Think Faster Than Words: Efficient LLM Chain-of-Thought Reasoning via Dynamic Shortcut Decoding

Fan Liu[1,*], Yanhao Wang[1,†], Min Zhang[1,†], Zhikang Chen[2], Zeyuan Li[3,*], Lewei He[3,†], Jiahui Pan[3,†]

[1]East China Normal University, [2]The University of Oxford, [3]South China Normal University

9527liufan@gmail.com, yhwang@dase.ecnu.edu.cn, mzhang@cs.ecnu.edu.cn, chenzk202212@126.com

{2023024326, helewei, panjiahui}@m.scnu.edu.cn

Abstract

This paper proposes shortcut decoding, an efficient framework for accelerating Chain-of-Thought (CoT) reasoning in Large Language Models (LLMs). Existing methods that prune or employ early stopping to reduce latency often compromise reasoning reliability. Motivated by the observation that LLMs frequently converge to correct solutions internally before completing explicit textual reasoning, we propose a dual-signal adaptive controller that integrates lightweight probes over internal hidden states with step-level entropy. This controller detects convergence of reasoning during generation and adaptively selects between a fast-exit path and a stability-verified path to remove redundant steps while preserving answer correctness. Experiments across multiple mathematical reasoning benchmarks demonstrate that shortcut decoding reduces token usage by approximately 35%, maintains accuracy comparable to full CoT decoding, and achieves final-answer accuracy comparable to the full CoT baseline, outperforming existing early-stopping methods without updating the base model. Our code is available at https://github.com/kuromi9527/shortcut_decoding.

1 Introduction

Chain-of-Thought (CoT) prompting and its structured variants have fundamentally enhanced the reasoning capabilities of Large Language Models (LLMs) by explicitly decomposing complex problems (Wei et al., 2022; Wang et al., 2023; Kojima et al., 2022; Yao et al., 2023a; Zhou et al., 2023). While recent Large Reasoning Models (LRMs), such as OpenAI-o1 and DeepSeek-R1, scale test-time compute to achieve competition-level performance (OpenAI, 2024; DeepSeek-AI, 2025), this paradigm incurs substantial computational costs. Empirical evidence suggests that LRMs frequently *overthink*: they continue to generate redundant checks or digressions long after the core reasoning is internally resolved, sometimes even drifting from a correct solution (Wei et al., 2025; Yang et al., 2026; Li et al., 2024; Chen et al., 2025; Sui et al., 2025). A full CoT rollout typically enumerates all reasoning steps from (1) to (n) before emitting the final answer, even when a substantial portion of the later steps is logically redundant. While system-level optimizations can alleviate latency (Kwon et al., 2023; Dao et al., 2022; Leviathan et al., 2023), they do not address this algorithmic redundancy.

In this paper, we address the critical challenge of automatically truncating redundant segments while preserving accuracy. Prior efforts to mitigate this efficiency bottleneck have primarily relied on system-level optimizations (Dao et al., 2022; Leviathan et al., 2023) or on model compression techniques such as knowledge distillation (Yu et al., 2024; Li et al., 2026). However, these approaches often require expensive retraining or fail to dynamically adapt to the varying difficulty of individual queries. While heuristic-based early stopping (e.g., checking output entropy) offers a training-free alternative (Mao et al., 2025; Laaouach, 2025), it frequently suffers from the confident error problem, where models maintain low uncertainty even while hallucinating.

To overcome these limitations, we distinguish our approach by grounding it not on speculative heuristics but on the *empirical phenomenon* supported by recent probing studies (Zhang et al., 2025; Turpin et al., 2023): LLMs exhibit a *thinking-faster-than-they-speak* behavior. Specifically, the high-dimensional internal representation often converges to the correct answer well before the textual generation concludes. Motivated by this misalignment between internal belief saturation and

[*]Both authors contributed equally to this work.

