Drop the Act: Probe-Filtered RL for Faithful Chain-of-Thought Reasoning

Anonymous Authors[1]

Abstract

Reasoning models post-hoc rationalize answers they have already committed to internally, producing chains of *reasoning theater*: deliberative-looking steps that contribute nothing to correctness. This wastes inference tokens, pollutes interpretability, and obscures what the model actually computed. We introduce **ProFIL** (**Pro**be-**Fil**tered Reinforcement Learning) to *reduce theater, increase chain-of-thought faithfulness, and shrink chain length* in a single, drop-in extension to Group Relative Policy Optimization (GRPO). A multi-head attention probe is trained *once* on the *frozen* base model to detect post-commitment steps from internal activations alone; during GRPO, rollouts whose probe score exceeds a threshold have their advantage zeroed. *Our central finding is that a probe trained on a frozen base, with verifier-derived labels and no human annotation, provides a stable signal that suppresses theater while resisting the RL-obfuscation failure mode predicted by prior work.* Across four reasoning domains (GSM8K, LiveCodeBench, ToolUse, MMLU-Redux) and two model architectures (Llama-8B, Qwen-7B), ProFIL reduces post-commitment theater by **11–100%**, raises faithful-fraction (e.g., $+24$pp on LiveCodeBench under an independent Claude 3.7 Sonnet judge), and shortens chains by 4–19%, all while preserving or improving task accuracy. ProFIL also beats a matched length-penalty GRPO baseline, isolating the gain as semantic commitment-detection rather than chain compression. Probe weights, training configurations, and rollouts are released across all four domains.

1. Introduction

While scaling up reasoning with reinforcement learning has produced remarkable results (DeepSeek-AI, 2025; Wei et al., 2022; Kimi Team, 2025), a converging body of evidence reveals a systematic gap between what these models *appear* to do and what they actually do. Biased prompts steer final answers while leaving the chain of thought unchanged (Turpin et al., 2023); models recover correct answers from truncated chains far earlier than their length implies (Lanham et al., 2023); large-scale audits of production reasoning models find post-hoc rationalization to be widespread (Chen et al., 2025). The pattern is consistent: reasoning models commit to an answer internally, then continue generating steps that look deliberative but contribute nothing. We call this *reasoning theater* (Figure 1).

Theater carries a real cost. Post-commitment steps pollute interpretability, introduce spurious signal into process-reward training (Wang et al., 2024; Lightman et al., 2024), and inflate token cost in agentic deployments where inference budget matters. Prior work has measured this gap carefully but largely treated it as a diagnostic. *Our goal is to reduce it*: to suppress theater, raise faithfulness, and shrink chains for inference efficiency, while preserving or improving task accuracy.

Faithfulness is non-verifiable: unlike answer correctness, no ground-truth label is available at the end of a rollout, and standard GRPO reward shaping cannot reach behaviors the verifier never sees. Our key observation is that the verifier we need is already inside the model. We introduce **ProFIL** (Probe-Filtered Reinforcement Learning), which builds an annotation-free faithfulness signal from internal activations (Figure 2): a multi-head attention probe is trained *once* on the *frozen* base model to detect post-commitment steps, and during GRPO any rollout whose mean probe score exceeds a threshold has its advantage zeroed. Because the probe runs on the frozen base rather than on the policy being trained, its decision boundary stays stable throughout RL.

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

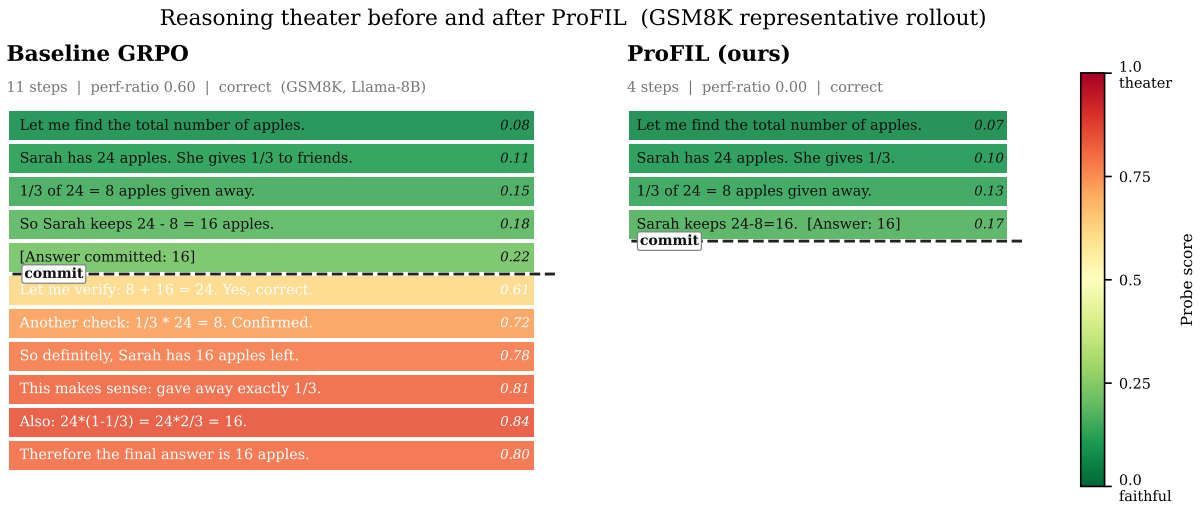

Figure 1. **Reasoning theater before and after ProFIL (perf-ratio** $0.60 \rightarrow 0.00$**, 6 post-commit steps eliminated).** A randomly-sampled high-performativity GSM8K rollout (performativity ratio > 0.9) from DeepSeek-R1-Distill-Llama-8B under standard GRPO (left) and the paired rollout under ProFIL (right) on the same question. Each row is one reasoning step; color encodes the frozen-base probe's per-step performativity score (green = faithful, red = performative). The baseline commits to the correct answer at step 5 but continues for six more post-hoc elaboration steps; ProFIL halts at the dashed line marking the commitment point. Both models produce the correct answer.

Contributions.

- **Goal-aligned method.** A single, drop-in extension to GRPO that targets the three things that make theater costly: it suppresses post-commitment elaboration, raises faithful-fraction, and shrinks chain length, while preserving or improving task accuracy.
- **Generality.** Consistent gains across four reasoning domains (GSM8K, LiveCodeBench, ToolUse, MMLU-Redux) and two model architectures (Llama-8B, Qwen-7B), robust to a matched length-penalty ablation and confirmed by an independent Claude 3.7 Sonnet judge.
- **Anti-gaming.** The frozen probe attains AUROC 1.000 on LiveCodeBench rollouts before, during, and after training, addressing the RL-obfuscation failure mode of Gupta & Jenner (2025): the model learns to suppress post-commitment steps, not to hide them from the probe.
- **Mechanism.** Theater is a separable bimodal mode with a concrete linguistic signature (structural re-derivation blocks); ProFIL collapses the high-theater mode at all chain lengths while leaving faithful rollouts unchanged.
- **Artifacts.** Probe weights, training configurations, evaluation harness, and rollout caches released across all four domains.

