate of the post-commitment fraction. After training the probe is less accurate on every policy, including the outcome-only one that never saw it (trained at full rather than diagnostic scale, so the comparison does not fully separate reward from scale). Its three-way accuracy is 68 to 74% against 80% before training, and its boundary falls 10 to 13 sentences late (Appendix J). The probe still works as a stopping rule on every trained policy, beating fixed-fraction truncation at matched savings (Table 23; Section 4.6), and no policy shows signs of delaying commitment.

The same diagnosis explains why we carry the self-limiting fraction penalty rather than the targeted form to full scale. Of the two defects of the fraction form (Section 3.3), the targeted form removes the fading pressure but not the incentive to defer, and the difficulty-scaled form also multiplies by group solve rate. At full scale CAP-T is competitive at step 25 (AIME 2025 0.575) but by step 50 reproduces the length arm's failure signature (Table 2). A plausible reason is that an absolute penalty never turns itself off, so once the tail is consumed the pressure moves on to text that still matters. If so, the fading pressure of the fraction form, introduced above as a defect, is the property that protects hard problems over long training.

4.5 Cross-Model Results

We next test whether the probe and reward transfer to other families. The probe, trained separately on each model, works on every model we try (Table 4), with three-way accuracy between 0.755 and 0.880 across six models and five architectures. Three-way accuracy alone is not the right yardstick, however. On Qwen3-14B a position-only classifier already reaches 0.828 against a majority-class 0.747, and the probe's advantage lies in locating the boundary (94% within two sentences against 47%; Appendix B).

Whether the reward transfers depends on whether text after the probe's boundary is inert, which the perturbation test measures. Corrupting half the numerals after the boundary preserves the answer in 92.7% of Phi's runs but only 57.4% of Llama's (Appendix G), and the paired RL arms follow

Model	Full CoT ↑	Exit (k=10) ↑	Saved ↑	Fixed-fraction
Qwen3-14B	0.939	**0.912**	**54.9%**	0.887 @ 48.1%
Llama-8B	0.868	**0.795**	**39.5%**	0.768 @ 28.5%
Phi-14B	0.957	**0.918**	**44.4%**	0.805 @ 44.9%

Table 5: Probe-gated early exit against the fixed-fraction rule with the closest savings, on every MATH-500 held-out trace; paired differences are in Appendix F. Bold is our operating point.

that ordering (Appendix H). On Llama-8B CAP shows no gain, and every arm lands within noise of every other (averages over MATH-500, AIME 2025, GPQA and ZebraLogic of 0.413–0.431). On Phi-14B its point estimates exceed GRPO on three of four sets, by 3.4 points on AIME 2025 and 2.0 on GPQA, raising the average from 0.701 to 0.714 (one run, no interval). With two families at diagnostic scale, this is consistent with, but does not prove, the view that the reward helps where the boundary marks a fixed answer. Validation macro-F1 does not order the two families (Table 4).

4.6 Inference-Time Use of the Probe

The probe can also serve as a stopping rule at inference. Early exit stops generation after k consecutive FINAL predictions, and we first apply it to all 1,200 base-model traces of the 150 held-out MATH-500 problems, with no filtering. At k=10 the rule fires on 99% of traces and removes 54.9% of chain-of-thought tokens for 2.7 points of accuracy (0.912 against 0.939, 95% interval $[-5.2, -0.4]$ over problems; Table 5). At matched savings both alternatives cost more. Halving every chain gives up 5.2 points by sentences and 7.0 by tokens, and stopping after ten consecutive high-confidence sentences gives up 28.8 for 55% savings. On other families the rule needs no retuning. Phi gives up 3.9 points for 44.4%, where fixed-fraction truncation at 44.9% costs 15.2, and Llama, whose probe is the weakest, gives up 7.3 points for 39.5%, where the fixed rule costs 10.0 at only 28.5%.

The MATH-500 operating point does not transfer cheaply to other datasets. On every trace of GPQA, ZebraLogic and AIME 2025 it saves 34 to 42% of tokens for 6.2, 17.2 and 22.1 points (Appendix F), although it still beats confidence stopping on GPQA and AIME 2025 (not on ZebraLogic) and fixed truncation on the two hard sets. Figure 3c plots savings against each dataset's boundary position.

The exit rule also composes with training. On MATH-500, the Qwen3-8B probe that shaped training also serves the step-50 policies (Appendix F). On GRPO it removes 52.2% of tokens for 4.2 points, where halving by sentences costs 7.9. On CAP, whose chains are already a third shorter, it removes a further 23.2% for 2.2 points ($[-4.6, +0.3]$). Both end near 1.9k tokens, where CAP scores 0.908 against 0.891 ($[-0.6, +4.1]$), so training absorbs part of what the exit removes.

5 Conclusion and Limitations

A frozen commitment probe, read during a forward pass that GRPO already performs, gives reinforcement learning an estimate of where the answer is fixed. As a penalty on post-commitment text, CAP shows no detectable accuracy difference from outcome-only training in a single full-scale run while generating 9 to 28 percent fewer tokens, and it avoids the MMLU-Pro loss of a length penalty that compresses more. Confidence-shaped, answer-text and advantage-filter controls do not beat the outcome-only control at diagnostic scale. At inference the same probe removes over half of the chain-of-thought tokens on MATH-500 for under three points and loses far fewer points than confidence-triggered stopping on MATH-500, GPQA and AIME 2025, although its MATH-500 setting costs 6 to 22 points out of distribution.

Two limitations qualify these results. First, full-scale comparisons rest on one trajectory per arm (seeds replicated only at step 25 and at diagnostic scale), so the bootstrap covers evaluation noise but not optimization noise. The accuracy tie with outcome-only training is a null result, not an equivalence, and the one significant gap over the length penalty rests on 65 MMLU-Pro questions (a 1,000-question draw agrees; Appendix E). Second, the perturbation test shows that the answer is fixed at the boundary but not that the text before it is uniquely necessary.

AI USE STATEMENT

Apart from the LLM uses described as part of the method (sentence labels for the reflection classifier, Section 3.3, and names in the synthetic tasks, Appendix A), we used LLMs only for writing. They assisted with drafting and editing prose, with checking the manuscript for internal inconsistencies, and with verifying bibliographic metadata against the original sources. The authors reviewed all AI-assisted text and take full responsibility for the content of this paper.

REFERENCES

Marah Abdin, Sahaj Agarwal, Ahmed Awadallah, Vidhisha Balachandran, et al. Phi-4-reasoning technical report. *arXiv preprint arXiv:2504.21318*, 2025.

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

Daman Arora and Andrea Zanette. Training language models to reason efficiently. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Amos Azaria and Tom Mitchell. The internal state of an llm knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, 2023.

Paul C. Bogdan, Uzay Macar, Neel Nanda, and Arthur Conmy. Thought anchors: Which llm reasoning steps matter? *arXiv preprint arXiv:2506.19143*, 2025.

Siddharth Boppana, Annabel Ma, Max Loeffler, Raphaël Sarfati, Eric Bigelow, Atticus Geiger, Owen Lewis, and Jack Merullo. Reasoning theater: Disentangling model beliefs from chain-of-thought. In *International Conference on Machine Learning (ICML)*, 2026.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *International Conference on Learning Representations*, 2023. arXiv:2212.03827.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, et al. Do not think that much for 2+3=? on the overthinking of long reasoning models. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025a.

Yanda Chen, Joe Benton, Ansh Radhakrishnan, Jonathan Uesato, Carson Denison, John Schulman, et al. Reasoning models don't always say what they think. *arXiv preprint arXiv:2505.05410*, 2025b.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Yuchen Zhang, Jiacheng Chen, et al. Process reinforcement through implicit rewards. *Transactions on Machine Learning Research*, 2026.

Hexuan Deng, Wenxiang Jiao, Xuebo Liu, Jun Rao, and Min Zhang. Rea-rl: Reflection-aware online reinforcement learning for efficient reasoning. In *International Conference on Learning Representations (ICLR)*, 2026.