[†]Corresponding authors.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 28825–28836

external realization, we propose the **shortcut decoding** framework. Prior research indicates that explicit CoT text does not always faithfully reflect the model's latent state (Lyu et al., 2023; Lightman et al., 2024; Turpin et al., 2023; Manakul et al., 2023). While external signals such as semantic entropy quantify confusion, they often fail to detect "confidently wrong" errors (Liu et al., 2025; Farquhar et al., 2024; Kossen et al., 2024; Mao et al., 2025; Laaouach, 2025; Li et al., 2026). Conversely, internal probes of hidden states can effectively predict the correctness of reasoning well before generation concludes (Fu et al., 2025; Shen et al., 2025; Xu et al., 2025).

According to these insights, we define a task-specific reasoning convergence point as an intermediate step at which the model's hidden states already capture the correct solution, even though the model may subsequently continue generating additional CoT text. Our goal is to detect such a step $S_{i^\star}$ during decoding and then switch directly to final answer generation, thereby skipping the remaining reasoning steps. To achieve this, we propose a dual-signal early-exit framework built atop a frozen reasoning model without updating the base model parameters. After each reasoning step S_i, the controller collects two complementary signals. The first is an internal confidence score $S_{\text{probe}}(S_i)$, predicted from the step representation by a lightweight MLP probe g_ϕ. The second is an external uncertainty score $\bar{H}_i$, computed as the step-averaged output entropy. These two signals are jointly used to determine whether to terminate the reasoning process. Specifically, a fast exit is triggered when either $S_{\text{probe}}(S_i)$ is very high or $H_{\text{avg}}(S_i)$ reaches an extremely low level, indicating strong convergence. When $S_{\text{probe}}(S_i)$ is high but $\bar{H}_i$ remains moderate, a stable exit strategy is applied, which requires the signals to remain consistent over multiple consecutive steps before exiting. Our main contributions are summarized as follows:

- **Empirical Validation of "Thinking-Faster-than-Speaking" Hypothesis:** We provide empirical evidence that the internal hidden states of LLMs offer predictive signals for final correctness significantly earlier than explicit CoT completion, laying a foundation for early stopping based on internal states.
- **Proposal of the Entropy-Probe Dual-Signal Framework:** We propose a novel framework that unifies internal semantic consistency and external uncertainty. This dual-signal approach enables robust and adaptive early stopping, addressing the limitations of relying solely on entropy or internal probes.
- **Design of the Adaptive Controller:** We design an adaptive controller that dynamically evaluates internal probe scores and step-level entropy during reasoning. It takes different actions based on the reasoning state (e.g., rapid convergence, gradual convergence, or confusion), achieving efficient inference under strict accuracy constraints.
- **Achieving Significant Efficiency-Accuracy Trade-Off:** Extensive experiments on mathematical reasoning benchmarks show that our method substantially reduces token usage while maintaining or improving accuracy, outperforming state-of-the-art dynamic early-exit baselines (Wei et al., 2025; Yang et al., 2026; Li et al., 2024). This effectively mitigates semantic drift and redundancy in long CoT reasoning.

2 Related Work

To address the high computational cost and overthinking behavior of LLMs on complex reasoning tasks, prior work has mainly followed two directions: (i) explicit self-correction and structured CoT methods that improve robustness at the cost of longer reasoning and (ii) dynamic, training-free early stopping that intervenes during inference without changing base model parameters.

2.1 Self-Correction and Structured CoT

A primary line of research aims to enhance the robustness of reasoning by structuring the generation process. Agentic and reflective paradigms, such as ReAct and Reflexion, organize reasoning into iterative "act-and-reflect" cycles to revise answers based on feedback (Yao et al., 2023b; Shinn et al., 2023). Similarly, structured prompting schemes, including Self-Consistency, Tree-of-Thoughts, Least-to-Most, and stepwise verification, explicitly decompose problems or explore diverse reasoning paths to ensure reliability (Wang et al., 2023; Yao et al., 2023a; Zhou et al., 2023; Kojima et al., 2022; Lyu et al., 2023; Lightman et al., 2024).

