When Internalization Fails: Finding Better Targets for Reasoning Compression

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

Rohan Wadhawan
Abridge
rohan.wadhawan@abridge.com

Manley Roberts
Abridge
manley@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. We compare three approaches in a teacher-student distillation setup: (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 reasoning traces from DEEPSEEK-R1 and QWQ-32B, we train 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 with long, diverse reasoning traces; and (3) training on post-think (text the teacher generates after reasoning) outperforms generic summaries by 4–5 percentage points at matched token budgets. These results show that curriculum-based methods that are effective on simple tasks do not transfer to complex reasoning, while post-think provides a better distillation target because it preserves the teacher's solution path.

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. Furthermore, generating more tokens can even hurt performance through overthinking (Hassid et al., 2025). We seek to 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 (referred to throughout this paper as ICoT-SI or merely 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$200 token), structured traces. COCONUT (Hao et al., 2024) extends this approach by replacing textual reasoning tokens with a small number of learned latent tokens. These methods are attractive because they eliminate reasoning tokens at inference time. However, they have only been shown to work on simple tasks with homogeneous reasoning patterns. We test whether they scale to competition-level mathematics, where reasoning traces are long ($\sim$5,000 tokens), diverse, and exploratory. We find that ICoT-style curricula provide little benefit over direct distillation on these tasks, indicating that the success of these methods on short, structured traces (GSM8K, multiplication) does not extend to long, exploratory traces.

Specifically, we use a teacher-student distillation setup with reasoning traces from DeepSeek-R1 and QwQ-32B. We first establish a baseline by distilling on full traces then truncating at inference time after the first k tokens, which shows 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 curve established by inference-time first-k truncation (Figure 3).

We then explore direct distillation of the student model with fully shortened reasoning traces. We test several distillation target strategies: teacher-generated summaries at different lengths, first-k tokens of the original teacher trace, and post-think

[*]Work done during internship at Abridge.