Bowen Ding, Yuhan Chen, Futing Wang, Lingfeng Ming, and Tao Lin. Do thinking tokens help or trap? towards more efficient large reasoning model. *arXiv preprint arXiv:2506.23840*, 2025.

Yichao Fu, Junda Chen, Siqi Zhu, Zheyu Fu, Zhongdongming Dai, Yonghao Zhuang, et al. Efficiently scaling llm reasoning programs with certaindex. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Yuyao Ge, Shenghua Liu, Yiwei Wang, Lingrui Mei, Lizhe Chen, Baolong Bi, and Xueqi Cheng. Innate reasoning is not enough: In-context learning enhances reasoning large language models with less overthinking. *arXiv preprint arXiv:2503.19602*, 2025.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, 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*, 2025.

Michael Hassid, Gabriel Synnaeve, Yossi Adi, and Roy Schwartz. Don't overthink it. preferring shorter thinking chains for improved llm reasoning. *arXiv preprint arXiv:2505.17813*, 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 *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021.

Max Henning Höth, Kristian Kersting, Björn Deiseroth, and Letitia Parcalabescu. Atmanrl: Towards faithful reasoning via differentiable attention saliency. *arXiv preprint arXiv:2604.16158*, 2026.

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. *Transactions on Machine Learning Research*, 2026.

Jiameng Huang, Baijiong Lin, Guhao Feng, Jierun Chen, Di He, and Lu Hou. Efficient reasoning for large reasoning language models via certainty-guided reflection suppression. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

Wonjoong Kim, Yeonjun In, Sangwu Park, Dongha Lee, and Chanyoung Park. Pair: Prefix-aware internal reward model for multi-turn agent optimization. *arXiv preprint arXiv:2605.17877*, 2026.

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

Tomek Korbak, Mikita Balesni, Elizabeth Barnes, Yoshua Bengio, Joe Benton, Joseph Bloom, et al. Chain of thought monitorability: A new and fragile opportunity for ai safety. *arXiv preprint arXiv:2507.11473*, 2025.

Aviral Kumar, Vincent Zhuang, Rishabh Agarwal, Yi Su, JD Co-Reyes, Avi Singh, et al. Training language models to self-correct via reinforcement learning. In *International Conference on Learning Representations (ICLR)*, 2025.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, 2023. arXiv:2309.06180.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.

Baohao Liao, Xinyi Chen, Sara Rajaee, Yuhui Xu, Christian Herold, Anders Søgaard, Maarten de Rijke, and Christof Monz. Lost at the beginning of reasoning. *arXiv preprint arXiv:2506.22058*, 2025.

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

Bill Yuchen Lin, Ronan Le Bras, Kyle Richardson, Ashish Sabharwal, Radha Poovendran, Peter Clark, and Yejin Choi. Zebralogic: On the scaling limits of llms for logical reasoning. In *International Conference on Machine Learning (ICML)*, 2025.

Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong. Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025a.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. In *Conference on Language Modeling (COLM)*, 2025b.

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.

Liangchen Luo, Yinxiao Liu, Rosanne Liu, Samrat Phatale, Meiqi Guo, Harsh Lara, et al. Improve mathematical reasoning in language models by automated process supervision. *arXiv preprint arXiv:2406.06592*, 2024.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, et al. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. `https://huggingface.co/agentica-org/DeepScaleR-1.5B-Preview`, 2025. Notion blog and Hugging Face release.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, et al. Self-refine: Iterative refinement with self-feedback. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, et al. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2025.

OpenAI. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Swapnil Parekh and Naman Goyal. Drop the act: Probe-filtered rl for faithful chain-of-thought reasoning. *arXiv preprint arXiv:2605.11467*, 2026.

Yuxiao Qu, Tianjun Zhang, Naman Garg, and Aviral Kumar. Recursive introspection: Teaching language model agents how to self-improve. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. Gpqa: A graduate-level google-proof q&a benchmark. In *Conference on Language Modeling (COLM)*, 2024.

Daniel Scalena, Sara Candussio, Luca Bortolussi, Elisabetta Fersini, Malvina Nissim, and Gabriele Sarti. Beyond the commitment boundary: Probing epiphenomenal chain-of-thought in large reasoning models. *arXiv preprint arXiv:2606.13603*, 2026.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. In *International Conference on Learning Representations (ICLR)*, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

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

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations (ICLR)*, 2025.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, et al. Stop overthinking: A survey on efficient reasoning for large language models. *Transactions on Machine Learning Research*, 2025.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, 2024a.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, et al. MMLU-Pro: A more robust and challenging multi-task language understanding benchmark. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2024b. arXiv:2406.01574.

Lai Wei, Yuting Li, Kaipeng Zheng, Chen Wang, Yue Wang, Linghe Kong, Lichao Sun, and Weiran Huang. Advancing multimodal reasoning via reinforcement learning with cold start. *arXiv preprint arXiv:2505.22334*, 2025.

Violet Xiang, Chase Blagden, Rafael Rafailov, Nathan Lile, Sang Truong, Chelsea Finn, and Nick Haber. Just enough thinking: Efficient reasoning with adaptive length penalties reinforcement learning. *arXiv preprint arXiv:2506.05256*, 2025.

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

Ge Yan, Chung-En Sun, Linbo Liu, and Tsui-Wei Weng. Reflctrl: Controlling llm reflection efficiently via representation engineering. In *Conference on Language Modeling (COLM)*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. Dynamic early exit in reasoning models. In *International Conference on Learning Representations (ICLR)*, 2026.

Shiming Yang, Yuxuan Tong, Xinyao Niu, Graham Neubig, and Xiang Yue. Demystifying long chain-of-thought reasoning. In *International Conference on Machine Learning (ICML)*, 2025b.

Qinan Yu, Alexa R. Tartaglini, Peter Hase, Carlos Guestrin, and Christopher Potts. Outcome rewards do not guarantee verifiable or causally important reasoning. In *International Conference on Machine Learning (ICML)*, 2026.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, et al. Dapo: An open-source llm reinforcement learning system at scale. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Lifan Yuan, Wendi Li, Huayu Chen, Ganqu Cui, Ning Ding, Kaiyan Zhang, Bowen Zhou, Zhiyuan Liu, and Hao Peng. Free process rewards without process labels. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

Yu Yue, Yufeng Yuan, Qiying Yu, Xiaochen Zuo, Ruofei Zhu, Wenyuan Xu, et al. Vapo: Efficient and reliable reinforcement learning for advanced reasoning tasks. *arXiv preprint arXiv:2504.05118*, 2025.

Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. Reasoning models know when they're right: Probing hidden states for self-verification. In *Conference on Language Modeling (COLM)*, 2025a.

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 (EMNLP)*, 2025b.

Yunxiang Zhang, Muhammad Khalifa, Lechen Zhang, Xin Liu, Ayoung Lee, Xinliang Frederick Zhang, Farima Fatahi Bayat, and Lu Wang. Logit arithmetic elicits long reasoning capabilities without training. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 24943–24959, 2026.

Jiachen Zhao, Yiyou Sun, Weiyan Shi, and Dawn Song. Can aha moments be fake? towards quantifying decorative and true thinking in chain-of-thought. *arXiv preprint arXiv:2510.24941*, 2025.

This appendix records the settings a reader needs to reproduce the paper and the full versions of the tables the main text summarizes. Section A gives the training configuration, the shared data, and the reward weight of every arm. Section B defines the commitment labels, the probe architecture, and how the probe is trained and selected. Section C states the generation and scoring protocol behind every number we report. Section D expands the diagnostic-scale comparison to every set, including the tuning-exposed MATH-500 column the main table omits. Section E gives per-set intervals and raw counts for every full-scale arm, broader draws of the two narrowest sets, and a registry of every RL arm. Section F specifies how early exit produces an answer, evaluates the rule on every MATH-500 trace against fixed-fraction and confidence-triggered stopping, and gives the all-trace sweep across three model families. Section G reports the perturbation dose-response curve behind the boundary claim, with position-matched and readout-matched controls. Section H pairs the RL arms on the two transfer families. Section I reports the step-25 completeness checks behind the robustness claims in the main text. Section J describes the audit testing whether the frozen probe can be gamed by the policy it trains.