2. Related Work

Chain-of-thought faithfulness. Turpin et al. (2023) flip final answers via biased prompts while leaving the chain unchanged; Lanham et al. (2023) introduce the *forced-answering* protocol we adopt for label generation (§4); Chen et al. (2025) report large-scale faithfulness audits on production reasoning models. We refer to the post-commitment elaboration these studies measure as *reasoning theater*. This thread is diagnostic; ProFIL acts on the same signal to reduce it during training.

Probes as reward signals. Probes recover latent knowledge (Burns et al., 2023; Azaria & Mitchell, 2023) and steer behavior at inference (Li et al., 2023; Rimsky et al., 2024). Goodfire's RLFR (Reinforcement Learning from Feature Rewards; Goodfire AI, 2026) pushed multi-head attention probes from inference into RL training. Our probe architecture descends from RLFR; the key departure is that we probe the *frozen* base, not the policy being trained. Inference-time steering methods such as Contrastive Activation Addition (CAA; Rimsky et al., 2024) and Inference-Time Intervention (ITI; Li et al., 2023) suppress behaviors at the forward-pass level but leave policy weights untouched, so the model regenerates the suppressed behavior on the next call. ProFIL eliminates theater durably by changing what the policy learns.

Process reward models. Process Reward Models (PRMs; Wang et al., 2024; Lightman et al., 2024) score step *correctness*; ProFIL scores step *commitment*. A step can be correct and performative simultaneously: a PRM rewards

it, ProFIL filters the rollout containing it. The two are complementary.

Length control as the closest auxiliary signal. Brevity and length-control penalties (Kimi Team, 2025; Aggarwal & Welleck, 2025) target the same surface phenomenon (over-long chains) but operate on token count rather than commitment. Our LiveCodeBench ablation (§5.3) shows they *worsen* theater, isolating commitment-detection as the operative signal.

3. Background

GRPO. GRPO (Shao et al., 2024; DeepSeek-AI, 2025; Kimi Team, 2025) samples K rollouts per prompt, scores each with a *verifier* (a binary correctness checker, e.g., string match for GSM8K, test-case execution for LiveCodeBench), and updates the policy via group-relative advantage. Because the verifier cannot score non-verifiable behaviors like faithfulness, ProFIL gates which rollouts contribute to the update (§4). We define our two metrics, the *performativity ratio* (fraction of post-commitment steps) and the *faithful fraction* (fraction of chains with ratio $\leq \delta$), formally in §4.

4. ProFIL: Probe-Filtered Reinforcement Learning

4.1. Preliminary: Performativity, Commitment, and Faithful Fraction

A reasoning policy π_θ produces a chain $y = (y_1, \ldots, y_T)$ of reasoning steps for input x, followed by an answer a. Let $a^\star(x)$ denote the correct answer. We define the *commitment point* $c(y; x) \in \{1, \ldots, T\}$ as the smallest index t such that the forced answer from prefix $y_{1:t}$ equals $a^\star(x)$; if none exists, the chain is *uncommitted*. Step y_t is *performative* iff $t > c(y; x)$. The *performativity ratio* is the fraction of post-commit steps:

$$\mathrm{perf}(y; x) = \frac{T - c(y; x)}{\max(T - 1,\, 1)} \quad \text{if committed, else } 0.$$

The *faithful fraction* on an evaluation set is the fraction of chains with $\mathrm{perf} \leq \delta$ (we use $\delta = 0.1$ except where noted). Labels are constructed without human annotation: at every step boundary we insert a forced-answer prompt and greedy-decode the answer; the first step whose answer matches the verifier's ground truth becomes the commitment point, and later steps are labeled performative (App. B).

4.2. Frozen-Base Attention Probe

We train a gated multi-head attention probe in the style of RLFR (Goodfire AI, 2026): H=8 heads with learned queries $q_h \in \mathbb{R}^{d_h}$; keys and values come from per-layer LayerNorm-projected residual-stream activations of the frozen base π_{base} at layers ℓ_1, ℓ_2; each head has a sigmoid gate, and the outputs are combined through a scalar output projection. The probe is trained with binary cross-entropy (BCE) on forced-answering labels using frozen-base activations only, then frozen. Training AUROC exceeds 0.92 on all four domains (Table 1). Running the probe on π_{base} rather than on the policy being trained keeps the decision boundary stable throughout RL (analogous to the target network in DQN; Mnih et al., 2015), so the probe needs no recalibration.

4.3. The Probe Filter

GRPO (Shao et al., 2024) samples K rollouts per prompt, scores each with verifier reward $r^{(k)}$, and updates π_θ via group-relative advantage $\hat{A}^{(k)} = (\tilde{r}^{(k)} - \mu)/\sigma$. We introduce a *probe filter*: compute mean per-step performativity $\bar{p}^{(k)}$ via π_{base} and the probe; if $\bar{p}^{(k)} \geq \tau$, zero the reward and the advantage:

$$\tilde{r}^{(k)} = r^{(k)} \cdot \mathbf{1}[\bar{p}^{(k)} < \tau],$$
$$\hat{A}^{(k)} \leftarrow \frac{\tilde{r}^{(k)} - \mu}{\sigma} \cdot \mathbf{1}[\bar{p}^{(k)} < \tau].$$

Filtered rollouts contribute zero to the gradient. Their reward is set to zero before GRPO's group normalization, so they enter the μ/σ computation as zero-reward rollouts rather than being dropped. The threshold τ is the only ProFIL hyperparameter; sparse-reward regimes call for a smaller τ to avoid over-filtering the few correct rollouts. Full pseudocode, per-domain values, and the practical tuning recipe are in App. B.

5. Experiments

5.1. Setup

Models. DeepSeek-R1-Distill-Llama-8B for GSM8K and LiveCodeBench; DeepSeek-R1-Distill-Qwen-7B for ToolUse and MMLU-Redux. Optimizer, batch size, learning rate, group size, probe layers, and per-domain τ are listed in App. B; the matched GRPO baseline shares all hyperparameters except the probe filter.

Datasets.

