When Internalization Fails: Finding Better Targets for Reasoning Compression

Mourad Heddaya[*]
University of Chicago
mourad@uchicago.edu

Manley Roberts
Abridge
manley@abridge.com

Rohan Wadhawan
Abridge
rohan.wadhawan@abridge.com

Chenhao Tan
University of Chicago
chenhao@uchicago.edu

Abstract

Reasoning language models generate long reasoning traces that increase latency and cost. We study how to shorten these traces while preserving accuracy on competition-level mathematics. In a teacher-student distillation setup, we compare three approaches: (i) inference-time truncation after the first k tokens, (ii) Implicit Chain-of-Thought (ICoT)-style curricula that progressively shorten the teacher trace during training, and (iii) direct distillation to shorter reasoning traces. Using NUMINAMATH 1.5 with traces from DEEPSEEK-R1 and QWQ-32B, we distill into QWEN2.5-7B and measure accuracy against total tokens generated. We find: (1) with standard SFT and first-k truncation, models compensate by generating longer text after reasoning, undermining token savings; (2) ICoT-style curricula provide little benefit on competition-level mathematics, where reasoning traces are long and diverse; and (3) training on post-think, text the teacher generates after reasoning, achieves the best accuracy–efficiency trade-off among all shortened targets, outperforming generic summaries at matched token budgets. These results show that curriculum-based internalization methods effective on simple tasks do not transfer to complex reasoning, and that post-think provides a better distillation target.

1 Introduction

Reasoning language models generate long chain-of-thought traces to solve hard problems, trading latency and cost for accuracy. In many applications this trade-off is unacceptable: decoding thousands of tokens slows inference, increases serving cost, and degrades user experience. Generating more tokens can even hurt performance through overthinking (Hassid et al., 2025). We ask: *how can we shorten or eliminate reasoning traces while preserving accuracy?*

Prior work has shown that curriculum-based internalization can reduce reasoning length. Implicit Chain-of-Thought–Stepwise Internalization (hereafter ICoT) (Deng et al., 2024) progressively shortens reasoning chains during training, allowing models to internalize computation and eventually answer without explicit reasoning. It works well on GSM8K and multiplication – both tasks with short ($\lesssim$200 token), structured traces. COCONUT (Hao et al., 2024) extends this approach by replacing textual reasoning tokens with a small number of learned latent tokens. These methods are attractive because they eliminate reasoning tokens at inference time, but they have only been validated on simple tasks with homogeneous reasoning patterns. We test whether they scale to competition-level mathematics, where reasoning traces are long ($\sim$5,000 tokens), diverse, and exploratory. We find that ICoT-style curricula provide little benefit over direct distillation on these tasks, indicating that success on short, structured traces may not extend to long, exploratory ones.

Specifically, we use a teacher-student distillation setup with reasoning traces from DeepSeek-R1 (Guo et al., 2025) and QwQ-32B (Team, 2025). We first establish a baseline by distilling on full traces and truncating at inference time after the first k tokens, showing how accuracy degrades as reasoning shortens. We then test ICoT-style curricula that progressively remove segments of the reasoning trace during training. Despite their success on simple tasks, curricula provide little benefit: accuracy is no better than the baseline established by inference-time first-k truncation (Figure 3).

We then explore direct distillation with fully shortened reasoning traces, testing several target strategies: teacher-generated summaries at different lengths, first-k tokens of the original trace, and post-think (text the teacher generates *after* the `</think>` token but before the final boxed answer). Training on post-think achieves the best accuracy–

[*]Work done during internship at Abridge.

Findings of the Association for Computational Linguistics: ACL 2026, pages 14935–14946
July 2-7, 2026 ©2026 Association for Computational Linguistics