However, these reasoning chains face two critical issues: *faithfulness* and *efficiency*. From a

faithfulness perspective, models often generate explanations decoupled from their internal states, and self-correction is not guaranteed to rectify logical errors (Lyu et al., 2023; Turpin et al., 2023; Manakul et al., 2023). From an efficiency perspective, empirical studies show that accuracy gains often saturate while models overthink or spiral into redundancy (Wei et al., 2025; Yang et al., 2026; Li et al., 2024). Recent latent-space analyses suggest a solution: essential reasoning logic is often compressed in hidden states well before the textual CoT concludes, as demonstrated by probe-based and continuous CoT studies (Fu et al., 2025; Shen et al., 2025; Xu et al., 2025).

2.2 Dynamic Inference-Time Intervention and Early Stopping

A second direction focuses on dynamic intervention: monitoring the evolving CoT and terminating generation once convergence is detected (Wei et al., 2025; Yang et al., 2026; Li et al., 2024). These methods typically rely on different types of signals.

The first type utilizes surface-level patterns in the generated text. Approaches like *Dynamic Early Exit* (Yang et al., 2026) and *Escape Sky-high Cost* (Li et al., 2024) monitor stability or specific markers (e.g., "Wait") to truncate redundant explanations. While being training-free, these rely on handcrafted heuristics that may not generalize.

The second type leverages uncertainty and entropy. Step-level entropy methods aggregate next-token distributions to gauge confidence, triggering early exits when entropy remains low (Mao et al., 2025; Laaouach, 2025; Li et al., 2026). Broader research on semantic entropy further quantifies the risk of hallucinations (Liu et al., 2025; Farquhar et al., 2024; Kossen et al., 2024). However, a key limitation is the "confidently wrong" phenomenon: low entropy does not guarantee objective correctness (Turpin et al., 2023). Although early-exit mechanisms have been successfully applied in encoder-style or adaptive architectures (Xin et al., 2020; Schuster et al., 2022), adapting them to long-form CoT requires more robust signals.

A third type exploits hidden-state or answer-level signals: Eisenstadt et al. (2025) steer the model along a progress vector learned from hidden states; and Liu and Wang (2025) detect answer stability across truncation points. Parallel RL-based methods control reasoning lengths via training-time budget policies (Aggarwal and Welleck, 2025; Xu et al., 2026).

Our framework addresses the above limitations by combining external uncertainty with internal hidden-state probes (Fu et al., 2025), creating a unified controller that distinguishes true convergence from confident errors. Unlike progress-vector intervention or answer-level detection, our controller is an external observer at step boundaries that leaves the model's computation untouched and couples a step-level probe with step-level entropy to reject "confidently wrong" trajectories; unlike RL-based budget control, it is training-free and complementary, trimming residual "continued verification" at inference time.

3 Methodology

This section proposes the shortcut decoding framework, a plug-and-play method that accelerates LLM reasoning by pruning redundant Chain-of-Thought (CoT) steps without requiring fine-tuning of model parameters. An overall architecture of our framework is shown in Figure 1. The framework constructs a three-stage architecture atop the frozen base LLM. The first stage is *logical step segmentation*, where the LLM generates the CoT in an autoregressive manner, while the framework dynamically segments the continuous token stream into distinct logical reasoning steps. The second stage, *signal extraction*, operates at the boundary of each logical step to capture multi-dimensional indicators, including the internal cognitive state inferred from hidden-state probes and the external uncertainty quantified by output entropy. The third stage is *adaptive decision*, in which the controller evaluates the fused signals from the second stage to identify the Reasoning Completion Point (RCP) and determines whether to immediately terminate the generation process.

3.1 Problem Formulation

We consider a large reasoning language model f_θ that takes a natural-language question q as input and produces a chain-of-thought, followed by a final answer, as output. The generated token sequence can be written as

$$Y = (y_1, y_2, \ldots, y_N), \tag{1}$$

where the prefix corresponds to the chain-of-thought (CoT) and the tail corresponds to the final answer. The model defines a conditional distribution over output sequences and generates them in