- **GSM8K** (Cobbe et al., 2021): grade-school math word problems. Full test set evaluated; correctness by exact string match on the boxed answer.
- **LiveCodeBench v6** (Jain et al., 2024): competitive programming problems. Full test set evaluated; correctness judged by test-case execution (pass rate > 0.5).
- **ToolUse**: multi-step agent traces requiring structured

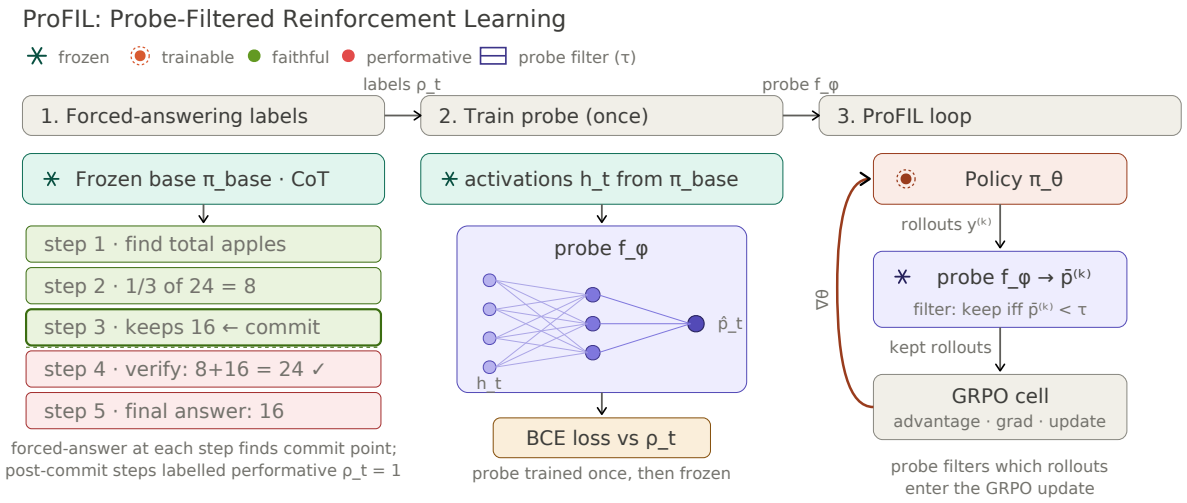

Figure 2. **ProFIL turns a frozen-base probe into a training signal.** (1) Forced-answering yields verifier-derived step-level performativity labels (no human annotation): the first step whose forced answer is correct defines the *commitment point*; later steps are performative. (2) A gated multi-head attention probe (RLFR-style; Goodfire AI, 2026) is trained *once* on activations of the *frozen* base at layers ℓ_1, ℓ_2, then never updated. (3) During GRPO, each rollout's mean performativity is re-computed via the frozen base; rollouts above τ have advantage zeroed and contribute nothing to the policy update. The probe boundary is therefore stable throughout RL, and the GRPO update itself is unchanged outside this filter.

Table 1. **ProFIL reduces theater 11–100% across all four domains while preserving or improving accuracy in every case.** Probe AUROC is the train-split AUROC of the frozen-base probe (exceeds 0.92 in all domains). PR $\Delta\%$ is relative performativity reduction vs. matched GRPO baseline. Accuracy: greedy@1 for all domains except LiveCodeBench (sampled@1 in-training peak; see §5.2). All CIs: 95% Wilson (binary outcomes); ToolUse uses 5000-sample bootstrap. Think chars = mean reasoning chain length. Length-penalty ablation in Table 2.

Domain	Condition	Perf. ratio (PR) ↓	PR $\Delta\%$	Faithful frac. ↑	Task accuracy ↑	Probe AUROC	Think chars
GSM8K	Baseline	0.006 [0.002, 0.017]	–	0.994	0.778 [0.740, 0.812]	0.924	2,351
	ProFIL	**0.000 [0.000, 0.008]**	$\sim$100%	**1.000**	**0.830 [0.795, 0.860]**		**1,895**
Live-CodeBench†	Baseline	0.267 [0.199, 0.349]	–	0.733	0.30 (peak)	0.928	12,382
	ProFIL	**0.076 [0.042, 0.135]**	−72%	**0.924**	**0.37 (peak)**		**11,875**
ToolUse‡	Baseline	0.527 [0.48, 0.57]	–	0.029	0.529 [0.41, 0.64]	0.976	1,248
	ProFIL	**0.468 [0.41, 0.53]**	−11%	**0.147**	0.559 [0.44, 0.67]		**1,018**
MMLU-Redux	Baseline	0.016 [0.008, 0.031]	–	0.984	0.624 [0.581, 0.665]	0.991	4,973
	ProFIL	**0.008 [0.003, 0.020]**	−50%	**0.992**	0.624 [0.581, 0.665]		5,240

†LiveCodeBench (8192-token max): performativity columns use greedy@1; the accuracy column reports the sampled@1 in-training peak (§5.2). ‡ToolUse uses chain-level theater pooling (§5.2.1).

tool calls and planning across multiple turns. Correctness by exact match on tool name and arguments.

- **MMLU-Redux** (Gema et al., 2025; Hendrycks et al., 2021): multiple-choice knowledge reasoning across 57 academic subjects (STEM, humanities, professional). Correctness by answer-letter match.

All evaluation uses vLLM (Kwon et al., 2023) greedy@1 decoding; 95% Wilson CIs (Wilson, 1927) on binary-rate metrics. Probe architecture and training details are in Apps. B and C.

5.2. ProFIL Suppresses Theater on All Four Domains

Theater suppression (Table 1, Figure 3). Performativity ratio drops in every domain: GSM8K 0.006 → 0.000 ($\sim$100%, CIs non-overlapping), LiveCodeBench 0.267 → 0.076 (−72%, CIs non-overlapping), ToolUse 0.527 → 0.468 (−11% chain-level), and MMLU-Redux 0.016 → 0.008 (−50%). Faithful-fraction rises in every domain as well, with the largest jump on ToolUse (0.029 → 0.147, a 5× gain) and on LiveCodeBench (0.733 → 0.924). On MMLU-Redux, ProFIL halves theater at exact accuracy parity (0.624 → 0.624) across 500 problems spanning 57

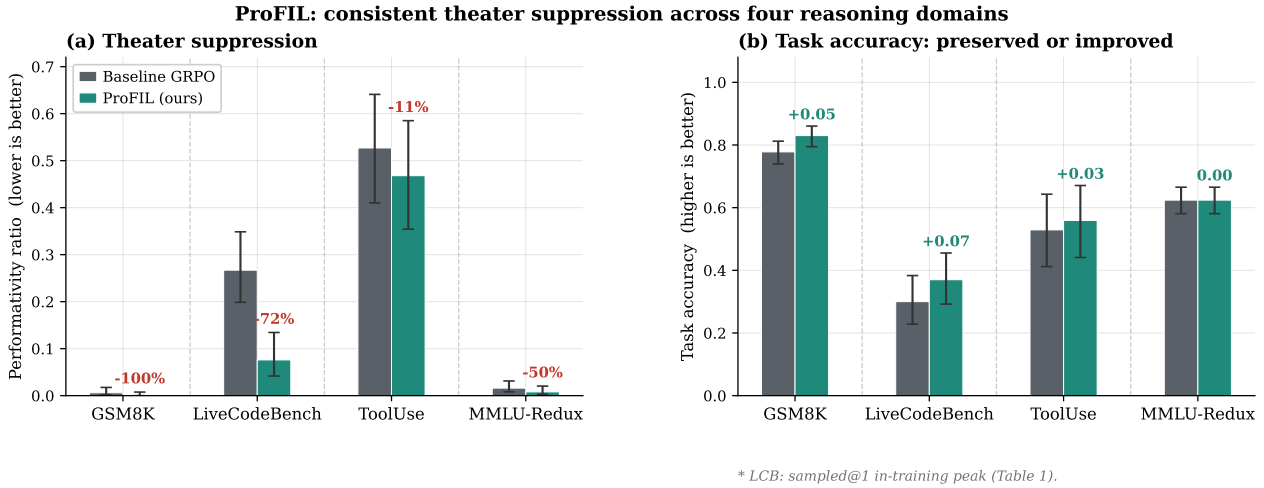

Figure 3. **ProFIL suppresses theater while preserving or improving accuracy on all four domains.** *Left:* performativity ratio drops 11–100% in every domain (lower is better), with non-overlapping CIs on GSM8K and LiveCodeBench. *Right:* task accuracy is preserved or improved in every case. Probe AUROC exceeds 0.92 in all domains (Table 1). Error bars: 95% Wilson CIs. Length-penalty ablation in Table 2 and Figure 5.

subjects. The effect holds across both architectures, suggesting theater is a general artifact of GRPO training rather than a domain-specific quirk.