A Training Configuration

This section gives the optimizer and rollout settings, the training data, the cost of running the probe during training, and the reward weights of every arm. The anonymized code supplement contains the reward function, the probe integration, the probe training scripts, the early-exit implementation, the synthetic data generator and the evaluation harness. The probes and run configurations will be released on publication.

All RL arms use verl GRPO, and within a scale every arm uses the same settings. At diagnostic scale, arms train for 30 steps at batch 60 on 512 problems with an 8k response budget. At full scale, arms train for 50 steps at batch 128 on the 23k pool in a fixed order, which consumes 6,400 problems, with a 16k response budget. Both scales use a learning rate of 10^{-6}, a KL coefficient of 0.001, an entropy coefficient of 0, eight rollouts per problem, and one epoch per batch. Training uses bf16 FSDP with parameter and optimizer offload and vLLM rollouts, on nodes of eight A100 or A800 GPUs.

All arms see the same data in the same order. They draw from one pool of 23k verifiable problems in a fixed shuffled order. The pool holds 20k problems from DeepScaleR and 3k synthetic finite-automaton tasks whose answers are checked by program rather than by string match. Their specifications and answers are generated programmatically, and an LLM only supplies scenario, state and action names, so no answer depends on model output. A full-scale run of 50 steps at batch 128 consumes the first 6,400 problems of that order, so no problem repeats within a run and every arm sees the identical 6,400. The diagnostic subset is a fixed random 512-problem sample of the pool, identical across arms; 30 steps at batch 60 pass over it about 3.5 times in the same order. Arms share data, order, rollout count and step budget, so the reward can be read as the only difference between them.

Running the probe adds little to the cost of training. The probe runs once per sentence of every rollout. A diagnostic step scores 60 problems $\times$ 8 rollouts, and a full-scale step 128 $\times$ 8. A rollout carries a few dozen sentences, so a step costs a few tens of thousands of probe calls. Each is a 64-dim attention head over cached states on CPU, and the cost stays under a percent of the step's wall time, which is dominated by generation.

The reward weights differ by arm and by scale. Every arm uses a no-answer penalty of $\lambda_3{=}0.5$. The CAP row of the diagnostic baseline table is the targeted form (CAP-T) with penalty weight 0.8 on the post-commitment span, the strongest diagnostic-scale variant. The ablations and the probe-drift audit use the fraction form with $(\lambda_1, \lambda_2) = (0, 0.8)$. At full scale CAP uses the fraction form with $(0.5, 0.4)$. The length control sweeps $\lambda_{\text{len}} \in \{0.4, 0.8, 1.2\}$ at diagnostic scale, and 0.8 is selected on validation.

The variants and controls are defined as follows. The targeted form uses the same penalty normalization (characters over 24k, capped at 1) on the post-commitment span only. The difficulty-scaled form multiplies that penalty by the GRPO group solve rate. The probe-free form applies the length penalty to correct rollouts only. The confidence twin thresholds the sentence's geometric-mean token

probability at τ=0.8. The filter control zeroes the advantage of rollouts with $\phi > 0.7$ by resetting their reward to the group mean.

B PROBE DETAILS

This section describes how the commitment labels are produced, how the probe is built, trained and evaluated, and how it compares with cheaper surface-cue predictors.

The labels the probe learns come from force-decoding, not from human annotation. Each chain is split into sentences, and an answer is force-decoded after every prefix. The prefix is labeled FINAL when that answer is semantically equivalent to the full-chain answer under a first-token confidence gate at α=0.5. It is labeled MID when a different answer is decodable, and NO when no answer is decodable yet. The commitment boundary b is the first index whose label and its successor are both FINAL, which requires the state to be stable rather than momentarily correct.

The probe is a causal-attention head with 64-dim hidden width and dropout 0.1, reading 64-token windows over layer-32 hidden states of Qwen3-8B, with last-token aggregation and three-way classification. It is trained on a question-level split that assigns 50%, 10% and 10% of problems to training, validation and test and holds the remainder in reserve, so the problems used for probe training, selection and evaluation are disjoint. Early stopping uses validation macro-F1, and layer and window are selected per family on the validation split. During training the probe runs once per rollout sentence on CPU in fp32 from cached hidden states.

Each model gets its own probe, trained with the same recipe on that model's own traces, and preparing it is a one-time cost per model. For Qwen3-8B, labeling the 3,867 MATH-500 traces by force-decoding took about nine hours of wall time on a few single GPUs (under 30 GPU-hours, including reruns), and the sweep over twelve layer and window settings took 24 minutes. A full-scale RL run of one arm takes about 240 GPU-hours, so the probe costs roughly a tenth of one run. The same frozen probe then serves both as the training reward and as the inference-time stopping rule, without retraining.

We report two probe metrics, and they measure different things. Boundary detection, reported as 94.3% in the main paper, counts hits within two sentences of the true boundary on FINAL-class onsets. Three-way accuracy scores every sentence, including the harder MID class, so the two metrics are not directly comparable. The three classes are also not balanced. FINAL covers roughly three in five sentences once the model has committed, which is why we report macro-F1 alongside accuracy and select on the former.

The stopping-rule parameter k trades two error modes against each other on the probe's own validation traces. At k=1 about 11% of chains fire before the true boundary, at k=2 about 6%, and at k=3 about 5%, while the fraction of boundaries the rule eventually detects moves the other way. The exit results in Section F show the same trade-off in accuracy and savings.

What the probe reads beyond surface cues. We test whether the probe measures anything a textual rule would not, since the answer is usually restated in the text once it is fixed. Table 6 scores five cheap controls on the same held-out sentences and labels as the Qwen3-14B probe. They are a position-only classifier, a lexical classifier over the sentence text with cue counts, a rule that predicts FINAL once the final answer string has appeared in the chain, a confidence rule on the stored token entropy, and the combination of all of these. The confidence control uses stored generation entropy because per-token log-probabilities of the sampled tokens were not saved with these traces. The training-time confidence twin of Section 3.3 uses the sentence-mean log-probability that the trainer computes anyway, so the two are related but not identical statistics.

Much of what the probe reads coincides with the answer having been written down. On three-way accuracy the answer-text rule matches the probe, and the probe's FINAL probability correlates with answer appearance at ρ=0.69.

The probe is nonetheless better on the measures a stopping rule needs. It finds the boundary within two sentences on 94% of chains, compared with 79% for the answer-text rule, and it fires before the true boundary on 6% of chains, compared with 21%. On the sentences before the answer has appeared in the text, the probe keeps a macro-F1 of 0.50, while the answer-text rule cannot distinguish

MID from NO (0.25). The force-decoded FINAL onset precedes the first textual appearance of the answer in 67% of chains, by a median of four sentences, and the exit rule exploits this margin.

Predictor (Qwen3-14B, held-out)	**Acc**	**Macro-F1**	**FINAL prec.**	**Boundary ± 2**	**Early**	**F1 before ans.**
Majority class	0.747	0.285	0.747	6.8%	93.2%	—
Position only	0.828	0.501	0.862	47.4%	52.6%	0.37
Lexical + cue counts	0.716	0.493	0.923	83.5%	15.8%	0.38
Answer-text rule	0.843	0.533	0.923	78.9%	21.1%	0.25
Confidence (entropy)	0.548	0.390	0.834	63.9%	28.6%	0.31
All surface features combined	0.820	0.561	0.928	80.5%	19.5%	0.39
Hidden-state probe	0.841	**0.625**	**0.952**	**94.0%**	**6.0%**	**0.50**

Table 6: The commitment probe against surface-cue controls on the same 19,307 held-out sentences (134 chains). Boundary ± 2 is the share of chains whose first stable FINAL pair falls within two sentences of the labeled boundary; Early is the share on which the predictor's boundary precedes the labeled one; the last column is macro-F1 on sentences before the answer string has appeared.