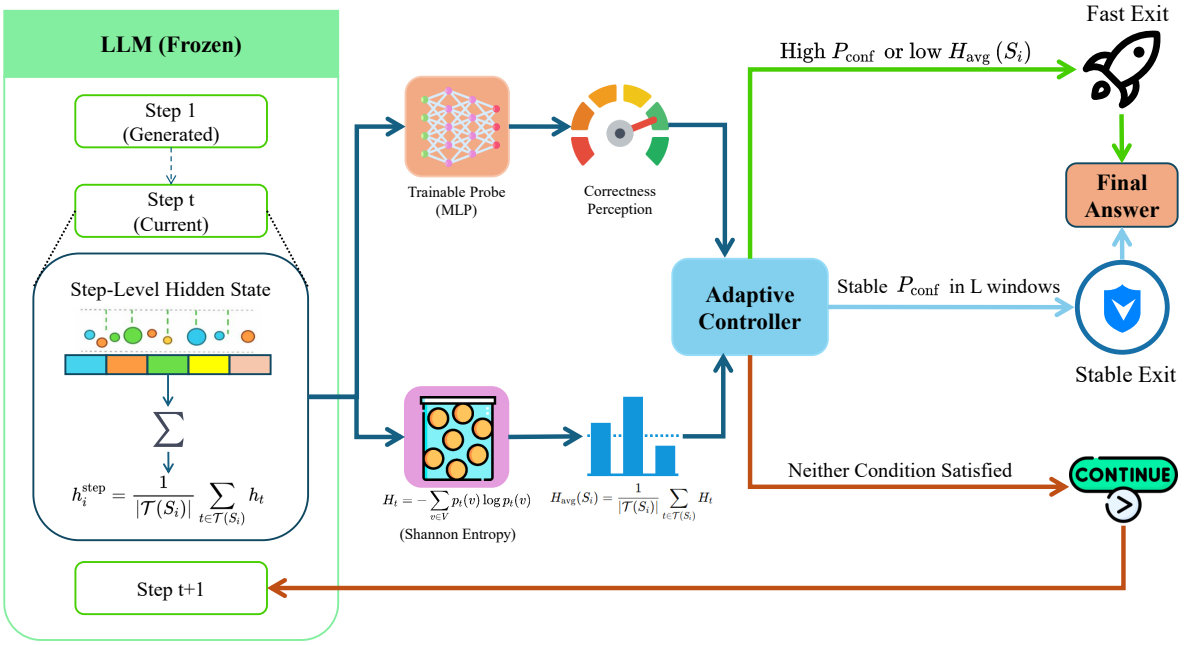

Figure 1: Architecture of the shortcut decoding framework. At each reasoning step, the adaptive controller takes the step-average entropy and the probe score as input and chooses among fast early stop, stable early stop, or continuing CoT generation based on different conditions.

an autoregressive fashion, i.e.,

$$p_\theta(Y \mid q) = \prod_{t=1}^{N} p_\theta(y_t \mid y_{<t}, q). \tag{2}$$

During generation, the model maintains a sequence of hidden states $\{h_t\}_{t=1}^{N}$, where $h_t \in \mathbb{R}^d$ denotes the final-layer hidden representation at position t. At each position, a linear projection followed by a softmax maps h_t to a probability distribution over the vocabulary V, i.e.,

$$p_t(v) = p_\theta(y_t = v \mid y_{<t}, q), \; v \in V, \tag{3}$$

which is used both to sample the next token and to construct uncertainty-based signals later.

To make early-stopping decisions at a more meaningful granularity than single tokens, we segment the CoT prefix into a sequence of *logical steps* as follows:

$$C = (S_1, S_2, \ldots, S_L), \tag{4}$$

where each S_i is a contiguous subsequence of tokens corresponding to one interpretable reasoning step. To facilitate online boundary detection, our controller operates on *natural paragraph delimiters*: a new reasoning step is identified whenever the output stream produces a double newline ("\n\n") or an end-of-thinking tag ("</think>").