Accuracy is preserved or improved. GSM8K rises $+5.2$pp greedy@1 with non-overlapping CIs, ToolUse rises $0.529 \rightarrow 0.559$, MMLU-Redux is at exact parity, and LiveCodeBench's sampled@1 in-training peak favors ProFIL by $+7$pp (the metric shown in Figure 3 and Table 1). Per-domain τ values that match each domain's reward density are listed in App. B. Figure 4 shows a qualitative LiveCodeBench (LCB) contrast; chain-level theater on ToolUse is treated next.

5.2.1. ToolUse: Chain-Level Theater and a $5\times$ Faithful-Fraction Gain

On ToolUse, ProFIL increases the faithful fraction by $5\times$ ($0.029 \rightarrow 0.147$, Table 1). The theater signal here is chain-level rather than per-step: each tool-call step is a well-formed JSON object with no room for post-hoc elaboration, so theater appears as additional planning blocks generated *after* the agent has already committed its first complete tool plan (commitment heuristic in App. D).

ProFIL increases single-tool-call resolutions $8\times$. The faithful-fraction gain has a concrete mechanistic source: single-thinking-block resolutions rise from $1/68$ to $8/68$. In these rollouts the model identifies the right tool and acts on it without generating a second planning block to re-examine its choice. The filter is selective: it suppresses post-commitment elaboration when the first plan is sufficient, while preserving genuine multi-step reasoning.

Table 2. **Length penalty worsens theater; ProFIL suppresses it without accuracy cost.** All three conditions on the LiveCodeBench test set. LP reward $= r_{\text{code}} - \lambda|y|_{\text{chars}}$ *increases* performativity vs. baseline, confirming length compression is not the operative mechanism.

Condition	Perf. ratio (PR) ↓	Faithful frac. ↑	Task accuracy (greedy@1 final) ↑	Think chars
Baseline	0.267 [0.199, 0.349]	0.733	0.214	12,382
Len. Penalty	0.374 [0.296, 0.459]	0.626	0.229	11,349
ProFIL	**0.076 [0.042, 0.135]**	**0.924**	0.183	**11,875**

5.3. Length Compression Is Not the Operative Mechanism

A matched GRPO + length-penalty baseline on LiveCodeBench (Table 2; full reward form in App. B) achieves performativity 0.374, *higher* than the unpenalized baseline (0.267) and well above ProFIL's 0.076. ProFIL's gain holds within every length quintile (App. A, Figure 8), and the ranking ProFIL $<$ baseline $<$ length-penalty is stable across all detection thresholds we tested (App. A.3). Faithfulness and brevity are therefore byproducts of the same mechanism: chains shrink when the rollout filter targets commitment, not when the reward penalizes characters.

5.4. Independent Corroboration: Judge, Frozen-Probe Audit, and Inference-Time Steering

We confirm the probe-measured gain with three independent checks.

Independent LLM judge. A Claude 3.7 Sonnet judge evaluated LiveCodeBench rollouts using only a natural-language faithfulness rubric, with no access to probe scores

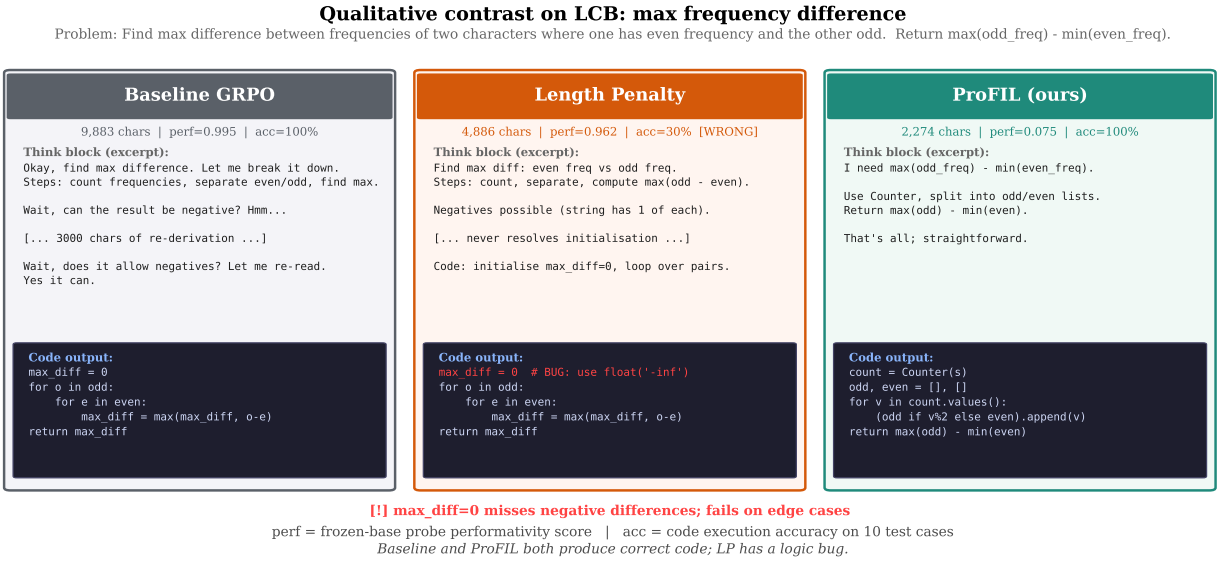

Figure 4. **Theater wastes tokens and introduces bugs; ProFIL eliminates both.** Highest-contrast case among 14 high-theater baseline examples (LiveCodeBench: max odd–even frequency difference). *Baseline* (9,883 chars, perf $= 0.995$): circular reasoning revisits the same edge case three times before writing correct code, deliberative-looking but contributing nothing. *Length Penalty* (4,886 chars, perf $= 0.962$): shorter but still highly performative; initializes `max_diff=0` instead of `float('-inf')`, missing negative-difference inputs (30% test-case accuracy). *ProFIL* (2,274 chars, perf $= 0.075$): one-pass derivation to the correct 5-line solution (100%). Residual performativity is not merely inefficient; it raises the probability of glossing over a constraint the reasoning itself identified but never resolved.