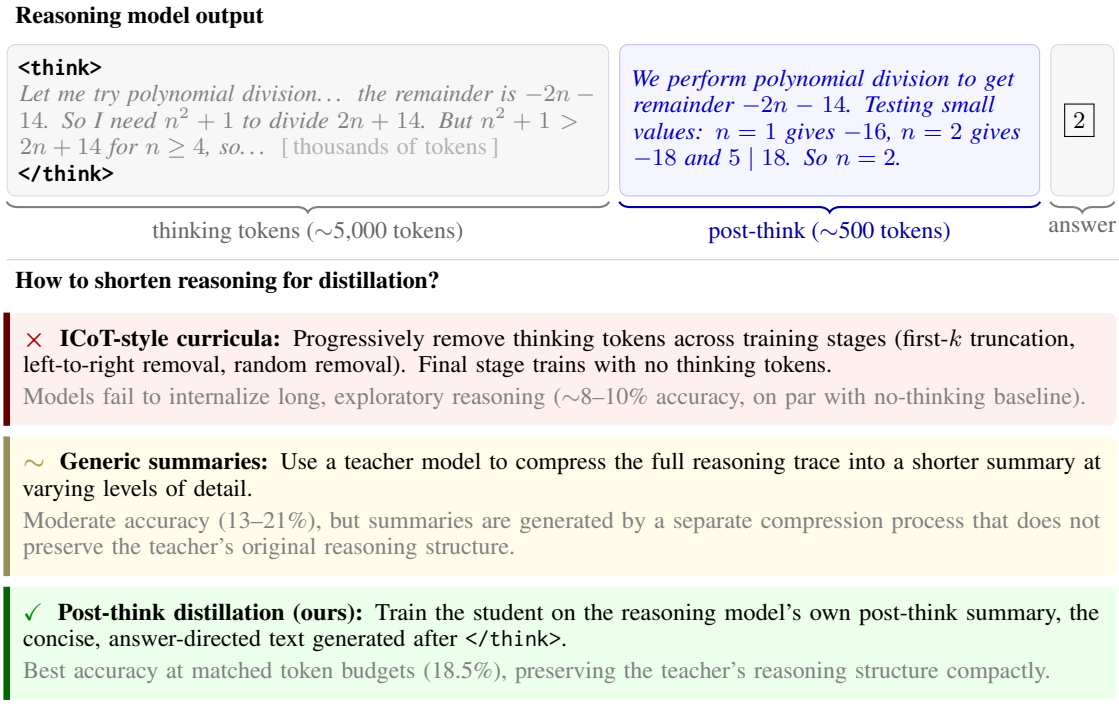


Figure 1: **Distillation approaches for shortening reasoning.** Reasoning models produce long thinking traces followed by a concise post-think summary before the final answer. We compare three strategies for shortening reasoning during distillation: ICoT-style curricula that progressively remove thinking tokens fail to internalize complex reasoning (×); generic summaries achieve moderate results but do not preserve the teacher's reasoning structure (∼); post-think distillation – training on the model's own post-think summary – achieves the best accuracy–efficiency trade-off (✓).

efficiency trade-off, outperforming generic summaries at matched token budgets.

Throughout, when we compare methods at a fixed *token budget*, we mean the *total* number of tokens decoded at inference time, including any continuation after the end-of-thinking marker (e.g., post-think text after </think>). Under standard autoregressive decoding, total token count is directly proportional to inference cost. This distinction matters because models forced to use less reasoning "compensate" by generating much longer post-think traces.

We make three contributions:

- We show that inference-time first-k truncation can mislead about efficiency: models compensate by generating longer post-think text. Visible reasoning shrinks, but total token count remains high.
- We reveal a boundary for internalization methods: ICoT-style curricula that succeed on short, structured traces (GSM8K, multiplication) fail on long, exploratory traces characteristic of competition-level mathematics.
- We show that post-think (text generated after the </think> token) is a better distillation target than generic summaries, consistently achieving the best accuracy–efficiency trade-off across both teachers at matched token budgets. We attribute this to post-think preserving the teacher's solution path.

2 Related Work

Many recent improvements in language model performance have come at the expense of increased inference-time latency and cost. From chain-of-thought prompting (Wei et al., 2023; Kojima et al., 2023) to self-consistency decoding (Wang et al., 2023) to scaling reasoning models, these methods increase inference-time token generation. A growing body of work seeks to improve the token-efficiency of reasoning models.

Implicit and latent reasoning. One line of work aims to *internalize* or *compress* reasoning within the model itself, so fewer or no reasoning tokens are decoded during inference. Stepwise internalization (ICoT-SI) trains models to generate answers directly by iteratively removing reasoning tokens

via curriculum learning (Deng et al., 2024). COCONUT replaces reasoning tokens with a small number of continuous hidden representations that are never decoded (Hao et al., 2024). While both approaches improve over no-reasoning fine-tuning on their original benchmarks, they underperform full-length reasoning and have only been tested on simple tasks with short ($\sim$200 token), structured traces such as multiplication and GSM8K. CODI (Shen et al., 2025) further improves on this line by compressing CoT into a continuous latent space via single-step self-distillation, matching full-length CoT performance on GSM8K while using far fewer tokens, and extending to CommonsenseQA.

However, these approaches have only been validated on tasks with relatively homogeneous reasoning patterns. Whether internalization scales to domains with long ($\sim$5,000 token), heterogeneous, exploratory traces remains an open question. Related approaches include KPOD (Feng et al., 2024), which distills keypoint tokens with progressive curricula and on-policy methods like GKD (Agarwal et al., 2024), which train on student-generated outputs to address distribution mismatch. These are complementary to our focus on compression targets; combining them with post-think training is a promising direction. We test ICoT-style curricula on competition-level math, adapting the ICoT-SI approach of Deng et al. (2024) for longer, more varied traces.

Length control, budgeted decoding, and early exit. A second line of work regulates *how much* a model thinks during training or inference.

Training-time. Xiang et al. (2025) use reinforcement learning with adaptive length penalties to produce shorter reasoning traces while preserving answer quality. Budget Guidance learns a token-by-token predictor of remaining thinking length, softly steering decoding to hit a target budget (Li et al., 2025). Token-Budget-Aware Reasoning predicts an optimal token budget for an (LLM, problem) pair and attaches an explicit budget in the prompt (Han et al., 2025).

Inference-time. Concise-CoT prompting shows that brief, targeted reasoning often suffices (Renze and Guven, 2024). Hassid et al. (2025) find that the *shortest* among parallel chains is frequently the most accurate and propose stopping when the first m chains finish. DEER (Yang et al., 2025) monitors transition cues (e.g., "wait"/branch points) and cuts off generation once a confident answer emerges, yielding 20–80% shorter traces with small accuracy gains.

While both training-time and inference-time approaches can reduce reasoning length on complex tasks, they do not achieve the aggressive reductions we target. Furthermore, many require expensive RL training or additional inference-time components.

Distilling reasoning ability. Our teacher-student setup aligns with distilling step-by-step rationales (Hsieh et al., 2023; Shridhar et al., 2023) and STaR-style self-taught reasoning (Zelikman et al., 2022). Prior works mostly distill *full* rationales; we focus on distillation with *shorter* traces to improve student efficiency.

Analysis of reasoning traces. Recent work analyzes reasoning model traces but restricts attention to text inside `<think>...</think>` (Marjanović et al., 2026; Bogdan et al., 2025; Qian et al., 2025; Bigelow et al., 2025; Liu and Wang, 2025). Some works interact with the `</think>` boundary: Wang et al. (2025) measure first-token entropy after it for early exiting and Yang et al. (2026) exploit it as a token-level trigger to control reasoning budget. However, none study the naturally occurring post-think trace or use it as a distillation target. Chen et al. (2025) use the term "post-thinking" to describe an answer-first generation order, a different meaning from ours.