The reflection classifier used in the main paper's offline analysis never enters the reward. It is a separate head of the same architecture, distilled from 815k GPT-labeled sentences. During training, reflective sentences are identified by a trigger lexicon instead.

C Evaluation Protocol

This section states how every reported number is generated and scored, which role each evaluation set plays, how we checked the sets for overlap with the training pool, and how intervals are computed.

Every reported evaluation of every arm uses the same generation settings, with three exceptions. Generation uses the model's chat template with the raw question as the only user message, temperature 0.7, top-p 0.9, eight samples per problem and a 16k token budget. The first exception is the diagnostic-scale MATH-500 validation column of the ablation table and Figure 3a, which come from the trainer's own greedy validation at the 8k training budget. The second is AIME 2025 in the diagnostic tables (Table 1 of the main text and Table 7), which is rescored at 32 samples. The third is the 1,000-question MMLU-Pro draw of Table 10, which is scored at four samples. We verified the budget from the token counts of every reported trace file. We also re-ran the two ZebraLogic evaluations (the base model and the outcome-only arm at step 50) that an earlier pass had generated under an 8k budget.

The evaluation sets play three roles that we keep apart. MATH-500 is the validation set that selects checkpoints and penalty weights. AIME 2025, GPQA-Diamond and ZebraLogic were used as development benchmarks in the diagnostic-scale comparisons before the full-scale arms were trained. MMLU-Pro non-math and AIME 2024 and 2026 were added after reward development and never informed a design choice. Checkpoint selection uses greedy pass@1 on MATH-500 as the validation signal. Apart from the ablation column noted above, no reported number comes from that greedy pass.

We audited overlap between the 23k training pool and every evaluation set by normalized exact match, shared 13-word n-grams, and sentence-embedding cosine similarity (script in the supplement). GPQA-Diamond, ZebraLogic and MMLU-Pro share nothing with the pool under any criterion. The three AIME sets share a 13-word phrase with some pool problem on 9 or 10 of 30 problems. All of these phrases are formulaic geometry or number-theory wording, with a maximum cosine of 0.83 and no embedding near-duplicate. MATH-500, the tuning set that the held-out average excludes, has 2 normalized-exact and 15 embedding near-duplicates (cosine above 0.9) in the DeepScaleR portion of the pool, and 5 of them fall among the 150 problems of the early-exit evaluation split. The probe-training and early-exit evaluation splits of MATH-500 are disjoint by problem id and share no near-duplicate text.

Scoring and intervals are computed the same way for every set. Scoring is math-verify string equivalence against the reference answer. The set sizes are MATH-500 (500), AIME 2024, 2025 and

2026 (30 each), GPQA-Diamond (198), ZebraLogic (50 puzzles drawn from the WildEval multiple-choice test split with NumPy seed 12) and MMLU-Pro non-math (the 65 non-math questions of the TIGER-Lab validation split, used in full). Bootstrap intervals resample problems within each set 10,000 times with paired arms.

AIME 2025 is scored with more samples in the diagnostic baseline table. There it is re-scored at 32 samples per problem, while the full-scale table uses eight samples for every set. We rescore because a 30-problem set at eight samples moves by several points on resampling alone, which is enough to reorder arms that differ by less than that. The other sets are large enough that eight samples suffice.

D Complete Diagnostic-Scale Results

This section reports the diagnostic-scale arms on every evaluation set, including the sets that the main comparison omits, and the transfer results of the fraction-form reward and its surface-signal twins.

Table 7 lists every diagnostic-scale arm on every set. Compared with the main table, it adds the tuning-exposed MATH-500 column and the 2024 and 2026 AIME editions, which are reported only here because the main table keeps AIME 2025 only. MATH-500 separates the arms by less than a point, as expected for a set that every arm was tuned on.

Method	MATH	A25	GPQA	Zebra	A24	A26	MMLU
DAPO	**.697**	.540	.578	**.873**	.660	.579	**.592**
Dr. GRPO	.691	.522	.572	.853	.642	.558	.569
k1.5 reward	.693	.549	.575	.845	.642	.604	.560
ALP	.695	.552	.580	.865	.650	.546	.537
CAP-T	.690	**.573**	**.591**	.853	.654	.583	.573
difficulty-scaled	.691	.567	.582	.858	**.667**	.579	.540
probe-free	.694	.557	.583	.855	.654	**.617**	.562

Table 7: Diagnostic-scale reward recipes on every set, 30 GRPO steps on a shared Qwen3-8B backbone, one seed each (the initial model and the length penalty are in Table 1 of the main text). A25/A24/A26 are the AIME editions. AIME 2025 uses 32 samples per problem, all other cells eight. Bold is best per column.

On the transfer sets, the fraction-form arm of Table 3 and its two surface-signal twins are not separated beyond noise at this scale (Table 8, same 16k protocol). This holds on AIME 2025, GPQA-Diamond and MMLU-Pro. The answer-text twin trails the other two by six to seven points on ZebraLogic, but ZebraLogic has 50 puzzles, and one seed cannot establish a difference on a set of that size. The controlled comparison of the reward signals therefore remains the two-seed MATH-500 ablation of Table 3, which the transfer screen neither confirms nor contradicts.

Arm (diagnostic scale, step 30)	A25	GPQA	Zebra	MMLU
CAP fraction form	.533*	.578	.812	.562
confidence twin (second seed)	.519	.580	.807	.585
answer-text twin (first seed)	.516	.588	.745	.571

Table 8: Transfer of the fraction-form reward and its surface-signal twins, 16k budget, one seed each. AIME 2025 uses 32 samples per problem except *eight; other cells use the sample counts of Table 7. The length penalty and CAP-T are in Table 7 and Table 1.

E Full-Scale Per-Set Detail and Arm Registry

This section gives the per-set detail behind the full-scale comparison, repeats the two narrowest held-out sets on broader draws, stratifies the results by problem difficulty, and lists every RL arm the paper reports.

The per-set results let a reader check the full-scale comparison without relying on an equal-weight average. For every full-scale arm and every held-out set, Table 9 gives the accuracy with a 95% problem-level bootstrap interval, the raw count of correct samples over samples drawn, and the mean response length. The intervals are wide on the 30-problem AIME set and the 50-puzzle ZebraLogic subset. For this reason the main text reads per-set differences alongside the averages and calls the tie with outcome-only training a null result, not an equivalence.

	AIME 2025			GPQA-Diamond			ZebraLogic			MMLU-Pro non-math		
Arm (step 50)	Acc [95% CI]	Correct/n	kTok	Acc [95% CI]	Correct/n	kTok	Acc [95% CI]	Correct/n	kTok	Acc [95% CI]	Correct/n	kTok
Qwen3-8B (initial)	0.463 [0.308, 0.608]	111/240	13.3	0.573 [0.513, 0.631]	907/1584	9.3	0.580 [0.482, 0.672]	232/400	7.4	0.565 [0.473, 0.656]	294/520	2.6
GRPO	0.554 [0.408, 0.700]	133/240	12.7	0.581 [0.522, 0.638]	921/1584	8.6	0.870 [0.800, 0.932]	348/400	7.4	0.575 [0.479, 0.667]	299/520	2.5
Length penalty 0.8	0.483 [0.342, 0.625]	116/240	8.1	0.581 [0.525, 0.639]	921/1584	4.7	0.920 [0.863, 0.968]	368/400	5.2	0.469 [0.375, 0.567]	244/520	1.1
CAP, fraction form + revision	0.546 [0.404, 0.679]	131/240	10.9	0.591 [0.533, 0.648]	936/1584	6.6	0.893 [0.828, 0.948]	357/400	6.7	0.556 [0.460, 0.652]	289/520	1.8

Table 9: Per-set detail behind the full-scale comparison: accuracy with a 95% problem-level bootstrap interval, correct samples over samples drawn (eight per problem), and mean response tokens in thousands. Every cell is generated under the 16k budget.