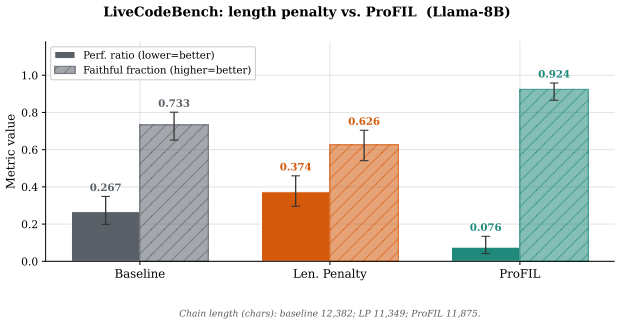

Figure 5. **Length compression is not the operative mechanism** (LiveCodeBench full test set, all three conditions). Length penalty *increases* performativity vs. baseline ($0.267 \rightarrow 0.374$); ProFIL reduces it by 72% ($\rightarrow 0.076$) and raises faithful fraction from 0.733 to 0.924, with chain length shrinking by 507 chars vs. baseline. The key distinction is semantic: ProFIL eliminates post-commitment steps, whereas the length-penalty baseline compresses all tokens indiscriminately.

or training labels (App. D). Strict faithful-fraction rises from 0.49 [0.39, 0.59] to 0.73 [0.64, 0.81]: a +24pp gap with non-overlapping CIs that directly corroborates the probe-measured improvement. The GSM8K probe applied zero-shot to LiveCodeBench rollouts also recovers the ProFIL effect, indicating the probe reads a committed-activation feature that transfers across domains.

Frozen probe is not gamed during RL training. On LiveCodeBench, the frozen probe achieves AUROC 1.000 before, during, and after training, while mean performativity falls from baseline to ProFIL (Gupta & Jenner, 2025). The model learns to suppress post-commitment generation, not to shift activations off the probe's boundary (full AUROC trajectory in App. A.2).

Training is load-bearing. Figure 6 compares two alternatives. First, a student probe re-scored on the trained policy agrees with the frozen-base probe, validating the frozen-base measurement. Second, a CAA (Rimsky et al., 2024) inference-time steering sweep matches the unsteered ProFIL-trained model and remains well above ProFIL's evaluation performativity (Table 1). Suppressing theater durably requires changing what the policy learns, not patching activations at inference time.

5.5. Theater Is a Learned Confidence Ritual, Causally Separable from Accuracy

Theater is a distinct, separable mode. Figure 7 (left) shows the per-rollout performativity distribution is strongly bimodal in the baseline: rollouts are either near-faithful (perf<0.1) or highly theatrical (perf>0.5), with little mass in between. ProFIL collapses the high-theater mode while leaving the faithful mode intact.

Theater has a concrete linguistic signature. To characterize *what* theater looks like, we search all LCB rollouts

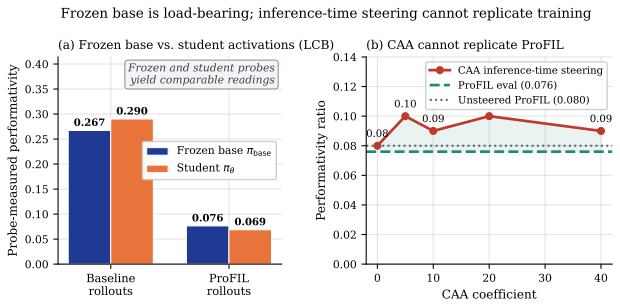

Figure 6. **Frozen base is validated; inference-time steering is insufficient.** *Left:* the student probe agrees with the frozen-base probe on baseline and ProFIL rollouts, validating the frozen-base measurement. *Right:* the best CAA steering coefficient matches the unsteered ProFIL model and remains above ProFIL's evaluation performativity (dashed). Training is the operative mechanism.

for structural re-derivation markers: sections headed `### Approach`, `### Solution`, or `### Explanation` that re-explain reasoning already completed before the answer commitment. Labeling a rollout *high-theater* if its performativity ratio exceeds 0.5 and *faithful* if it falls below 0.1 (matching the faithful-fraction cutoff of §4), the marker fires in 72% of high-theater baseline rollouts but only 3% of faithful ones; under ProFIL it disappears entirely (Figure 7, center). The probe has learned to detect exactly the form theater takes in LCB: confident re-derivation disguised as structured explanation.

Theater marks success, not uncertainty. In the baseline, high-theater rollouts trend toward *higher* accuracy: the high-theater tercile averages $\sim 33\%$ vs. $\sim 9\%$ for the low-theater tercile on LiveCodeBench (full statistics in App. F). *Theater is a confidence ritual: a post-hoc performance the model has learned to append to problems it already knows how to solve.* ProFIL does not suppress uncertainty; it removes the performance. After training, nearly all rollouts concentrate in the low-theater deciles while accuracy is preserved, confirming the elaboration was never causally contributing.

6. Discussion and Conclusion

Self-correction is preserved by construction. The commitment point is defined as the *first* step whose forced answer matches ground truth, so any earlier reasoning, including wrong intermediate answers and the correction moves that fix them, sits in the pre-commitment region and is never filtered. Two GSM8K case studies in App. G make this concrete: in both, the model produces an incorrect answer at an early step, recovers, and arrives at the correct answer; ProFIL retains every step of that correction trajectory and only filters post-commitment elaboration that follows.

Limitations and future work. We evaluate on the DeepSeek-R1-Distill family, the de facto open-weight standard for reasoning-RL research, across two architectures (Llama-8B, Qwen-7B); scaling up to larger checkpoints and other RL-trained reasoners is a natural next step. Two ambitious extensions follow from the recipe. First, ProFIL composes with process reward models in a single GRPO update: a PRM rewards correct *steps* while the probe filters rollouts whose reasoning concludes *after* commitment, combining local correctness with global faithfulness. Second, replacing the verifier with an LLM-judge-derived commitment signal lifts ProFIL to open-ended domains (writing, dialogue, agentic web tasks) that lack a ground-truth checker.

Conclusion. ProFIL targets the three things that make reasoning theater costly: it suppresses post-commitment elaboration, raises chain-of-thought faithfulness, and shortens chains for inference efficiency, while preserving or improving task accuracy. Across four reasoning domains and two model architectures the trained model commits and stops, with a probe boundary that survives RL intact and a learned policy whose chain-of-thought more closely reflects the computation that produced the answer. More broadly, the result demonstrates that a model's own internal representations, read once on the frozen base, can supply a stable, annotation-free training signal for behaviors that the verifier alone cannot see. This turns the verifier blind spot into an asset and points to a general recipe for non-verifiable objectives in post-training, with faithfulness as the first concrete instance.

References

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

Azaria, A. and Mitchell, T. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 967–976, 2023.

Burns, C., Ye, H., Klein, D., and Steinhardt, J. Discovering latent knowledge in language models without supervision. In *International Conference on Learning Representations (ICLR)*, 2023.