3 Methods

We study several approaches to shorten inference-time thinking while preserving accuracy on competition-level math.

3.1 Task & Dataset

We use NUMINAMATH 1.5 (Li et al., 2024), a dataset of competition-level mathematics problems from olympiads and contests, covering algebra, geometry, number theory, and combinatorics. Problems typically require multi-step proofs or derivations. We use reasoning traces from two teacher models.

For **DeepSeek-R1** (Guo et al., 2025), we use a subsample of NuminaMath 1.5 from the OPENR1-MATH dataset (Hugging Face, 2025), which pairs 93k problems with full reasoning traces and correct solutions. For **QwQ-32B** (Team, 2025), we use pre-generated traces on the same problem set. For both teachers, we select 10k problems and split them

into 8k training, 1k validation, and 1k test, ensuring matched problem sets for direct comparison.[1] Each teacher response has the following form:

```
[problem]
<think>[reasoning]</think>
[post-think]
\boxed{[answer]}
```

We use `boxed` formatting for answers, enabling systematic extraction.

Post-think. All teacher generations contain a thinking trace inside `<think>...</think>` tags, followed by a **post-think section**: text generated *after* the `</think>` token but before the final boxed answer. Unlike the exploratory reasoning inside the thinking trace, post-think text is a concise, answer-directed explanation, written with knowledge of the final answer and structured as exposition rather than exploration. Because the teacher has already solved the problem, post-think naturally recapitulates its own solution path rather than re-deriving it. This distinguishes post-think from generic summaries, which compress the full reasoning trace via prompting rather than arising naturally from the generation process. Initial experiments showed no accuracy difference between including or excluding post-think when training on full traces. We therefore train on reasoning-only traces (post-think removed) for all experiments except those explicitly distilling post-think.

3.2 Reasoning Distillation

We use a pretrained (base, non-instruction-tuned) `Qwen2.5-7B` checkpoint as our student model. We conduct supervised fine-tuning with LoRA applied to all layers and do early stopping based on validation loss.

We also tried base variants of Gemma 3 4B, Llama 3/3.1 8B, and OLMo 2 7B as students but found them impractical: Llama variants achieved below 5% accuracy even with full thinking traces, OLMo 2 7B is constrained by a 4,096-token context length (too short for our task), and Gemma 3 4B training was prohibitively slow under our available compute. We therefore prioritized varying the teacher model (DeepSeek-R1 and QwQ-32B) over the student model, as generalizing across teachers more directly validates our core claims about training data methodology.

[1]DeepSeek-R1 is licensed under MIT, QwQ-32B under Apache 2.0, NuminaMath 1.5 under Apache 2.0, and OpenR1-Math under Apache 2.0.

3.2.1 Approaches Overview

We compare three approaches for controlling or shortening reasoning length.

First-k truncation (inference-time baseline). At test time we append `</think>` after k reasoning tokens for $k \in \{50, 100, 250, 500, 1000, 1500\}$. This sweep yields a baseline accuracy–length curve. Note: this is distinct from the *first-k tokens curriculum* (a multi-stage training curriculum described below) and from *first-k tokens as a direct distillation target* (single-phase training on fixed-length prefixes).

ICoT-style curriculum learning. We progressively shorten traces during fine-tuning to encourage internalization (Deng et al., 2024). The original ICoT work iteratively removes the leftmost token from the thinking trace until none remain. While feasible for simple tasks ($N \times N$ multiplication and GSM8K), our traces are much longer (often thousands of tokens) and lack the regular step-by-step structure of arithmetic, making token-by-token removal infeasible. With traces averaging $\sim$5,000 tokens, token-level removal yields $\sim$5,000 curriculum stages versus $\sim$30 with segment-level removal (at Δ=1); for a fixed total budget B, this means each stage receives impractically few training steps ($B/5000$). Moreover, free-form reasoning traces lack regular structure: token position n has no consistent semantic role across examples, making token-level removal arbitrary. Segments (double-newline-delimited reasoning steps, $\sim$167 tokens each) are natural units corresponding to complete thoughts. We therefore test four alternative curricula:

- **First-k tokens curriculum:** progressively training on shorter prefixes of the thinking trace (k=1500, 1000, 500, ..., 0 tokens).
- **Left-to-right segment removal:** left-to-right deletion of contiguous segments inside the thinking trace.
- **Random segment removal:** random deletion of contiguous segments inside the thinking trace.
- **Iterative summarization:** replacing the entire thinking trace with increasingly shortened teacher-generated summaries (see Appendix B for more information on the distribution of lengths of these summaries).

Direct distillation to shortened traces. As a non-curriculum alternative, we first train on full reasoning traces, then continue training on a single shortened target. We test the following targets:

- Teacher-generated summaries at six compression levels, where level 1 is longest and level 6 is shortest. Levels 1–3 (median 335–664 tokens for R1) are most comparable to post-think length; see Appendix B for token distributions.
- Official solution explanations from the original NuminaMath 1.5 dataset.
- First-k tokens of the reasoning traces.
- Post-think section from the teacher models.

3.2.2 Segment-Removal Curricula

We split each reasoning trace into segments (separated by double newlines) and progressively remove segments across training stages. At each stage, the student trains on the remaining segments.

Let a teacher reasoning trace inside the `<think>...</think>` tags be split by double newlines (`\n\n`) into

$$T = \langle s_1, s_2, \ldots, s_M \rangle.$$

We index curriculum *stages* by $t \in \{0, 1, 2, \ldots\}$. Each stage is defined by a binary mask $\mathbf{m}^{(t)} \in \{0,1\}^M$, where $m_i^{(t)} = 1$ if s_i is kept at stage t and 0 otherwise. Then, the target distillation trace is

$$Y^{(t)} = \texttt{Concat}\big(\{\, s_i : m_i^{(t)} = 1 \,\}\big).$$

We initialize with $\mathbf{m}^{(0)} = \mathbf{1}$ (all segments kept).