Because MMLU-Pro and ZebraLogic are the two narrowest held-out sets, we also evaluate the full-scale arms on broader draws of both under the same protocol (Table 10). The MMLU-Pro draw has 1,000 non-math questions of the test split, stratified over categories with seed 12 and scored with four samples per question. The ZebraLogic draw has 200 puzzles drawn with the same seed as the 50 in the main table. Its first 50 are the main-table puzzles, and each puzzle is scored with eight samples.

The ordering of the main table holds on both broader draws. On MMLU-Pro the length penalty loses 8.5 points to GRPO and 6.6 to CAP, while GRPO and CAP are within two points of each other. On ZebraLogic the three RL arms lie within noise of one another. CAP again generates fewer tokens than GRPO on both sets.

	MMLU-Pro test, non-math, 1,000			ZebraLogic, 200 puzzles		
Arm (step 50)	Acc [95% CI]	Correct/n	kTok	Acc [95% CI]	Correct/n	kTok
Qwen3-8B (initial)	0.549 [0.524, 0.576]	2198/4000	4.2	0.596 [0.548, 0.644]	954/1600	7.0
GRPO	**0.578** [0.553, 0.605]	2311/4000	3.9	0.863 [0.825, 0.899]	1380/1600	6.9
Length penalty 0.8	0.493 [0.468, 0.518]	1973/4000	2.0	**0.889** [0.856, 0.920]	1423/1600	5.0
CAP, fraction form + revision	0.559 [0.534, 0.585]	2238/4000	2.9	0.881 [0.846, 0.914]	1409/1600	6.2

Table 10: Broader draws of the two narrowest held-out sets, 16k budget, math-verify or letter scoring, problem-level bootstrap intervals. Bold is best per accuracy column among the RL arms.

The main text also cites three sets that Table 9 does not show. On the tuning-exposed MATH-500 the full-scale arms score 0.730 (GRPO, 4.9k tokens), 0.703 (CAP, 3.4k), 0.724 (length penalty, 2.3k) and 0.710 (CAP-T, 2.4k). On AIME 2024 they score 0.671, 0.667, 0.654 and 0.667, and on AIME 2026 0.558, 0.583, 0.550 and 0.583, in the same order (eight samples, 16k budget). The initial model scores 0.600 and 0.537 on AIME 2024 and 2026.

To see how the arms behave at different difficulty levels, we split the four held-out sets by each problem's solve rate under GRPO (eight samples). Table 11 compares the mean response length and accuracy of the three full-scale arms within each stratum.

GRPO solve rate	**#Probs**	**Mean tokens (k)**			**Accuracy**		
		GRPO	CAP	Length	GRPO	CAP	Length
hard ($\leq 1/4$)	85	9.7	8.0 (-18%)	6.0 (-38%)	0.026	0.078	0.094
medium	69	8.7	7.2 (-18%)	5.2 (-40%)	0.455	0.473	0.399
easy ($\geq 3/4$)	189	6.2	4.8 (-23%)	3.4 (-46%)	0.947	0.925	0.903

Table 11: Full-scale arms stratified by per-problem GRPO solve rate over AIME 2025, GPQA-Diamond, ZebraLogic and MMLU-Pro (step 50, 16k budget). Because strata are defined by GRPO's own samples, GRPO's accuracy is biased toward the extremes of each bin.

Table 12 lists every reinforcement-learning arm the paper reports, with its objective, coefficients, data consumed, seeds and checkpoints available, and the rule that selects the reported checkpoint.

R_{out} is the verifiable outcome reward, λ_3=0.5 multiplies the no-answer indicator in every arm, and ℓ is the character length of the response divided by 24,000 and capped at 1 (ℓ_{tail} the same quantity for the post-commitment span).

The reported checkpoint depends on the scale. Diagnostic arms run 30 steps at batch 60 on the 512-problem subset with an 8k budget and are reported at step 30. Full-scale arms run at batch 128 on the 23k pool with a 16k budget and are reported at their validation-best checkpoint under greedy MATH-500, which is step 50 for every completed arm.

Arm	Objective	Scale, seeds, steps	Reported in
GRPO (outcome-only)	$R_{\text{out}} - \lambda_3 \mathbf{1}$[no answer]	diag. 2 seeds; full 1 seed to 50	Tables 2, 3; App. I
DAPO, Dr. GRPO	their published objectives on R_{out}	diag. 1 seed	Table 1
k1.5 length reward	group-relative length reward on R_{out}	diag. 1 seed	Table 1
ALP	$R_{\text{out}} - \lambda \operatorname{rate}\ell$, group solve rate as dose	diag. 1 seed	Table 1
Length penalty	$R_{\text{out}} - \lambda_{\text{len}}\,\ell$, $\lambda_{\text{len}} \in \{0.4, 0.8, 1.2\}$	diag. 2 seeds at 0.8; full 0.8: 1 seed to 50, 2nd to 25; full 0.4: 1 seed to 25	Tables 1, 2; Fig. 3a; App. I
CAP, targeted form	$R_{\text{out}} - \lambda_2\,\ell_{\text{tail}}$, λ_2=0.8	diag. 2 seeds; full 1 seed to 50	Tables 1, 2, 3 (as CAP-T)
CAP, fraction form	$R_{\text{out}} - \lambda_2\,\phi$, λ_2=0.8	diag. 2 seeds; full 1 seed to 25	Table 3, ablations; App. I
CAP, fraction form + revision	$R_{\text{out}} + \lambda_1\, r_{\text{rev}} - \lambda_2\,\phi$, (0.5, 0.4)	full: seed 1 to 50, seed 2 to 25	Table 2; App. I
Difficulty-scaled targeted	$R_{\text{out}} - \lambda_2 \operatorname{rate}\ell_{\text{tail}}$	diag. 1 seed	Table 7
Probe-free conditioned	$R_{\text{out}} - \mathbf{1}$[correct] $\lambda_{\text{len}}\,\ell$, λ_{len}=0.4	diag. 1 seed	Table 7
Confidence twin	fraction form with states from geometric-mean token probability at τ=0.8	diag. 2 seeds	Table 3
Answer-text twin	fraction form with the boundary at the first sentence by which the rollout's final answer string has appeared	diag. 2 seeds	Table 3
Advantage filter	rollouts with $\phi > 0.7$ reset to the group-mean reward	diag. 1 seed	Table 3
Corrective variant	$R_{\text{out}} - \mathbf{1}$[correct] $\lambda_2\,\phi + \mu\, N(1 \to 2 \mid \text{correct})$	diag. 1 seed	this table only (four-set diagnostic average 0.654, equal to the length penalty)

Table 12: Registry of every RL arm. Table numbers refer to the main text. All arms share the learning rate, KL coefficient, rollout count, data order and evaluation protocol of Section A.

F Early Exit: Protocol and Full Sweep

This section specifies how early exit produces an answer and how we evaluate it. It then reports the rule on MATH-500, on three transfer sets and on three model families, and finally on the trained full-scale policies.

The exit rule closes the thinking segment at the k-th consecutive FINAL prediction. The segment is closed with the model's end-of-thinking marker, and the answer is decoded normally from the truncated context. Traces with no detected commitment run to completion and are scored in full. The rule fires on 93 to 99% of traces. The probe adds one CPU forward pass per sentence, which takes milliseconds against seconds of generation.

Every number in this section is computed on every trace of the evaluation split, with no filtering. Each chain is re-read once with the base model, every sentence is scored by the probe, and the chain is cut at the rule's stopping point, where the answer is decoded. The MATH-500 evaluation split is the 150-problem partition disjoint from the probe's training problems (Appendix B). GPQA, AIME and ZebraLogic never touched the probe in any form. Correctness throughout uses the math-verify equivalence of the main protocol.