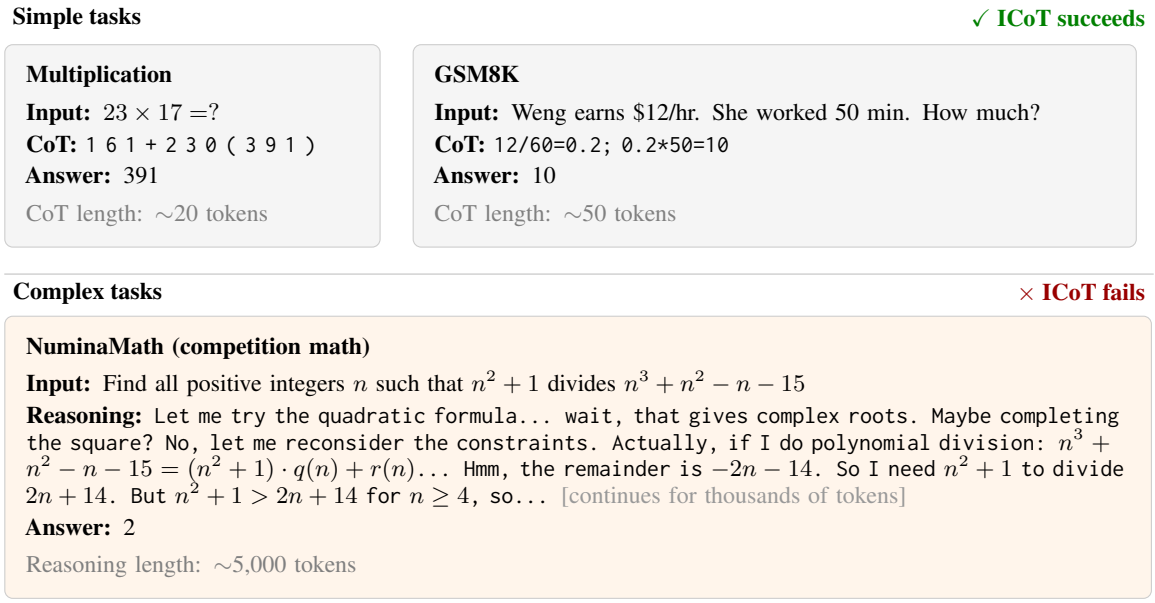


Figure 1: **The complexity gap.** ICoT progressively removes reasoning steps during training. This succeeds on short, structured traces (∼20–50 tokens). Competition math reasoning is ∼100× longer (∼5,000 tokens) and involves exploration and backtracking with no obvious discrete segments to remove.

(text the teacher generates *after* the </think> token but before the final boxed answer). Training on post-think outperforms generic summaries by 4–5 percentage points 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>). This matters because models, when forced to use less reasonining, "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, outperforming them by 4–5 percentage points at matched token budgets. We attribute this to post-think preserving the teacher's solution path.

2 Related Work

Most significant improvements in language model performance over the past several years 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 more recent work in scaling reasoning models, many methods increase inference-time token generation. In response, a growing body of work has sought to improve the token-efficiency of reasoning models.

2.1 Implicit, and latent, reasoning.

One line of work aims to *internalize* or *compress* reasoning within the language model itself, so fewer or even no reasoning tokens are decoded during inference. Stepwise internalization (ICoT-SI) trains models to generate answers directly by iteratively removing reasoning tokens, using a curriculum-learning approach (Deng et al., 2024). Similarly, COCONUT replaces reasoning tokens with a small number of continuous hidden representations that are never decoded directly (Hao et al., 2024). While both approaches show improvement over standard no-reasoning fine-tuning, they underperform full-length reasoning and are only tested on relatively simple tasks with short (∼200 token), structured reasoning traces, like multiplication and GSM8K. CODI (Shen et al., 2025) then further improves on this line of work by compressing CoT

into a continuous latent space via a single-step self-distillation and matches the performance of full-length CoT on GSM8K while using far fewer tokens. They further show their method can be applied to CommonsenseQA, a slightly more complex and realistic task.

However, these approaches have only been validated on tasks with relatively homogeneous reasoning patterns. Whether internalization scales to domains with long (∼5,000 token), heterogeneous, exploratory reasoning traces (such as competition-level mathematics) 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 for future work. In this work, we test ICoT-style curricula on competition-level math problems, adapting their curriculum approach with key adjustments for our setting of longer, more varied traces.

2.2 Length control, budgeted decoding, and early exit.

A second line of work seeks to regulate *how much* a model thinks during training or inference.

Training-time. Xiang et al. use reinforcement learning with adaptive length penalties to produce a policy that generates shorter reasoning traces while preserving answer quality (Xiang et al., 2025). 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 uses it to attach an explicit budget in the prompt (Han et al., 2025).

Inference-time. Concise-CoT prompting shows that brief, targeted reasoning often suffices on many problems (Renze and Guven, 2024). Most directly, 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 trial answer emerges, yielding 20 to 80% shorter traces with some small accuracy gains.

While both training-time and inference-time approaches can be effective at reducing reasoning length and preserving performance on complex tasks, they do not lead to the aggressive reductions in reasoning length that we are targeting. Furthermore, many of these approaches either require expensive RL training or additional inference-time components, which can render them impractical.

2.3 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; in this work we focus on distillation with *shorter* reasoning traces to yield better efficiency for the student model.

3 Methods

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

3.1 Task & Dataset

We use NUMINAMATH 1.5, a dataset of competition-level mathematics problems drawn 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**, we use a subsample of NuminaMath 1.5 included in the OPENR1-MATH dataset, which includes 93k problems paired with full reasoning traces and correct solutions generated by DeepSeek-R1. For **QwQ-32B**, we use pregenerated 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 across teachers for direct comparison. For each teacher model, we obtain responses of the following form:

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

We use the `boxed` formatting standard to indicate the answer to the problem, making it easier to systematically extract.