Budgeting and stage schedule. All curricula start from a student trained on full reasoning traces (without post-think), ensuring fair comparison. Table 1 summarizes the key hyperparameters. We fix a total curriculum training budget of B steps (before the final no-thinking phase) and a *step size* Δ controlling how many segments are removed per stage. Following Deng et al. (2024), we cap total removals at κ segments; after reaching κ, we drop all remaining thinking tokens and train on problem $\rightarrow$ answer only until convergence. We choose κ and Δ so the curriculum covers the full reasoning trace for most examples while maintaining a fixed number of stages (see Appendix C.1 for concrete values). Let $S = \lceil \kappa/\Delta \rceil$ be the number of removal stages. We split B evenly so each stage uses

$$N = \big\lfloor B/S \big\rfloor \quad \texttt{steps}.$$

Symbol	Meaning
B	Total curriculum step budget (before no-thinking)
Δ	Segments removed per stage
κ	Max total segments removed before no-thinking
S	Number of removal stages, $S = \lceil \kappa/\Delta \rceil$
N	Steps per stage, $N = \lfloor B/S \rfloor$

Table 1: **Curriculum schedule summary.** Hyperparameters for segment-removal curricula.

3.2.3 Iterative Summarization Curricula

We shorten teacher traces by replacing them with successively shorter, teacher-written summaries that preserve key parts of the solution.

Target lengths. Let T be the full teacher trace inside `<think>...</think>`. We define a schedule of target lengths, $\mathcal{L} = \langle 1500, 1000, 500, 250, 100, 50 \rangle$, and design a distinct prompt for each target to produce summaries of roughly that length using the same teacher model that generated the original trace (i.e., DeepSeek-R1 summaries for R1 traces, QwQ-32B summaries for QwQ traces).

Procedure. Starting from $Y^{(0)} = T$, each stage summarizes the previous stage's trace to the next target:

$$Y^{(t)} = \texttt{Summ}\left(Y^{(t-1)}\right), \qquad t = 1, \ldots, |\mathcal{L}| - 1,$$

where $\texttt{Summ}(\cdot)$ denotes the teacher-generated summary targeting the next length in $\mathcal{L}$. After the $\ell = 50$ stage we remove all remaining content and train only on problem $\rightarrow$ answer pairs.

3.2.4 Additional Training Details

Following Deng et al. (2024), we implement *removal smoothing*: for each training example, with probability 0.05 we randomly remove u additional segments (where u is drawn uniformly from $\{1, \ldots, 5\}$) on top of the current stage's mask, to prevent overfitting to exact stage boundaries. We reset the optimizer state between stages.

Direct distillation training protocol. For direct distillation experiments (post-think, summaries, official solutions, first-k prefixes), we use a two-phase procedure: phase 1 trains on full reasoning

traces (identical to the curriculum starting point), then phase 2 continues training on the shortened target. The LoRA adapter from phase 1 is carried over, but the optimizer state and learning rate schedule are reset. Phase 2 runs until convergence, determined by early stopping on validation loss.

3.3 Evaluation

We generate with temperature 0.3 and a maximum length of 10,000 tokens. The student outputs final answers in `\boxed{}` format, and we evaluate accuracy against ground-truth solutions using HuggingFace's `math-verify` tool.

4 Results

Summary. (1) *Post-think* delivers the best accuracy at matched token budgets; (2) ICoT-style curricula provide *little to no benefit* over direct distillation on long, heterogeneous traces; (3) naive first-k truncation can overstate efficiency because hidden post-think continuation after `</think>` undermines compute savings; and (4) excluding post-think from training exposes a clear length–accuracy trade-off.

Inference-time truncation misleads on efficiency. When post-think is included in training, truncating reasoning after the first k tokens at inference time appears to preserve accuracy. However, models compensate by generating longer text after `</think>`, so shorter reasoning does not reduce total tokens (Figure 2). Post-think grows as thinking shrinks, undermining token savings. This pattern holds for both R1 and QwQ traces (Table 2).

Removing post-think from training reveals the length-accuracy tradeoff. Given this compensation effect, we exclude post-think from all subsequent experiments. This yields a clear baseline: accuracy decreases with reasoning length, dropping sharply when reasoning is removed entirely (Figure 3 and Table 2).

4.1 ICoT-style curricula provide little benefit

We compare four ICoT-style curricula (first-k tokens, left-to-right removal, random removal, and iterative summarization) against direct distillation. Curricula provide little benefit – and several perform no better than the no-thinking baseline (Figure 4a and Table 2). Varying the schedule or removal rate yields no consistent improvement.

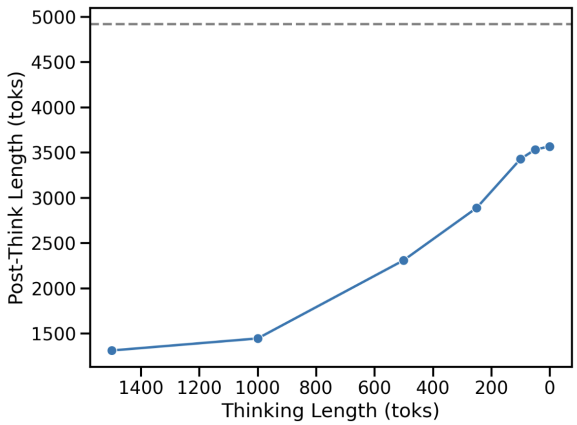


Figure 2: **Models compensate for truncated reasoning (DeepSeek-R1).** When we truncate reasoning at inference time, models generate longer post-think text. As reasoning length decreases, post-think length increases, keeping total token count high. This compensation undermines the apparent token savings from shorter reasoning. QwQ shows similar patterns.