These choices replace a filtered analysis from an earlier version. That analysis scored only the traces an attribution pipeline accepted (432 of the 1,200 traces of the 150-problem evaluation split, 36%) with a substring correctness flag. It gave 2.1 points for 52.6% savings on MATH-500 but overstated the transfer sets, where a one-letter or one-word reference answer can appear in the continuation by chance. The filter also changed which traces were scored. The no-chain answer is right on 34.7% of all 1,200 traces, compared with 3.9% of the covered traces, which quantifies the selection the old filters induced.

MATH-500 on Qwen3-14B. On MATH-500 the probe rule costs less accuracy than either alternative at the same savings. Table 14 gives the full set of rules on the 1,200 traces. The 28 traces that never close the thinking segment are kept and scored at the budget. The k=10 rule fires on 99.2% of traces and costs 2.7 points $[-5.2, -0.4]$ for 54.9% of tokens. We compare it with two alternatives matched at the same savings. The first truncates every chain at a fixed fraction of its sentences or tokens. The second is a confidence-triggered rule that stops after ten consecutive sentences whose mean token log-probability exceeds a threshold, with the threshold swept to match the probe rule's savings. Both alternatives cost more, and the confidence rule costs an order of magnitude more. This parallels the confidence-twin null in the training experiments.

Transfer sets. Table 13 runs the same evaluation on the three transfer sets. With the MATH-500 setting (k=10), the rule saves 34 to 42% of tokens there at costs of 6 to 22 points. It beats the confidence rule by large margins on GPQA and AIME 2025 and by 2.6 points on ZebraLogic, which is within noise. It beats fixed-fraction truncation by 5 points on ZebraLogic and by 17 points

on AIME 2025. On GPQA it lies on the fixed-truncation frontier (6.2 points for 40%, compared with 3.8 for 30% and 11.0 for 49%).

These results show that the rule transfers as a ranking of where to stop, but its operating point does not transfer. The rule is at least as good as the confidence rule on every set and matches or beats a position rule at every savings level we test. However, the same k that costs under three points on MATH-500 is too aggressive on sets where the model commits later and where many chains never close (89 of the 240 AIME traces run to the budget). We report this result instead of re-tuning k per set, because re-tuning would require a held-out split of each set.

	GPQA (n=1,584)		**ZebraLogic** (n=400)		**AIME 2025** (n=240)	
Rule	**Cost**	**Saved**	**Cost**	**Saved**	**Cost**	**Saved**
Full chain (accuracy)	0.638	0%	0.940	0%	0.688	0%
Exit k=10	-6.2 [-8.9, -3.5]	40%	-17.2 [-26.5, -9.0]	34%	-22.1 [-32.5, -12.9]	42%
Fixed fraction, matched	-3.8	30%	-22.5	28%	-39.2	50%
Fixed fraction, 50% tokens	-11.0	49%	-35.2	50%	-40.8	50%
Confidence rule, matched	-14.1	42%	-19.8	34%	-40.8	45%

Table 13: All-trace early exit on the transfer sets with the MATH-500 setting, Qwen3-14B, math-verify correctness; costs in accuracy points against the full chain, brackets are 95% cluster bootstrap intervals over problems. Matched rules are the fixed-fraction and confidence-triggered rules closest to the probe rule's savings.

Rule (all 1,200 traces)	**Accuracy**	**Cost [95% CI]**	**Saved**
Full chain	0.939	—	0%
No chain	0.347	-59.2	100%
Fixed 50% of sentences	0.887	-5.2 [-8.2, -2.6]	48.1%
Fixed 50% of tokens	0.869	-7.0 [-10.4, -3.9]	49.1%
Fixed 70% of sentences	0.918	-2.1	28.4%
Confidence rule, matched savings	0.651	-28.8 [-34.1, -23.7]	55.1%
Exit k=5	0.901	-3.8 [-6.8, -1.2]	60.2%
Exit k=10	**0.912**	$-$**2.7** [-5.2, -0.4]	**54.9%**

Table 14: Deployment-level early exit on every MATH-500 held-out trace of Qwen3-14B, math-verify correctness, paired cluster bootstrap over problems. Paired against the probe rule on the same traces, halving by tokens costs 4.3 more points $[0.9, 8.1]$ and halving by sentences 2.6 more $[-0.5, 5.6]$, while both save fewer tokens.

Three model families. Table 15 compares the same rules across three model families, using every MATH-500 held-out trace of each family and each family's own probe.

We report k=10 because its cost stays near three points on the family we diagnose. It does not maximize savings. We chose it by inspecting the Qwen3-14B MATH-500 evaluation traces, so the MATH-500 numbers are in-sample for the choice of k. The transfer sets and the other two families use the same k without re-tuning and are out-of-sample for it. Even on MATH-500 the 95% interval over problems excludes zero, so the rule is not free. A practitioner willing to give up another point can move to k=5 for five more points of savings, and the whole curve is available to tune against a deployment budget.

The spread in savings across sets follows the median boundary position of each set, which is 0.246 on MATH-500, 0.371 on AIME, 0.372 on GPQA and 0.587 on ZebraLogic. A task on which the model commits late has less text after commitment to remove, which is consistent with the savings on ZebraLogic being the smallest.

On-policy exit on the full-scale policies. We also apply the exit rules to the step-50 CAP and GRPO policies of Table 2 (Table 16). The probe is the Qwen3-8B probe used as the training reward, and k=10 is unchanged. Each column uses all 1,200 traces (eight per problem) of the 150 held-out MATH-500 problems at a 16k budget.

	Qwen3-14B (n=1,200)		**Llama-8B** (n=600)		**Phi-14B** (n=600)	
Rule	**Acc**	**Sav**	**Acc**	**Sav**	**Acc**	**Sav**
Full chain	.939	0%	.868	0%	.957	0%
No chain	.347	100%	.127	100%	.317	100%
Fixed 50% of sentences	.887	48%	.620	49%	.805	45%
Fixed 70% of sentences	.918	28%	.768	29%	.912	25%
Fixed 80% of sentences	.926	19%	.807	19%	.937	16%
Fixed 50% of tokens	.869	49%	.607	49%	.773	48%
Confidence, matched	.651	55%	.630	36%	.840	25%
Exit k=5	.901	60%	.790	46%	.908	54%
Exit k=10	**.912**	**55%**	**.795**	**39%**	**.918**	**44%**

Table 15: Early exit on every MATH-500 held-out trace of three families, math-verify correctness. Accuracy and the fraction of chain-of-thought tokens saved. The confidence rule stops after ten consecutive sentences whose mean token log-probability exceeds a threshold; the threshold shown matches the probe rule's savings most closely. Bold is the operating point the main paper reports.

On both policies the probe rule still beats fixed truncation at neighbouring savings. Fixed truncation becomes expensive on the CAP policy, whose chains are already 35% shorter (2,594 against 4,007 reasoning tokens). Halving its chain costs 26 points, because little redundant tail remains.

After exit, the two policies can be compared directly. The mean remaining chain is 1,892 tokens for CAP and 1,859 for GRPO. Accuracy differs by $+1.7$ points $[-0.6, +4.1]$ and the token count by $+33$ $[-40, +102]$ (paired problem-level bootstrap, 2,000 resamples). These samples are larger than those of the drift audit in Appendix J, which scores 44 to 103 traces per arm from a 100-problem regeneration, so the two sets of exit numbers are not directly comparable.