Post-think. We observe that all generations from the teacher models 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. The teacher has already solved the problem and is now explaining its solution. 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 reasoning traces. Therefore, we 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 for distillation. We initially experimented with other student models but found them ineffective for our setting. We conduct supervised fine-tuning with LoRA applied to all layers of the model. We do early-stopping based on the validation loss.

In particular, we tried base variants of Gemma 3 4B, Llama 3/3.1 8B, and OLMo 2 7B as student models. We did not use them in the main experiments due to practical issues: Llama variants achieved extremely low accuracy (below 5%) even when trained with full thinking traces; we did not investigate the cause. OLMo 2 7B is constrained by a 4,096-token context length which is too short for our task, and Gemma 3 4B training was prohibitively slow and expensive under our available compute.

3.2.1 Approaches Overview

We compare three approaches for controlling or shortening reasoning length.

First-k truncation (inference baseline). At test time we append </think> after k reasoning tokens for $k \in \{50, 100, 250, 500, 1000, 1500\}$. This sweep over the value of k yields a baseline accuracy–generation length curve.

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 there are none left. While this approach is feasible in the simpler tasks explored in that work ($N \times N$ 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. Instead, we test four alternative curriculums better suited for our task and more generalizable to other real-world settings:

- **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 removed segments with increasingly shortened teacher-generated summaries (see Appendix A for more information on the distribution of lengths of these summaries).

Direct distillation to shortened traces. As a non-curriculum alternative, we use direct distillation: first train on full reasoning traces, then continue training on a single shortened target (rather than progressively shortening through multiple stages). We test the following shortened 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 A for token distributions.
- Official solution explanations from the original NuminaMath 1.5 dataset.
- First-k tokens of the reasoning traces.
- The post-think section from the teacher models.

3.2.2 Segment-Removal Curricula

Intuitively, 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 whatever segments remain.

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, $S = \lceil \kappa/\Delta \rceil$
$N^{(t)}$	Steps per stage, $N^{(t)} = \lfloor B/S \rfloor$

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

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 model trained on full reasoning traces (without post-think), ensuring a fair comparison across methods. We summarize the key curriculum hyperparameters in Table 1. We fix a total curriculum training budget of B training 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 continue training on problem → answer only until convergence. We choose κ and Δ such that the curriculum covers the full reasoning trace for most examples while maintaining a fixed number of stages (and thus fixed training time). Let $S = \lceil \kappa/\Delta \rceil$ be the number of removal stages. We split B evenly so each stage uses

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

3.2.3 Iterative Summarization Curricula

We shorten teacher traces by replacing them with successively shorter, teacher-written summaries that preserve important 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 teacher model.

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 → answer pairs.

3.2.4 Additional Training Details

Following Deng et al. (2024), we implement *removal smoothing*: at each stage, with probability 0.05 we randomly remove up to 5 additional segments to prevent overfitting to exact stage boundaries. We reset the optimizer state between stages.

3.3 Evaluation

To evaluate model performance, we do generation with a temperature of 0.3 and maximum length of 10000 tokens. We train the student model to output its final answers using \boxed{} formatting and use HuggingFace's `math-verify` tool to evaluate its accuracy against ground-truth solutions.

4 Results

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

4.1 Inference-time truncation misleads on efficiency

We first examine what happens when post-think text is included in training. With standard SFT that includes post-think, truncating reasoning after the first k tokens at inference time appears to preserve accuracy. However, models compensate by generating longer text after the </think> token, so shorter

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	6	-	–
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 A for other levels). Post-think achieves the best accuracy-efficiency trade-off. See Appendix C for COCONUT details.