COCONUT (Hao et al., 2024), which replaces reasoning tokens with a small number of latent representations, provides further evidence. Adapting it to our longer traces by removing segments rather than tokens, it achieves only 4.2% accuracy (compared to 8.1% for no reasoning; see Table 2 and Appendix D). Together with our curriculum results, this indicates that internalization methods effective on short, structured traces do not extend to long, exploratory traces characteristic of competition math. Results are consistent across both teachers.

4.2 Post-think outperforms generic summaries at matched budgets

We train students to generate the teacher's post-think. At matched token budgets, post-think consistently achieves the best accuracy–efficiency tradeoff (Figure 4b and Table 2). This holds across both teachers.

Training on official solution explanations from NuminaMath 1.5 (human-written, answer-directed explanations independent of the teacher's reasoning) achieves accuracy comparable to the no-thinking baseline and much worse than post-think, despite also being answer-directed (Table 2). This gap suggests that answer-directedness alone does not explain post-think's effectiveness; we return to this in §5.

Trace generation model	DeepSeek-R1		QwQ-32B	
Method	**Acc.**	**Tokens**	**Acc.**	**Tokens**
Baselines				
Full trace (no post-think)	0.292	6,974	0.293	9,131
No thinking	0.081	9	0.112	9
ICoT-style curricula (final stage)				
First-k tokens curriculum	0.081	9	0.101	8
Iterative summarization	0.102	102	0.120	90
Left-to-right removal	0.071	46	0.086	91
Random removal	0.101	239	0.079	238
COCONUT (Hao et al., 2024)	0.042	8	-	–
Direct Distillation				
Official solution†	0.090	274	0.090	274
Summary level 1	0.187	664	0.168	1,912
Summary level 2	0.145	477	0.211	1,145
Summary level 3	0.134	335	0.164	453
Post-think	**0.185**	**511**	**0.183**	**541**

Table 2: **Key results on Qwen2.5-7B across two teacher models.** Median total tokens reported. ICoT-style curricula show final-stage results. Summary levels 1–3 are most comparable to post-think length (see Appendix B for other levels). Post-think achieves the best accuracy-efficiency trade-off. †Official solutions are teacher-independent, so results are identical across columns. See Appendix D for COCONUT details.

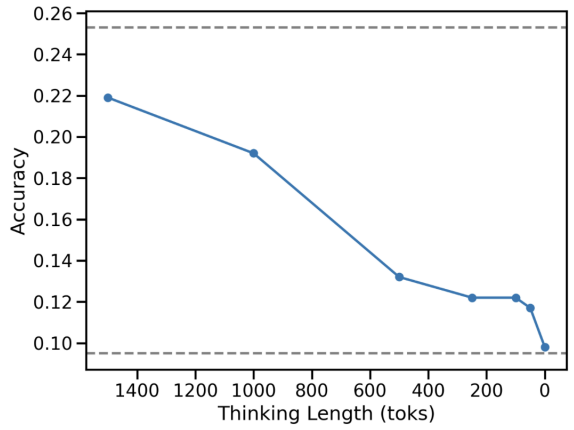


Figure 3: **Accuracy decreases as reasoning shortens (DeepSeek-R1).** When we train without post-think and vary reasoning length with first-k truncation, accuracy decreases monotonically with a sharp drop when reasoning is removed entirely. Dashed horizontal lines indicate full-thinking distillation (upper) and no-thinking distillation (lower) baselines. QwQ shows similar patterns.

4.3 What makes a good summary?

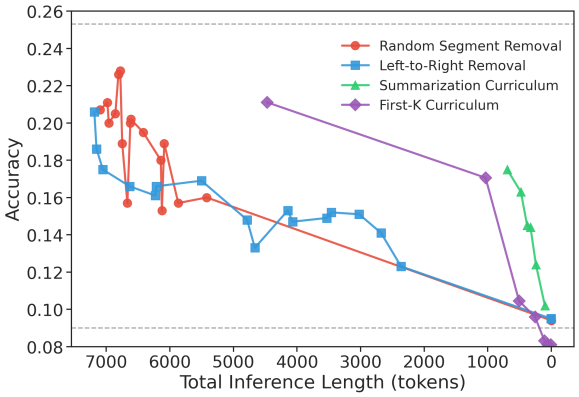
Summaries of similar length can differ substantially in accuracy. The contrast between post-think and official solutions shows that answer-directedness alone is insufficient. To understand post-think's advantage, we compare it against Summary Level 3, the best-performing summary at a comparable token budget for DeepSeek-R1 (Table 2).

Analyzing logical connectives reveals that "therefore" appears 25.8× more frequently per token in post-think than in Level 3 summaries (6.1 vs. 0.2 per 10k tokens), while overall connective density is comparable (0.54 vs. 0.49 per 100 tokens). This suggests that iterative compression strips conclusive deductive markers while retaining general discourse connectives. Post-think's advantage over Level 3 summaries is limited to easier problems (those requiring shorter teacher reasoning traces); neither method achieves meaningful accuracy on harder problems (see Appendix A for detailed breakdown). We discuss possible explanations in §5.