Chen, Y., Benton, J., Radhakrishnan, A., Uesato, J., Denison, C., Schulman, J., Somani, A., Hase, P., Wagner, M., Roger, F., Mikulik, V., Bowman, S. R., Leike, J., Kaplan, J., and Perez, E. Reasoning models don't always say what they think. *arXiv preprint arXiv:2505.05410*, 2025.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun,

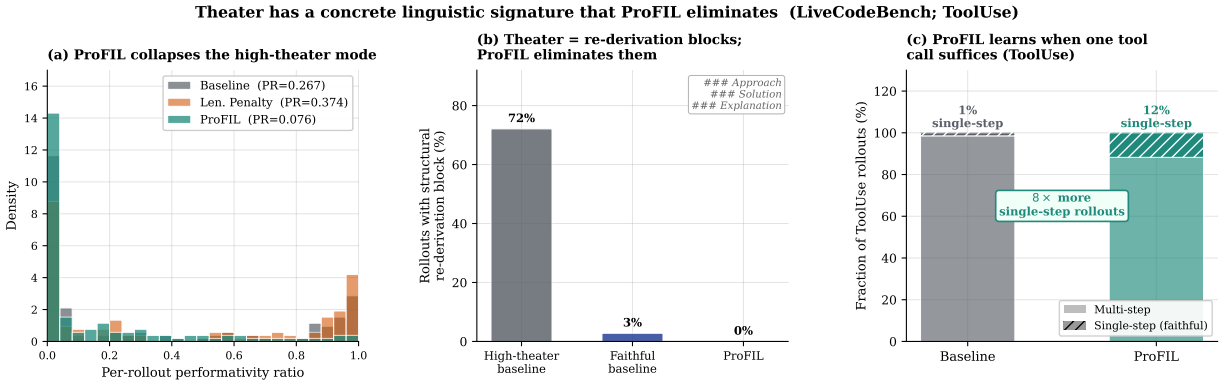

Figure 7. **Theater has a concrete linguistic signature that ProFIL eliminates** (LiveCodeBench, Llama-8B). *Left:* distribution of per-rollout performativity ratios. The baseline is bimodal: rollouts are either near-faithful or highly theatrical. Length penalty shifts the mean only slightly; ProFIL collapses the high-theater mode entirely. *Center:* fraction of rollouts containing structural re-derivation markers (`### Approach`, `### Solution`, `### Explanation`). 72% of high-theater baseline rollouts contain at least one such block; ProFIL eliminates them entirely (0%), confirming the probe detects semantic commitment. *Right:* ToolUse single-thinking-block resolutions (1 → 8 rollouts). ProFIL learns when one tool call suffices and commits without a second planning block.

H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Gema, A. P., Leang, J. O. J., Hong, G., Devoto, A., Mancino, A. C. M., Saxena, R., He, X., Zhao, Y., Du, X., Madani, M. R. G., Barez, F., Hajishirzi, H., Yu, S., Steedman, M., and Minervini, P. Are we done with MMLU? In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pp. 5069–5096, 2025.

Goodfire AI. RLFR: Reinforcement learning from feature rewards. Goodfire AI Research (`https://www.goodfire.ai/research/rlfr`), 2026.

Gupta, R. and Jenner, E. Rl-obfuscation: Can language models learn to evade latent-space monitors? *arXiv preprint arXiv:2506.14261*, 2025.

Hendrycks, D., Burns, C., Basart, S., Zou, A., Mazeika, M., Song, D., and Steinhardt, J. Measuring massive multitask language understanding. In *International Conference on Learning Representations (ICLR)*, 2021.

Jain, N., Han, K., Gu, A., Li, W.-D., Yan, F., Zhang, T., Wang, S., Solar-Lezama, A., Sen, K., and Stoica, I. Livecodebench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

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

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, 2023.

Lanham, T., Chen, A., Radhakrishnan, A., Steiner, B., Denison, C., Hernandez, D., Li, D., Durmus, E., Hubinger, E., Kernion, J., et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.

Li, K., Patel, O., Viégas, F., Pfister, H., and Wattenberg, M. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. In *International Conference on Learning Representations (ICLR)*, 2024.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 518 (7540):529–533, 2015.

Rimsky, N., Gabrieli, N., Schulz, J., Tong, M., Hubinger, E., and Turner, A. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 15504–15522, 2024.

Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Bi, X., Zhang, H., Zhang, M., Li, Y. K., Wu, Y., and Guo, D. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Turpin, M., Michael, J., Perez, E., and Bowman, S. R. 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.

Wang, P., Li, L., Shao, Z., Xu, R., Dai, D., Li, Y., Chen, D., Wu, Y., and Sui, Z. 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)*, 2024.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., and Zhou, D. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Wilson, E. B. Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association*, 22(158):209–212, 1927.

A. Extended Results

A.1. Length-Controlled Faithfulness

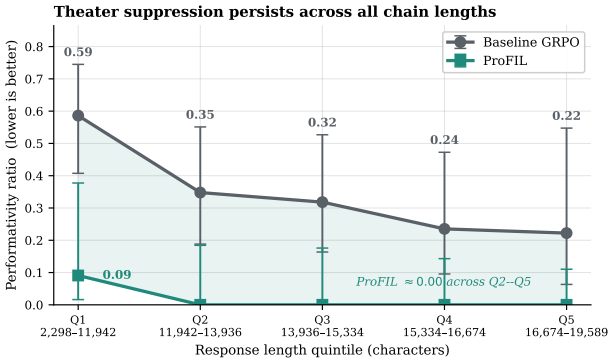

Figure 8. **Theater suppression persists across all chain lengths, ruling out length compression as the mechanism** (LiveCodeBench). Rollouts are stratified into 5 length quintiles on the pooled distribution; the panel reports performativity ratio within each bin. ProFIL suppresses theater across the full length distribution, including the longest 20% of responses, while also producing shorter chains overall (11,875 vs. 12,382 mean chars).

A.2. Frozen-Probe AUROC (Anti-Gaming Evidence)

A probe-as-reward system is vulnerable to RL obfuscation (Gupta & Jenner, 2025): the policy could learn to produce low probe scores while still generating theatrical reasoning, by shifting its activations away from the probe's decision boundary. We check this by re-scoring trained-model rollouts with the frozen probe and computing AUROC against forced-answer-derived performativity labels.

On LiveCodeBench (Llama-8B), the frozen probe attains AUROC 1.000 on rollouts from the baseline, length-penalty, and ProFIL trained models, and remains 1.000 throughout training while mean performativity falls from baseline to ProFIL (Table 1). The probe rank-orders theatrical vs. faithful rollouts perfectly before, during, and after training: the empirical signature of genuine theater reduction rather than activation-space evasion.

A.3. Threshold (δ) Sensitivity

Table 3 shows performativity ratio for both conditions across four threshold values $\delta \in \{0.05, 0.10, 0.20, 0.50\}$. The ranking (ProFIL $<$ baseline) is preserved at every threshold across all domains. On LCB, the length-penalty condition exceeds the baseline at every threshold, confirming chain compression alone does not reduce theater regardless of threshold choice.