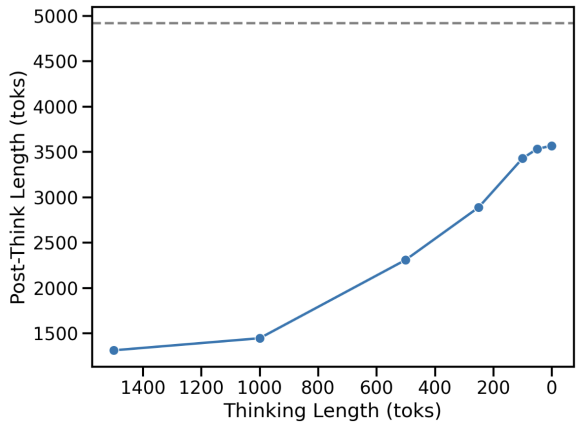


Figure 2: **Models compensate for truncated reasoning.** When we truncate reasoning at inference time, models generate longer post-think text. As reasoning length decreases, post-think length increases, keeping total token count high. This compensation undermines the apparent token savings from shorter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

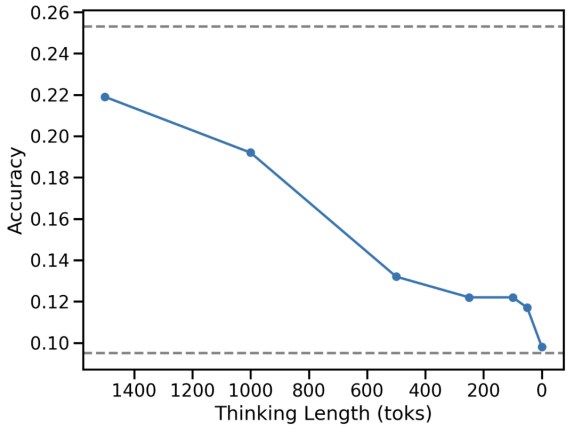


Figure 3: **Accuracy decreases as reasoning shortens.** When we train without post-think and vary reasoning length with first-k truncation, accuracy decreases monotonically with a sharp drop when reasoning is removed entirely. Dashed horizontal lines indicate full-thinking distillation (upper) and no-thinking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

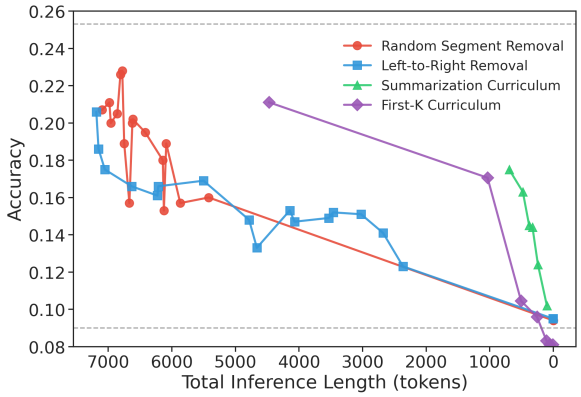
reasoning does not reduce total tokens (Figure 2). Post-think grows as thinking shrinks, undermining token savings. This pattern holds for both R1 and QwQ teacher traces (Table 2).

4.2 Removing post-think from training reveals the length-accuracy tradeoff

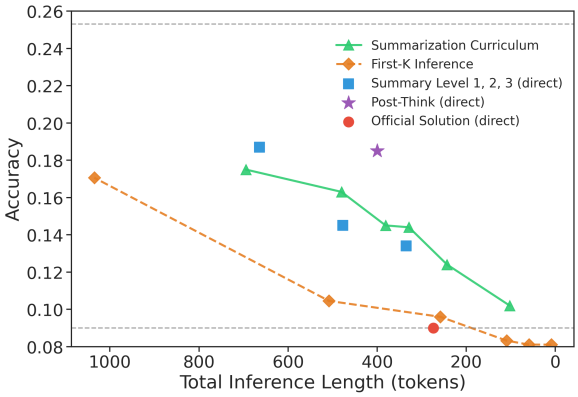
Given this compensation effect, we exclude post-think from all subsequent experiments. This yields a clear baseline: accuracy decreases as reasoning length decreases, with a sharp drop when reasoning is removed entirely (Figure 3 and Table 2). This establishes the baseline trade-off for comparing training-based shortening methods.