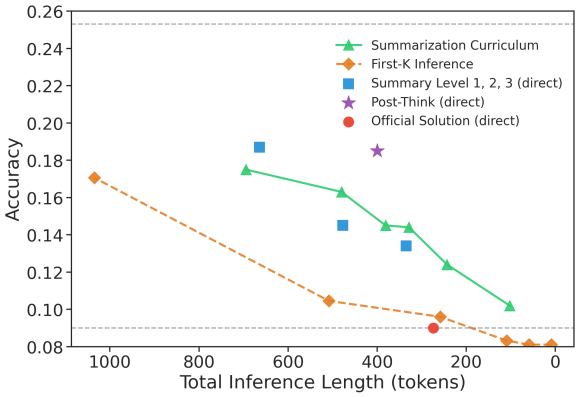
5 Discussion

5.1 Why do curricula fail on competition math?

ICoT-style curricula and latent reasoning methods that work on simple benchmarks fail on competition math. COCONUT performs no better than SFT without reasoning (Appendix D), and curricula show little benefit despite working well on GSM8K (Deng et al., 2024; Hao et al., 2024). We hypothesize two reasons: (1) competition math traces are long (∼5,000 tokens vs. ∼200 for GSM8K) and structurally diverse, so progressive removal disrupts reasoning coherence, unlike arithmetic where steps are uniform and independently meaningful; (2) for latent methods like COCONUT, a small number of latent tokens cannot encode the information in thousands of tokens of exploratory reasoning, whereas short, formulaic traces are more compress-



(a) **ICoT-style curricula fail to internalize reasoning (DeepSeek-R1)**, achieving final no-think accuracy on par with a no-think baseline. Curriculum choice matters little. Accuracy vs. total tokens for four curricula (first-k tokens, left-to-right removal, random removal, iterative summarization). Dashed horizontal lines indicate full-thinking (upper) and no-thinking (lower) baselines.



(b) **Post-think outperforms other distillation targets (DeepSeek-R1).** Accuracy vs. total tokens for direct distillation methods; iterative summarization curriculum included as the best-performing ICoT-style method (see panel a). Post-think summary achieves the best accuracy–length trade-off. Dashed horizontal lines indicate full-thinking (upper) and no-thinking (lower) baselines.

Figure 4: **Comparing shortening methods.** (a) ICoT-style curricula show no consistent benefit over direct distillation. (b) Post-think summary achieves the best accuracy-efficiency trade-off among all methods. DeepSeek-R1 teacher traces shown; QwQ exhibits similar patterns.

ible.

5.2 Why does post-think outperform generic summaries?

Post-think outperforms teacher-generated summaries at matched token budgets (Figure 4b, Table 2). We hypothesize that this stems from differences in contextual role: post-think is generated as a natural continuation of the teacher's reasoning process, serving as a key final step toward the boxed answer. Separately, our inference-time truncation experiments (§4) show that when students are forced to reason less, they compensate by generating longer post-think, suggesting that the model treats this region as functionally important for reaching the answer.

The analysis in §4.3 supports the hypothesis that post-think preserves deductive structure that compression strips away: iterative compression preserves general discourse connectives but strips conclusive deductive markers like "therefore," consistent with compression disrupting the logical scaffolding linking reasoning steps to conclusions. Prior work on step-by-step distillation supports the broader intuition that preserving reasoning structure distills more effectively (Hsieh et al., 2023; Shridhar et al., 2023).

Evidence from official solutions. Official solutions provide additional support for this hypothesis. They are also answer-directed and written with knowledge of the answer, yet perform comparably to the no-thinking baseline. The key difference is that official solutions use human problem-solving approaches that differ from the teacher's. This suggests that successful distillation requires more than knowing the correct answer: the target must reflect a solution path the student can learn to reproduce. We speculate that post-think succeeds because it recapitulates the teacher's own reasoning, while official solutions may fail because they introduce human problem-solving patterns that are less consistent across problems and therefore harder for the student to learn.

6 Conclusion

We study how to shorten reasoning traces while preserving accuracy on competition math. We compare three approaches: inference-time truncation, ICoT-style curricula, and direct distillation to shortened targets.

We find: (1) first-k truncation misleads because models compensate with longer post-think text, undermining token savings; (2) ICoT-style curricula provide little benefit on long, diverse traces, unlike their success on simple tasks; (3) training on teacher post-think achieves the best accuracy–efficiency trade-off, outperforming generic summaries at matched budgets.

Limitations

We focus on competition math, and generalization to other domains remains untested. While we observe consistent patterns across two teachers (DeepSeek-R1 and QwQ-32B), post-think effectiveness may vary with teacher quality. We use 7B-parameter students, and whether larger students can internalize long traces where smaller ones cannot is unclear. We report single runs, so replication across seeds would strengthen statistical conclusions. Our segment-level COCONUT adaptation trades token-level granularity for tractability, and alternative adaptations may yield different results. Our hypothesis about why post-think outperforms summaries rests on indirect evidence (the official solutions comparison), and controlled experiments could isolate the specific properties driving this advantage.

Future Work

Analyzing what makes post-think effective could inform better summary strategies. Testing on other domains (code, scientific reasoning, commonsense) would assess generalization. Combining post-think with ICoT-style curricula may yield further gains. Finally, mechanistic interpretability could reveal whether post-think training internalizes reasoning or pattern-matches surface features (Bai et al., 2025).

References

Rishabh Agarwal, Nino Vieillard, Yongchao Zhou, Piotr Stanczyk, Sabela Ramos, Matthieu Geist, and Olivier Bachem. 2024. On-policy distillation of language models: Learning from self-generated mistakes. In *International Conference on Learning Representations*.

Xiaoyan Bai, Itamar Pres, Yuntian Deng, Chenhao Tan, Stuart Shieber, Fernanda Viégas, Martin Wattenberg, and Andrew Lee. 2025. Why can't transformers learn multiplication? reverse-engineering reveals long-range dependency pitfalls. *Preprint*, arXiv:2510.00184.

Eric Bigelow, Ari Holtzman, Hidenori Tanaka, and Tomer Ullman. 2025. Forking paths in neural text generation. In *International Conference on Learning Representations*.

Paul C. Bogdan, Uzay Macar, Neel Nanda, and Arthur Conmy. 2025. Thought anchors: Which LLM reasoning steps matter? In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Xiaoshu Chen, Sihang Zhou, Ke Liang, and Xinwang Liu. 2025. Distilling reasoning ability from large language models with adaptive thinking. *IEEE Transactions on Neural Networks and Learning Systems*, 36(11):19820–19833.