	CAP step 50		**GRPO step 50**	
Rule	**Acc (cost [95% CI])**	**Sav**	**Acc (cost [95% CI])**	**Sav**
Full chain	.929	0%	.932	0%
Fixed 50% of sentences	.667 (-26.2)	49%	.853 (-7.9 [-11.2, -4.8])	49%
Fixed 50% of tokens	.662 (-26.7)	48%	.850 (-8.2)	49%
Fixed 70% of sentences	.863 (-6.7 [-10.6, -2.8])	29%	.914 (-1.8)	29%
Fixed 80% of sentences	.888 (-4.1)	19%	.921 (-1.2)	19%
Confidence, matched	.750 (-17.9)	28%	.641 (-29.2)	52%
Exit k=5	.892 (-3.7)	31%	.858 (-7.4)	59%
Exit k=10	**.908** (-2.2 [-4.6, $+0.3$])	**23%**	**.891** (-4.2 [-7.1, -1.6])	**52%**

Table 16: Early exit on the trained full-scale policies, every MATH-500 held-out trace, math-verify correctness, paired cluster bootstrap over problems. The probe is the frozen Qwen3-8B probe that supplied the CAP reward; the GRPO policy never saw it.

G Perturbation Dose-Response

This section reports the perturbation test behind the boundary claim. It gives the full dose-response curve, a position-matched re-analysis, and a control that holds the readout location fixed.

At the 50% dose used in the main text, the answer survives corruption after the boundary more often than corruption before it. Post-boundary corruption preserves the answer in 88.8% of runs on Qwen3-14B, 92.7% on Phi-4-reasoning-plus and 57.4% on the Llama distill. The same corruption before the boundary, read at the boundary, preserves it in 48.3%, 38.2% and 48.8%.

The corruption procedure is the same at every dose. Every numeric literal in the target region is a candidate, and the requested fraction of them is sampled uniformly at random. Each sampled literal is replaced by a different number of the same magnitude, and the model continues from the corrupted prefix. The pre-boundary condition is measured on the same traces truncated at the boundary, which

reads the answer the model has settled on at that point (unedited prefixes reproduce the final answer in essentially every trace). The post-boundary and whole-chain conditions read the answer after the full chain. In the whole-chain condition, both regions are edited and the tail is left in place.

Table 19 gives the full dose-response curve for Phi-4-reasoning-plus, where each cell averages three independent corruptions of each of 419 traces. Corrupting literals after the boundary is close to free until almost all of them are corrupted, while the same corruption before the boundary costs accuracy from the first dose. The whole-chain column stays close to the post-boundary column at low and moderate doses, so an intact tail lets the end readout survive corruption of the prefix it was computed from.

A position-matched re-analysis shows that position in the chain does not produce the asymmetry. Position is a possible confound, because the pre-boundary region always precedes the post-boundary region. The re-analysis uses the same records (Table 17). Traces whose boundary falls early put post-boundary edits in the middle of the chain, and traces whose boundary falls late put pre-boundary edits there. Comparing the two at the same relative position keeps the commitment contrast and removes the position contrast. The gap survives on all three models. Within either side of the boundary, moving the edited region along the chain shifts preservation by at most eight points.

Model	Post edits, early bound.	Pre edits, late bound.	Matched gap
Qwen3-14B	0.899 (813)	0.533 (75)	+41.0 [33.1, 48.9]
Phi-4-reasoning-plus	0.953 (492)	0.315 (327)	+60.8 [55.6, 65.7]
Llama-8B (R1 distill)	0.596 (525)	0.486 (543)	+12.3 [5.5, 19.0]

Table 17: Position-matched perturbation check at the 50% dose. Answer preservation for post-boundary edits in chains that commit before 30% of their length against pre-boundary edits in chains that commit after 50%, so both edited regions sit in the middle of the chain (record counts in parentheses). The matched gap is the difference in points over edits whose mean position lies between 20% and 60% of the chain, with a 95% cluster bootstrap interval over traces. The Qwen late-boundary cell rests on 25 traces.

Readout-matched control. The position-matched re-analysis cannot match the readout location, which is tied to the condition by construction, and this control shows that the readout location is what produces the gap. We re-ran the test on Qwen3-14B with the answer read at the same point in every condition. This point is the end of the first post-boundary sentence that lies at least halfway through the post-boundary tail (median at 63% of the chain). The control covers 383 traces of 88 MATH-500 problems, with three corruptions each at the 50% dose (Table 18).

Read at this common point, both conditions preserve the answer almost always. The unedited chain reproduces the final answer on every trace. Pre-boundary corruption preserves it in 99.7% of runs and post-boundary corruption in 96.4%, and matching the number of edited literals between the two regions leaves them within a point of each other. Moving the readout of the pre-boundary condition from the boundary to this later point is enough on its own to raise preservation from 48.0% (on these 383 traces; 48.3% on the full set) to 99.7%. The answer string appears verbatim inside the post-boundary tail before the readout point in 90.9% of traces (median seven occurrences). A readout placed after the boundary can copy this restated answer, so the original pre/post contrast measures where the answer is read. It does not show which text is necessary.

Taken together, the perturbation evidence supports two statements and no more. First, the answer is fixed by the boundary, since an unedited prefix cut there reproduces it. Second, corrupting the text after the boundary rarely changes it. The evidence does not show that the text before the boundary is uniquely load-bearing once the tail exists, and the main text is worded accordingly.

H Paired Arms on Transfer Families

This section tests whether the CAP reward helps on the two model families outside Qwen. Table 20 pairs CAP with GRPO (and, on Llama, a length arm) under the diagnostic recipe and the 16k evaluation protocol.

Condition (Qwen3-14B, 50% dose)	Readout	Preserved
No edit	mid-tail	1.000 [0.990, 1.000]
Pre-boundary edits	mid-tail	0.997 [0.992, 0.999]
Post-boundary edits (before readout)	mid-tail	0.964 [0.952, 0.974]
Pre-boundary edits, count-matched	mid-tail	1.000
Post-boundary edits, count-matched	mid-tail	0.991 [0.984, 0.995]
Post-boundary edits (before readout)	chain end	1.000
Pre-boundary edits, chain intact	chain end	0.999
Pre-boundary edits (original protocol)	boundary	0.480
Post-boundary edits (original protocol)	chain end	0.888

Table 18: Readout-matched perturbation control on 383 Qwen3-14B MATH-500 traces (three corruptions each; Wilson 95% intervals). With the readout held fixed after the boundary, corrupting either region alone is nearly harmless; the paired post-minus-pre difference is -3.3 points $[-5.5, -1.6]$ over problems, and -0.9 $[-1.6, -0.3]$ with edit counts matched. The last two rows repeat the original protocol on the same traces.

Literals corrupted	Post-boundary	Whole chain	Pre-boundary
10%	0.994	0.986	0.841
20%	0.985	0.958	0.685
50%	**0.927**	**0.868**	**0.382**
80%	0.628	0.384	0.220
100%	0.266	0.045	0.134

Table 19: Answer preservation on Phi-4-reasoning-plus as a growing fraction of numeric literals is corrupted in each region, 419 traces per cell. Bold is the row the main paper reports.

The two families give opposite results. On Llama, whose probe is the weaker of the two, no reward separates from any other. The four-set averages span 0.413 to 0.431, a band that evaluation noise alone can produce. On Phi, whose probe is the strongest by three-way accuracy, CAP beats GRPO on three of four sets and lifts the average from 0.701 to 0.714. The two families differ both in probe accuracy and in how inert their post-boundary text is (Appendix G). This contrast parallels, across families, the confidence-twin null.

I Step-25 Completeness Checks

This section reports three checks behind the robustness claims of the main text. All three are run at a matched step-25 snapshot of full-scale training under the main protocol (Table 21).

The first check concerns seed stability. A second CAP training seed averages 0.672, compared with 0.668 for the first seed, and the two seeds stay within 0.9 points of each other on the noisy AIME column.

The second check attributes the effect to the penalty. A pure-penalty arm scores 0.666, compared with 0.668–0.672 for the launched joint form. The small revision bonus in the full-scale configuration therefore has no detectable effect at this snapshot, which is consistent with the penalty on post-commitment text doing the work.