4.3 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—several perform no better than the no-thinking baseline (Figure 4a and Table 2). Varying the schedule or removal rate yields similar results, with no consistent benefit.



(a) **ICoT-style curricula fail to internalize reasoning**, 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 summary outperforms other distillation targets.** 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.

COCONUT (Hao et al., 2024), which replaces reasoning tokens with latent representations, provides further evidence. Adapting it to our longer traces by removing segments rather than tokens, we find it achieves only 4.2% accuracy (compared to 29.2% for full traces and 8.1% for no reasoning; see Table 2 and Appendix C). Combined with our curriculum results, this indicates that internalization methods effective on short, structured traces (GSM8K, multiplication) do not extend to long, exploratory traces characteristic of competition math. Results are consistent across both teachers where available.

4.4 Post-think outperforms generic summaries at matched budgets

We train students to generate the teacher's post-think. At matched token budgets, post-think achieves higher accuracy than generic summaries (Figure 4b and Table 2), representing an improvement of 4–5 percentage points. This holds across both teachers, showing that post-think transfers better than generic summaries.

Training on official solution explanations from NuminaMath 1.5 (human-written, answer-directed explanations independent of the teacher's reasoning) achieves only 9% accuracy. This is similar to the no-thinking baseline and much worse than post-think (18.5%), despite official solutions also being answer-directed. This gap suggests that answer-directedness alone does not explain post-think's effectiveness; we return to this in Discussion.

4.5 What makes a good summary?

Summaries of similar length can differ substantially in accuracy. The contrast between post-think (18.5%) and official solutions (9%) shows that answer-directedness alone is insufficient. Properties beyond length and answer-directedness determine effectiveness; we analyze this in Discussion.

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 C), 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 has long, diverse reasoning traces (∼5,000 tokens vs. ∼200 for GSM8K) with high variance in structure, making it hard to identify which segments to remove; (2) internalization may be difficult in this setting because solutions often require exploration and backtracking that cannot be compressed into a short sequence or internalized into the model's hidden states.

5.2 Why post-think outperforms generic summaries

Post-think outperforms teacher-generated summaries at matched token budgets as a distillation target (Figure 4b, Table 2). We hypothesize that this difference stems from differences in the contextual roles of post-think and a standard post-hoc summary; post-think is generated as a natural continuation of the teacher's reasoning process and scales up as the reasoning trace is limited–indicating that it is in some sense a key final step to producing the final boxed answer.

Summaries, by contrast, require artificially compressing traces which may disrupt the reasoning structure by reordering, merging, or omitting intermediate conclusions. Prior work on step-by-step distillation supports the intuition that preserving this structure distills reasoning more effectively than alternatives (Hsieh et al., 2023; Shridhar et al., 2023).

Evidence from official solutions. The poor performance of official solutions (9%) provides the strongest support for this hypothesis. Official solutions are also answer-directed, written with knowledge of the answer, yet they perform comparably to the no-thinking baseline (8%). 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 from the teacher's outputs. Post-think succeeds because it recapitulates the teacher's own successful reasoning; official solutions fail because they introduce different computational patterns.

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 outperforms generic summaries by 4–5 percentage points at matched budgets.

Future Work

First, analyzing what makes post-think effective could inform better summary strategies. Second, testing these methods on other domains (code, scientific reasoning, commonsense) would test generalization. Third, 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).