Yuntian Deng, Yejin Choi, and Stuart Shieber. 2024. From explicit cot to implicit cot: Learning to internalize cot step by step. *Preprint*, arXiv:2405.14838.

Kaituo Feng, Yan Gu, Xuekai Fu, Wenjie Peng, Zheng Yuan, Shuiqiang Huang, and Jingqun Jiang. 2024. Keypoint-based progressive chain-of-thought distillation for llms. In *International Conference on Machine Learning*.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, and 1 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. 2025. Token-budget-aware llm reasoning. *Preprint*, arXiv:2412.18547.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. 2024. Training large language models to reason in a continuous latent space. *Preprint*, arXiv:2412.06769.

Michael Hassid, Gabriel Synnaeve, Yossi Adi, and Roy Schwartz. 2025. Don't Overthink it. Preferring Shorter Thinking Chains for Improved LLM Reasoning. *arXiv preprint*. ArXiv:2505.17813 [cs].

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alexander Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. 2023. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. *Preprint*, arXiv:2305.02301.

Hugging Face. 2025. Open r1: A fully open reproduction of deepseek-r1.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2023. Large language models are zero-shot reasoners. *Preprint*, arXiv:2205.11916.

Jia Li, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. 2024. Numinamath. [https://huggingface.co/AI-MO/NuminaMath-1.5](https://github.com/project-numina/aimo-progress-prize/blob/main/report/numina_dataset.pdf).

Junyan Li, Wenshuo Zhao, Yang Zhang, and Chuang Gan. 2025. Steering llm thinking with budget guidance. *Preprint*, arXiv:2506.13752.

Xin Liu and Lu Wang. 2025. Answer convergence as a signal for early stopping in reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 17896–17907.

Sara Vera Marjanović, Arkil Patel, Vaibhav Adlakha, Milad Aghajohari, Parishad BehnamGhader, Mehar Bhatia, Aditi Khandelwal, Austin Kraft, Benno Krojer, Xing Han Lù, Nicholas Meade, Dongchan Shin, Amirhossein Kazemnejad, Gaurav Kamath, Marius Mosbach, Karolina Stańczak, and Siva Reddy. 2026. DeepSeek-R1 thoughtology: Let's think about LLM reasoning. *Transactions on Machine Learning Research*.

Chen Qian, Dongrui Liu, Haochen Wen, Zhen Bai, Yong Liu, and Jing Shao. 2025. Demystifying reasoning dynamics with mutual information: Thinking tokens are information peaks in LLM reasoning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Matthew Renze and Erhan Guven. 2024. The benefits of a concise chain of thought on problem-solving in large language models. In *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*, page 476–483. IEEE.

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. 2025. Codi: Compressing chain-of-thought into continuous space via self-distillation. *Preprint*, arXiv:2502.21074.

Kumar Shridhar, Alessandro Stolfo, and Mrinmaya Sachan. 2023. Distilling reasoning capabilities into smaller language models. *Preprint*, arXiv:2212.00193.

Qwen Team. 2025. Qwq-32b: Embracing the power of reinforcement learning.

Xi Wang, James McInerney, Lequn Wang, and Nathan Kallus. 2025. EAT: Entropy after ⟨/Think⟩ for reasoning model early exiting. *Preprint*, arXiv:2509.26522.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *Preprint*, arXiv:2203.11171.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

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

Chenxu Yang, Qingyi Si, Yongjie Duan, Zheliang Zhu, Chenyu Zhu, Qiaowei Li, Minghui Chen, Zheng Lin, and Weiping Wang. 2025. Dynamic early exit in reasoning models. *Preprint*, arXiv:2504.15895.

Wang Yang, Debargha Ganguly, Xinpeng Li, Chaoda Song, Shouren Wang, Vikash Singh, Vipin Chaudhary, and Xiaotian Han. 2026. Mid-think: Training-free intermediate-budget reasoning via token-level triggers. *Preprint*, arXiv:2601.07036.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah D. Goodman. 2022. Star: Bootstrapping reasoning with reasoning. *Preprint*, arXiv:2203.14465.

A Problem Difficulty Analysis

We use the length of the original DeepSeek-R1 reasoning trace as a proxy for problem difficulty: longer traces indicate harder problems requiring more reasoning steps. We split problems into five equal-sized groups (quintiles) based on trace length.

The post-think advantage is statistically significant (McNemar $p < 0.001$) and concentrates on easier problems. For the easiest two quintiles (problems with shortest teacher traces), post-think improves accuracy by +12.1 percentage points (95% CI: [+7.5, +16.6]) over Level 3 summaries. For the hardest three quintiles, the improvement is only +0.7 percentage points and not statistically significant.

This pattern suggests that post-think's deductive structure provides the most benefit for problems where logical reasoning chains are shorter and more tractable.

B Summary Level Token Distributions

We generate summaries at six different target lengths using iterative prompting with teacher models (DeepSeek-R1 and QwQ-32B). These summaries are used in two contexts: (1) for the iterative summarization curriculum, where we progressively train on shorter summaries, and (2) for direct distillation, where we train directly on a single summary level. Figures 5 and 6 show the token count distributions for each summary level across the training set for both teacher models.

For our main results (Table 2), we report levels 1–3 as they are most comparable in length to post-think. Levels 4–6 compress too aggressively to be competitive, reducing traces to under 100 tokens where accuracy is near the no-thinking baseline.