We adopt a unified prompting template that additionally encourages the model to write CoT in a numbered format such as "(1) . . . (2) . . . (3) . . . ", but this numbered template only serves as an optional soft guide for clearer structure and is not a prerequisite of the step detector. Reasoning models such as DeepSeek-R1 and Qwen3 inherently organize their chain-of-thought into paragraph-separated steps, regardless of whether a numbered format is requested. Thus, the "\n\n" detector reliably identifies step boundaries under any prompt style. Only at detected step boundaries do we compute step-level signals and consider early-stopping decisions.

3.2 Entropy-Based External Signal and Training-Free Early Stopping

We use output-distribution entropy as an external uncertainty signal. At decoding position t, the model yields a next-token distribution p_t over the vocabulary V, and the token entropy is

$$H_t = -\sum_{v \in V} p_t(v) \log p_t(v). \tag{5}$$

To reduce token-level noise, we aggregate entropy at the reasoning-step level. Let $\mathcal{T}(S_i)$ be the set of token positions in step S_i. The step-average entropy is

$$H_{\text{avg}}(S_i) = \frac{1}{|\mathcal{T}(S_i)|} \sum_{t \in \mathcal{T}(S_i)} H_t. \tag{6}$$

Entropy-only Baseline During online decoding, at each step boundary, we maintain (i) current CoT token usage under a per-problem thinking budget and (ii) a window of the most recent step entropy values. We trigger early stopping at step S_i if (a) the remaining budget is sufficient for final answer generation and (b) all step-average entropies in the recent window are below a fixed threshold. Once triggered, we terminate CoT, treat the current history as the reasoning context, and switch to an answer-only prompt to produce the final answer.

The baseline is training-free and model-agnostic. However, entropy mostly reflects confidence rather than correctness: the model may follow an incorrect premise with consistently low entropy, leading to premature exits on "confident-but-wrong" trajectories. This motivates adding an internal signal that more closely correlates with the correctness and completeness of reasoning.

3.3 Adaptive Early Stopping with Entropy and Probe Signals

As shown in Figure 1, our framework keeps the base model frozen and adds an adaptive controller that decides whether to stop or continue after each reasoning step S_i. The controller takes two step-level signals: an internal probe score and an external entropy score, as input.

For each step S_i, we first construct a step-level hidden representation by averaging the final-layer hidden states within the step, i.e.,

$$h_i^{\text{step}} = \frac{1}{|\mathcal{T}(S_i)|} \sum_{t \in \mathcal{T}(S_i)} h_t, \tag{7}$$

which summarizes the internal reasoning state after completing step S_i. On top of h_i^{step}, we attach a lightweight probe g_ϕ implemented as a small multilayer perceptron, i.e.,

$$s(S_i) = g_\phi\big(h_i^{\text{step}}\big), \tag{8}$$

and obtain a step-level probe score through a sigmoid, i.e.,

$$S_{\text{probe}}(S_i) = \sigma\big(s(S_i)\big), \tag{9}$$

which estimates whether the current reasoning state is sufficient for safely producing the final answer. The probe is trained with step-level labels while the base model f_θ remains frozen. In parallel, the entropy branch provides the external uncertainty signal $H_{\text{avg}}(S_i)$ defined in Section 3.2.

At inference time, the adaptive controller obtains the pair $\big(H_{\text{avg}}(S_i), S_{\text{probe}}(S_i)\big)$ at each step boundary and chooses among three actions: *fast exit*, *stable exit*, and *continue*, matching the three outgoing paths in Figure 1. Specifically, *fast exit* targets highly confident steps and triggers immediate termination whenever either condition holds

$$H_{\text{avg}}(S_i) \le \theta_{\text{entropy\_extreme}} \tag{10}$$

or

$$S_{\text{probe}}(S_i) \ge \theta_{\text{aha}}. \tag{11}$$

When *fast exit* is triggered, we stop CoT generation at step S_i and directly generate the final answer from the current context. *Stable exit* handles gradual convergence, where $S_{\text{probe}}(S_i)$ is high but $H_{\text{avg}}(S_i)$ is not yet at an extreme low level. To avoid reacting to transient fluctuations, the controller requires step-level consistency over a window of length L and triggers *stable exit* only when the probe scores remain above θ_{stable} and the recent entropies stay in a medium-to-low range for consecutive steps. If neither of the early stopping conditions is met, the controller continues CoT generation by prompting the LLM to perform the next reasoning step. In this case, the signals are recomputed, and the decision is repeated.