Limitations

We focus on competition math; generalization to other domains (code, commonsense reasoning) remains to be tested. 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; whether larger students can internalize long traces where smaller ones cannot is unclear. We report single runs; replication across seeds would strengthen statistical conclusions. Our segment-level COCONUT adaptation trades token-level granularity for tractability; alternative adaptations may yield different results. Our hypothesis about why post-think outperforms summaries is supported by indirect evidence (the official solutions comparison), but controlled experiments could isolate the specific properties that drive this advantage.

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.

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*.

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.

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

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

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.

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.

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

A 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.

The summary levels represent progressively shorter compressions of the original reasoning traces. As shown in the histograms, there is considerable variance in summary lengths within each level, reflecting the diversity of problem complexity in the dataset. The vertical dashed lines indicate the median token counts for each level, demonstrating that the summarization process successfully achieves progressively shorter targets while maintaining some flexibility based on problem difficulty.

For our main results (Table 2), we report levels 1–3 as they are most comparable in length to posthink, which has median token counts of 511 (R1) and 541 (QwQ).

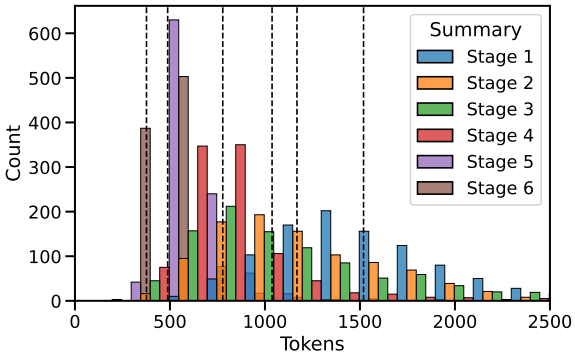


Figure 5: **Token distributions for DeepSeek-R1 summary levels.** Each level represents progressively shorter summaries, with vertical dashed lines indicating median token counts.

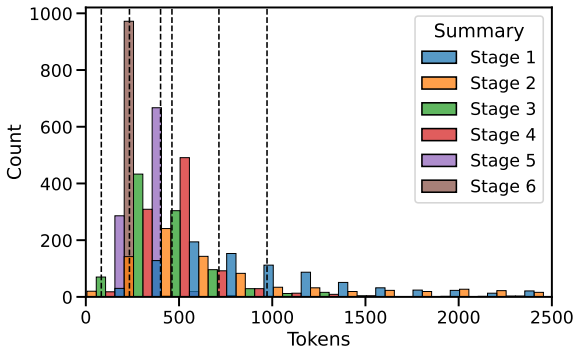


Figure 6: **Token distributions for QwQ-32B summary levels.** Each level represents progressively shorter summaries, with vertical dashed lines indicating median token counts.

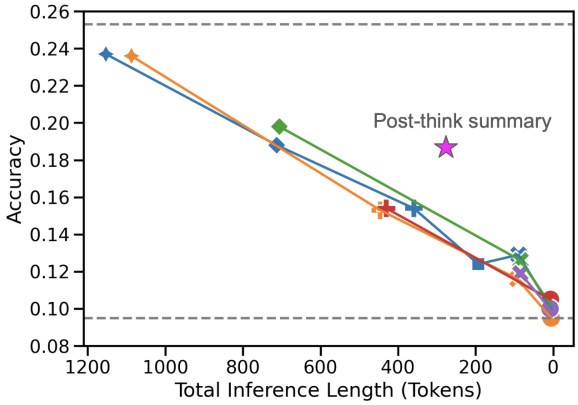


Figure 7: **Post-think outperforms generic summaries at equal length.** Accuracy vs. total tokens; post-think distillation dominates at matched budgets. Dashed horizontal lines indicate full-thinking (upper) and no-thinking (lower) baselines. Results shown for both teachers in Table 2.

B Curriculum Details

B.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.

B.2 Summarization Curriculum

C 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.

C.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 newline characters (\n). 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.

C.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, 6 latent tokens replace the reasoning trace entirely. This matches the latent token count used in prior work on GSM8K (Hao et al., 2024).