The third check shows that the size of the length penalty matters little at this snapshot. A milder $\lambda_{\text{len}}{=}0.4$ arm reaches 0.678, within two tenths of a point of the 0.8 arm's second seed. Both length arms sit about a point above CAP on this average at step 25. By step 50 the 0.8 arm has given back its AIME gain while CAP holds, so the separation on the hard sets in the main text arises between the two snapshots. This reading rests on a single trajectory per arm, a limitation the main text states.

Family	Arm	MATH	AIME25	GPQA	Zebra	Average
Llama-8B (probe 0.758)	no RL	0.660	0.275	0.403	0.313	0.413
	GRPO	0.658	0.267	0.414	0.365	0.426
	Length	0.644	0.258	0.397	0.383	0.420
	CAP (ours)	0.660	0.263	0.406	0.395	0.431
Phi-14B (probe 0.880)	no RL	—	0.579	0.576	0.763	—
	GRPO	0.793	0.604	0.606	0.800	0.701
	CAP (ours)	**0.802**	**0.638**	**0.626**	0.790	**0.714**

Table 20: Our reward (CAP, bold rows) against its controls on two transfer families; Average is the mean of the four sets shown. The weaker probe (Llama) yields no separation; the stronger (Phi) beats GRPO on three of four sets.

Arm (step 25)	**MATH**	**AIME25**	**GPQA**	**Zebra**	**Average**
CAP seed 1	0.690	0.563	0.566	0.853	0.668
CAP seed 2	0.690	0.554	**0.598**	0.848	0.672
CAP pure penalty	**0.694**	0.546	0.583	0.840	0.666
Length λ_{len}=0.4	0.692	0.554	0.590	**0.875**	0.678
Length seed 2	**0.694**	**0.579**	0.590	0.855	**0.680**

Table 21: Completeness checks at a matched step-25 snapshot of full-scale training, main protocol. The second CAP seed lands within half a point of the first, the pure penalty matches the launched joint form, and the two length penalties track each other. At this snapshot the length arms lead on this average; the separation on the hard sets reported in the main text appears between steps 25 and 50. Average is the mean of the four columns.

J PROBE DRIFT AUDIT

This section tests whether the policy that the frozen probe trains learns to game it. Such a policy could in principle learn to delay or disguise commitment.

The audit relabels fresh traces from each trained policy independently of the probe. After training, we regenerate 100 problems $\times$ 4 traces from each trained checkpoint, and the full force-decoding labeling pipeline of the main paper relabels them from scratch. The three audited policies are the diagnostic-scale CAP fraction-form arm at step 30, the diagnostic-scale length-penalty arm (λ_{len}=0.8) at step 30, and the full-scale outcome-only GRPO arm at step 50. The outcome-only policy therefore differs from the other two in scale as well as in reward, and comparisons across arms should be read with this in mind. We then score the frozen probe against these fresh labels and re-run the whole stopping-rule sweep on the trained policies (Table 23).

Exit remains functional on all three arms. At k=10, the setting the main paper deploys, the trained CAP policy gives up 2.5 accuracy points for 31% of its tokens, the length policy 2.9 points for 33%, and the outcome-only policy 6.8 points for 41%. The more important result is that the probe rule still beats fixed-fraction truncation at matched savings on every trained policy. Near 47% saved, it reaches 0.593 against 0.568 on CAP and 0.641 against 0.544 on the length arm. On the outcome-only arm it matches the fixed rule's accuracy while saving nine points more. A probe that had been gamed into meaninglessness would have lost that advantage.

Four caveats apply to these numbers. First, the curves are not monotonic in k. With 44 to 103 traces per arm, individual points are loose by several accuracy points, so we read the shape of each curve and not any single cell. Second, the trace counts differ across arms because a trace enters the pool only when relabeling yields a boundary at all. This is a selection the audit cannot remove. Third, the accuracies in Table 23 use the audit pipeline's strict string match, which is stricter than the math-verify equivalence of the main protocol. Under math-verify the full-chain accuracies rise to 0.889, 0.922 and 0.932, and the k=10 costs become 1.2, 1.0 and 6.8 points, with the outcome-only arm still the most costly. Fourth, the traces that enter the audit are those the attribution pipeline could process, which excludes the longest chains and most chains from the outcome-only arm.

Gaming would leave one of two signatures, and we find neither specifically under CAP. First, a policy could defer commitment. This would push the measured boundary later while ϕ falls and total length holds. The training logs, which record mean ϕ, mean r_{rev} and response length per step, show that pattern in none of our arms. Second, a policy could learn representations the frozen probe misreads. This would show up as a drop in probe accuracy against freshly force-decoded labels that is specific to the CAP policy, and the relabeled traces above check for it. Probe accuracy drops on every trained policy, including the outcome-only one (Table 22), so this drop is not specific to CAP. The absence of these two patterns does not rule out a form of gaming we did not anticipate.

	CAP	**Length**	**GRPO**
Traces / sentences	81 / 5,664	103 / 6,832	44 / 3,523
Three-way accuracy (%)	67.7 [64.0, 71.7]	74.1 [70.6, 77.3]	70.7 [66.4, 75.3]
Macro-F1	0.508	0.534	0.486
FINAL precision (%)	91.6	94.7	93.7
FINAL recall (%)	72.5	77.4	76.2
Probe boundary lag (sentences)	+11	+10	+13
Fires before true boundary (%)	11.1	3.0	9.1
Sentences per trace	69.9	66.3	80.1
Tokens per sentence	19.7	19.4	19.4
Mean $P(\mathrm{FINAL})$	0.41	0.43	0.46

Table 22: Frozen-probe calibration against fresh force-decoded labels on each trained policy. Brackets are 95% trace-level bootstrap intervals. The pre-RL held-out reference is 80.4% three-way accuracy with 96.1% FINAL precision. The probe degrades on every trained policy, including the one that never saw it, errs late rather than early, and sees similar sentence structure on all three arms.

	CAP		**Length**		**GRPO**	
Rule	**Acc**	**Sav**	**Acc**	**Sav**	**Acc**	**Sav**
Full chain	.667	0%	.689	0%	.727	0%
No chain	.086	100%	.029	100%	.159	100%
Fixed 50%	.568	47%	.544	46%	.614	47%
Fixed 80%	.642	18%	.660	18%	.682	19%
Exit k=1	.630	51%	.650	50%	.568	59%
Exit k=2	.593	48%	.641	47%	.614	56%
Exit k=5	.630	40%	.650	40%	.682	49%
Exit k=10	**.642**	**31%**	**.660**	**33%**	**.659**	**41%**

Table 23: The stopping-rule sweep re-run on each trained policy after relabeling its regenerated traces from scratch (n=81, 103 and 44 traces). The probe rule stays ahead of fixed-fraction truncation at matched savings on every arm, which a probe gamed by its own reward could not do. Bold is the operating point the main paper reports.

The calibration check in Table 22 scores the frozen probe sentence by sentence against the fresh force-decoded labels of each trained policy. Its accuracy drops on every trained policy. Three-way accuracy falls from the pre-RL held-out reference of 80.4% to between 68% and 74% on every trained policy, including the outcome-only policy that never saw the probe. This is consistent with a distribution shift induced by RL and not with a probe-specific effect. However, the outcome-only policy was trained at full scale and the other two at diagnostic scale, so the comparison cannot fully separate the reward from the scale. The CAP policy sits 6 points below the length policy on this metric, a gap whose intervals overlap, and it is not separated from the outcome-only policy.

Where the probe errs, it errs late. Precision on the FINAL class stays above 91% on all three arms. The probe's boundary trails the force-decoded boundary by 10 to 13 sentences on average on every arm, and it never fires before the true boundary on more than 11% of traces.

We see no sign that the CAP policy has learned to restructure its text around the probe. Sentence lengths and the mean probability the probe assigns to FINAL are similar across arms, and the CAP policy's sentence count per trace lies between those of the two control arms.