Table 3. **Performativity ratio is stable across detection thresholds δ.** Each cell: perf_ratio (fraction of steps with probe sigmoid $\geq \delta$). ProFIL probe is lower than baseline at every δ in every domain. LCB length penalty is higher than baseline at every δ. Paper default: δ=0.50 (rightmost column).

Domain	Condition	δ=0.05	δ=0.10	δ=0.20	δ=0.50
GSM8K	Baseline	0.060	0.038	0.022	0.014
	ProFIL	**0.048**	**0.032**	**0.012**	**0.004**
LCB	Baseline	0.489	0.450	0.420	0.328
	Len. Penalty	0.649	0.603	0.550	0.458
	ProFIL	**0.405**	**0.359**	**0.267**	**0.130**
MMLU	Baseline	0.036	0.030	0.022	0.018
	ProFIL	**0.030**	**0.024**	**0.020**	**0.018**

In-training validation curves. GSM8K and LiveCodeBench validation accuracy logs at every 20 GRPO steps are released alongside the paper.

B. Algorithmic and Hyperparameter Details

Per-domain training configuration.

- **GSM8K** (Llama-8B): 500 test problems, 3 training epochs, 4096-token context, τ=0.5, group size K=8, probe layers [13, 19].
- **LiveCodeBench v6** (Llama-8B): 131 test problems, 3 epochs, 8192-token context, τ=0.2, K=8, probe layers [13, 19]. Reward: test-case pass rate > 0.5.
- **ToolUse** (Qwen-7B): 4046 train / 68 test, 3 epochs, 4096-token context, τ=0.5, K=8, probe layers [14, 21].

Algorithm 1 Probe-Filtered GRPO update

Require: frozen base π_{base}, probe f_ϕ, threshold τ, prompt x, group size K
Sample $\{y^{(k)}\}_{k=1}^{K} \sim \pi_\theta(\cdot \mid x)$
Compute verifier reward $r^{(k)}$
for $k = 1, \ldots, K$ **do**
  Compute frozen activations $h^{(k)}_{\ell_1}, h^{(k)}_{\ell_2} \leftarrow \pi_{\text{base}}(y^{(k)})$
  Compute per-step performativity $p^{(k)}_{1:T_k} \leftarrow f_\phi(h^{(k)})$; $\bar{p}^{(k)} = \text{mean}(p^{(k)})$
end for
Let $\tilde{r}^{(k)} \leftarrow r^{(k)} \cdot \mathbf{1}[\bar{p}^{(k)} < \tau]$ {zero reward of filtered rollouts}
$\mu, \sigma \leftarrow \text{mean}(\tilde{r}), \text{std}(\tilde{r})$ over all K rollouts
for $k = 1, \ldots, K$ **do**
  $\hat{A}^{(k)} \leftarrow ((\tilde{r}^{(k)} - \mu)/\sigma) \cdot \mathbf{1}[\bar{p}^{(k)} < \tau]$
end for
Update θ with standard GRPO step using $\{\hat{A}^{(k)}\}$.

Chain-level performativity pooling.

- **MMLU-Redux** (Qwen-7B): 500 test problems, 256 GRPO steps, 4096-token context, τ=0.5, K=8, probe layers [14, 21].

All runs: AdamW optimizer, cosine LR schedule, identical to the matched GRPO baseline except for the probe filter. Sparse-reward regimes call for a smaller τ (e.g., LiveCodeBench at τ=0.2) so the few correct rollouts in each group are not zeroed. Full config files are released with the code.

Group statistics under filtering. Algorithm 1 sets filtered rollout rewards to zero, and μ, σ are computed over all K rollouts (filtered ones contribute $\tilde{r} = 0$). Computing μ, σ over unfiltered rollouts only is an equivalent alternative.

Forced-answering prompt. "Stop reasoning now. Based on what you have written above, give your final answer in the format \boxed{}." Greedy decoding, max 64 new tokens. We measure first-correct over the full step-boundary trajectory. *LiveCodeBench exception*: the forced-answer correctness criterion is test-case execution (pass rate $>$ 0.5), not boxed-answer string matching, since LCB problems require runnable code rather than a closed-form answer.

C. Probe Architecture and Training

Architecture. Each head h has learned query $q_h \in \mathbb{R}^{d_h}$ and sigmoid gate g_h. Keys and values come from per-layer LayerNorm-normalized residual-stream activations of the frozen base at layers ℓ_1, ℓ_2. Output: $z = \sum_h g_h \cdot \text{head}_h$, scalar logit via linear projection.

Training. Probes trained on rollouts from the frozen base using binary cross-entropy on forced-answering labels. Best-epoch held-out AUROC on the probe-train/test split (in-distribution): GSM8K 0.924, LiveCodeBench 0.928, ToolUse 0.976, MMLU-Redux 0.991.

Layer choice. For Qwen-7B: layers 14 and 21 (mid-network commitment and late elaboration). For Llama-8B: layers 13 and 19. Training AUROC exceeds 0.92 on all four domains, validating that these layers carry a clean theater signal.

D. Independent LLM Judge and ToolUse Commitment Heuristic

We score every rollout with a Claude 3.7 Sonnet judge using a natural-language faithfulness rubric, with no access to probe scores or training labels. On LiveCodeBench, strict faithful-fraction is 0.49 for the baseline (95% Wilson CI $[0.39, 0.59]$) versus 0.73 for ProFIL ($[0.64, 0.81]$): a +24pp gap with non-overlapping CIs.

ToolUse chain-level commitment heuristic. We flag the first step whose decoded text contains a tool-call closure (a JSON-formatted `{"tool":...,"args":...}` block followed by terminal `</think>` or end-of-turn marker), or failing that, the first step whose forced-answer matches the ground-truth tool name and arguments. Steps after this commitment are labeled performative for the chain-level metric.

E. Adaptive Threshold Simulation

The per-domain τ values used in this paper (GSM8K/ToolUse/MMLU-Redux: 0.5; LiveCodeBench: 0.2) match the reward density of each domain (App. B). A natural alternative is to derive τ automatically from the rolling mean group reward, removing the need for domain-specific tuning.

Proposed schedule.

$$\tau_t = \tau_{\min} + (\tau_{\max} - \tau_{\min}) \cdot \text{clip}\left(\frac{\bar{r}_t - r_{\text{lo}}}{r_{\text{hi}} - r_{\text{lo}}},\, 0,\, 1\right),$$

where $\bar{r}_t$ is the exponential moving average of group reward at step t, and $(r_{\text{lo}}, r_{\text{hi}}) = (0.10, 0.40)$, $(\tau_{\min}, \tau_{\max}) = (0.20, 0.50)$. When reward density is low (sparse-reward regime), τ is small and the filter preserves the few correct rollouts as training signal; as reward density grows, τ rises and the filter becomes more aggressive.