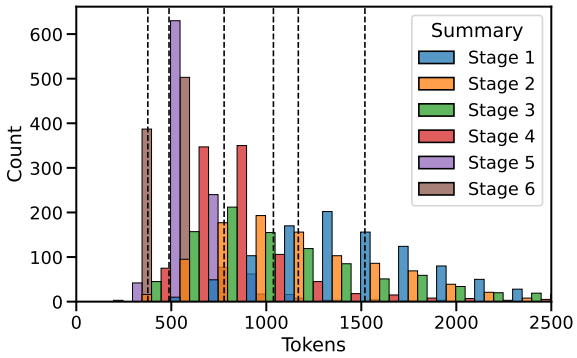


Figure 5: **Token distributions for DeepSeek-R1 summary levels.** Each level represents progressively shorter summaries, with vertical dashed lines indicating median token counts.

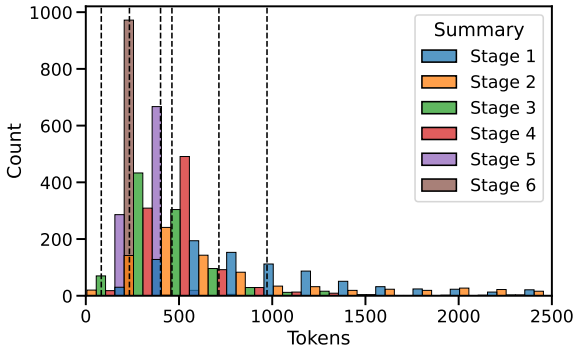


Figure 6: **Token distributions for QwQ-32B summary levels.** Each level represents progressively shorter summaries, with vertical dashed lines indicating median token counts.

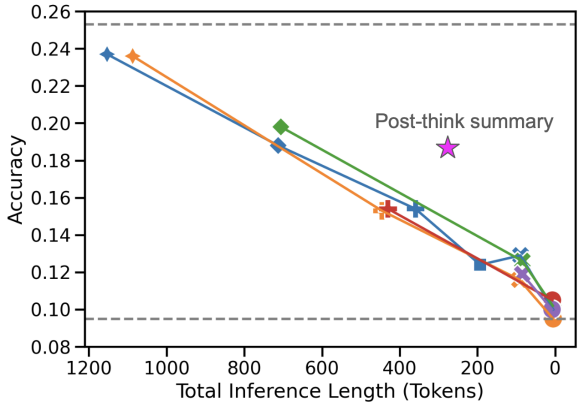


Figure 7: **Post-think outperforms iterative summarization at equal length (DeepSeek-R1, Qwen2.5-7B student).** Accuracy vs. total tokens; post-think distillation dominates at matched budgets. Dashed horizontal lines indicate full-thinking (upper) and no-thinking (lower) baselines. Iterative summarization results show different curricula, some of which omit certain intermediate stages according to a configured "step size"; each colored line shows performance after each step of a single curriculum. Results shown for both teachers in Table 2.

C Curriculum Details

C.1 Segment-Removal Hyperparameters

For segment-removal curricula (left-to-right and random removal), we set $\kappa = 105$ and $\Delta = 7$, yielding 15 removal stages plus a final no-thinking stage. We chose these values to balance training time and trace coverage: increasing both κ and Δ proportionally maintains the same number of stages (and thus the same training budget) while ensuring the curriculum covers the full reasoning trace for most examples. With these settings, the median trace (approximately 70 segments) reaches zero remaining segments by stage 10, and the majority of traces are fully covered before the final no-thinking stage.

C.2 Summarization Curriculum

For the iterative summarization curriculum, we use the target length schedule $\mathcal{L} = \langle 1500, 1000, 500, 250, 100, 50 \rangle$ tokens. At each stage, the teacher model summarizes the previous stage's output to approximately the next target length. We distribute the total training budget evenly across the six summarization stages, then train on problem→answer pairs (no thinking) until convergence. As with segment-removal curricula, we reset the optimizer state between stages.

D COCONUT Implementation Details

We attempted to replicate the COCONUT approach (Hao et al., 2024) on our competition-level mathematics dataset to evaluate whether latent reasoning could provide an efficient alternative to explicit reasoning traces. We note that our implementation of COCONUT pursues only very aggressive reduction, ending up with only 8 latent representations—and therefore is easily comparable to the no-reasoning baseline but not to most direct distillation approaches.

D.1 Adaptation to Long Reasoning Traces

The original COCONUT method progressively removes reasoning tokens during training, replacing them with continuous latent representations. However, our reasoning traces are significantly longer and more unstructured than those in GSM8K (average ∼5,000 tokens vs. ∼200 tokens). To adapt the method to our setting, we made the following

modifications:

Segment-based removal. Instead of removing individual tokens, we removed contiguous segments of text split by double newlines (\n\n), consistent with the segment definition used throughout the paper. This preserves local coherence within segments while progressively reducing the explicit reasoning trace.

Limitations of this adaptation. Our segment-level approach trades token-level granularity for tractability on long traces. This modification may not preserve properties essential to COCONUT's success on short-trace tasks; token-level removal with longer context lengths could yield different results. We view our negative result as evidence that straightforward adaptation fails, not that latent reasoning is fundamentally impossible for long traces.

D.2 Curriculum Structure

We use a 4-stage curriculum. At each stage, we replace one additional reasoning segment with 2 latent tokens, progressively compressing explicit reasoning into continuous representations. By the final stage, 8 latent tokens replace the reasoning trace entirely. This is comparable to the 6 latent tokens used in prior work on GSM8K (Hao et al., 2024), making this an aggressive compression ratio for traces that are $\sim 25\times$ longer. Scaling the number of latent tokens proportionally to trace length could yield better results.

E License Statements

This project includes components from: DeepSeek-R1, Copyright (c) 2025 DeepSeek. Licensed under MIT. QwQ-32B, Copyright (c) 2025 Alibaba Cloud. Licensed under Apache 2.0. NuminaMath 1.5, Copyright (c) 2024 Numina. Licensed under Apache 2.0. Open-R1, Copyright (c) 2025 Hugging Face. Licensed under Apache 2.0.