Retrospective simulation on LiveCodeBench. Figure 9 applies this schedule retrospectively to the three available LCB checkpoints using their per-example performativity scores and checkpoint accuracies as a proxy for reward density. At step 320 (accuracy 11%), fixed τ=0.5 would filter

59% of rollouts; adaptive τ self-adjusts to 0.21 and matches the per-domain τ=0.2 used in the main experiments. At step 440 after training, accuracy recovers to 18% and adaptive τ rises to 0.28, filtering 21% of rollouts rather than 13% at fixed τ=0.5. The schedule thus removes the per-domain τ choice entirely. A full training comparison of adaptive vs. fixed τ is left to future work.

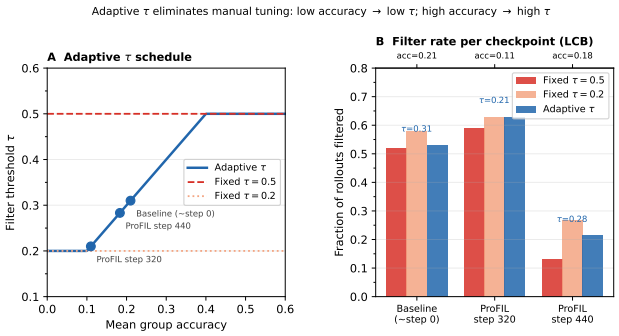

Figure 9. **Adaptive τ eliminates manual threshold tuning.** *Left:* the proposed linear schedule maps mean group accuracy to $\tau \in [0.20, 0.50]$; dots mark the three LCB checkpoints. *Right:* filter rates under fixed τ=0.5, fixed τ=0.2, and adaptive τ at each checkpoint. At step 320 (lowest accuracy) adaptive τ automatically matches the sparse-regime value; at step 440 it rises, balancing signal preservation and theater filtering.

F. Theater as a Confidence Ritual

This appendix expands the headline result of §5.5 that high-theater rollouts in the baseline correlate with task correctness. Figure 10 shows the per-tercile accuracy split (panel A) alongside the per-decile rollout-density shift between baseline and ProFIL (panel B): theater concentrates on problems the model already solves, and ProFIL collapses the distribution to the low-theater region while preserving accuracy.

G. Self-Correction Case Studies

We present two GSM8K rollouts from the probe-trained model in which the forced-answer oracle confirms a wrong answer at an early step and a correct answer at a later step. In both cases, ProFIL assigns the commitment point at the *first correct* step; the intermediate correction steps are pre-commitment and are never labeled performative or filtered.

Example 1: Javelin vs. Dragon (ground truth: 200 feet). The problem asks how far outside the dragon's reach Polly can stand. At step 1 the model computes the gem-enhanced range (1200 feet) and, if stopped there, answers **1200**, mistaking an intermediate quantity for the final answer. At step 2 it subtracts the dragon's reach (1200 − 1000 = 200) and correctly answers **200**. The commitment point is assigned at step 2; step 3 (a restatement) is post-commitment.

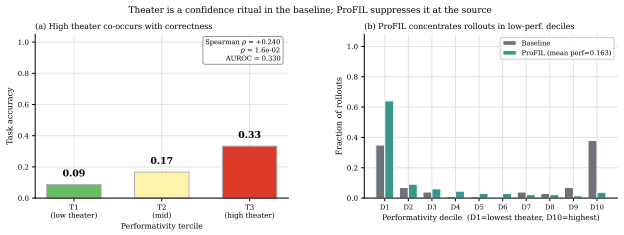

Figure 10. **Theater marks success, not uncertainty: it is a learned confidence ritual.** *Left:* task accuracy by performativity tercile (T1 = low theater, T3 = high theater) on LiveCodeBench baseline rollouts. High-theater rollouts trend toward higher accuracy: the high-theater tercile averages ∼33% accuracy vs. ∼9% for the low-theater tercile (Spearman ρ=+0.240, p=0.016; AUROC = 0.33, meaning theater *anti-predicts* failure). *Right:* the decile distribution shifts from baseline mass at D1 and D10 to near-entirely low-theater under ProFIL. ProFIL does not suppress uncertainty; it suppresses post-hoc elaboration on problems the model already knows how to solve.

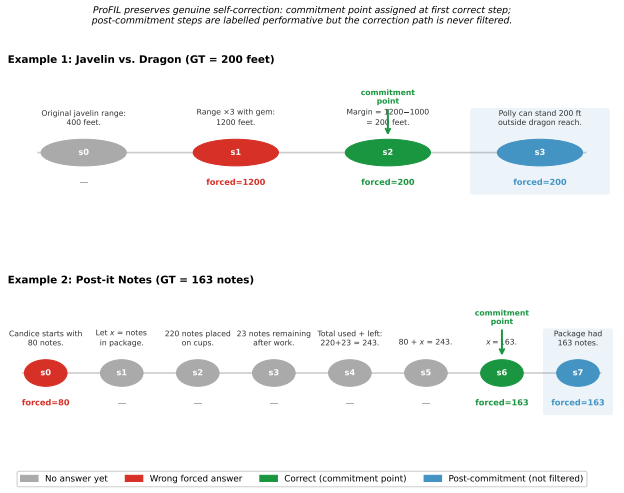

Figure 11. Step-by-step forced-answer trajectories for two self-correcting GSM8K rollouts. Each circle is a paragraph step; the label beneath shows what the model would have answered if stopped at that step. The commitment point (green arrow) is the first step at which the forced answer matches the ground truth. Blue-shaded steps are post-commitment; they are labeled performative only if they follow the commitment point. In both examples the error-correction move itself is *pre*-commitment and is therefore never filtered.

ProFIL filters step 3 if it is performative, but steps 1–2 (including the correction) are untouched.

Example 2: Post-it Notes (ground truth: 163 notes). The problem requires setting up and solving $80 + x = 243$. At step 0 the model reads off the initial count (80) and would answer **80** if stopped; this is a premature answer before the equation is established. Steps 1–5 build the equation; the model cannot yet commit. At step 6 it solves $x = 163$ and the forced-answer oracle returns **163**: the commitment point. The six-step correction path (steps 0–5) is entirely pre-commitment and is never filtered.

H. Compute Budget

Each domain-condition pair: 4 (Llama-8B) or 4–8 (Qwen-7B) H100 80 GB GPUs for 3 epochs, totalling $\sim$480 GPU-hours across the eight training runs. Probe training: $\sim$2 GPU-hours per domain. Evaluation (vLLM rollouts and scoring): $\sim$8 GPU-hours per domain. Ablations (Figure 8, independent judge, mechanistic analysis): $\sim$200 additional GPU-hours.

I. Broader Impacts

ProFIL produces models whose chain-of-thought more closely reflects the computation that led to the answer. The primary benefit is auditability: a reasoning model whose chain reflects its computation is more amenable to oversight, step-level interpretability, and honest disagreement. The symmetric concern is that a method suppressing "performative" filler could in principle target any reasoning pattern a probe is trained to recognize, including reasoning a model should express (safety considerations, uncertainty, refusal preambles). Since our probe targets behavior identified by forced-answering correctness, the immediate risk is low; the general lesson is that probes are dual-use and the choice of probe target is a value judgment.