4 Experiments

4.1 Experimental Setup

Datasets and Models To assess model performance across reasoning tasks of varying difficulty levels, we selected three representative benchmarks: MATH-500 (covering algebra, geometry, and other competition-level domains), GSM8K (grade school math word problems), and AIME 2024/2025 (high-difficulty competition problems). We employed DeepSeek-R1-7B, DeepSeek-R1-14B, and Qwen3-8B for comparison. These models exhibit strong CoT generation capabilities, enabling us to verify each method's generalizability across model scales and architectures.

Baselines We compare our proposed framework against several baselines: (1) Standard CoT (Baseline); (2) No-thinking (direct answer generation), serving as a performance lower bound; and (3) state-of-the-art dynamic early-stopping methods, DEER (Yang et al., 2026) and Dynasor (Fu et al., 2025). Additionally, we examine ablation variants, Pure Entropy and Pure Probe, to validate the necessity of the dual-signal design in our framework.

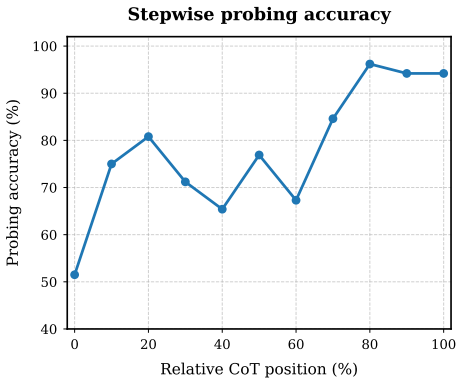

Figure 2: Stepwise probing accuracy across relative CoT positions. The probe accuracy increases rapidly and saturates at around 80% of the CoT, well before the final answer is written out.

Evaluation Metrics The evaluation metrics primarily include final-answer accuracy (Acc), average token consumption (Tokens), and compression ratio (CR). The compression ratio quantifies efficiency gains and is defined as

$$\mathrm{CR} = \frac{\sum_{i=1}^{N} T_{\text{early}}^{(i)}}{\sum_{i=1}^{N} T_{\text{full}}^{(i)}}, \tag{12}$$

where $T_{\text{early}}^{(i)}$ and $T_{\text{full}}^{(i)}$ represent the token counts for the early-stopped and full reasoning paths, respectively. A lower CR value indicates greater computational savings.

Inference Implementation For evaluation, we run early-exit decoding and full-CoT decoding separately, using the same prompts and decoding settings. When simulating an early exit, we reuse the KV cache at the selected step boundary to start answer-only decoding from the same prefix, ensuring a fair comparison without information leakage. All token statistics reported are computed only on the early-exit path.

4.2 Does the Model "Think Faster Than It Speaks"?

We first verify the core hypothesis of this work: the internal hidden states of large reasoning models contain forward-looking signals that can predict the correctness of the final answer before the explicit CoT text is fully generated. If this holds, then success or failure can be judged from intermediate internal representations, providing a solid basis for shortcut decoding. We attach a lightweight MLP probe to the frozen base model. At the end of each logical step in the CoT, the probe reads the corresponding step-level hidden representation and is trained, under binary supervision, to predict whether the current reasoning path will eventually lead to a correct final answer. The probe output is mapped through a sigmoid to obtain an estimated probability that "continuing from this step will yield a correct answer."

Figure 2 reports the probe accuracy as a function of relative CoT progress. Even within the first ≈10% of the reasoning trajectory, the probe already reaches around 70–80% accuracy, well above chance, indicating that meaningful correctness signals emerge very early. More importantly, the accuracy rises further, reaching the mid-90% range in the late stage (roughly 75–85% of the CoT), after which it remains essentially stable. This suggests that before the model spends the last ≈15–25% of its computation on generating detailed derivations and formatted explanations, its internal representations have largely converged on the correct conclusion. These observations provide empirical evidence for the claim that the model thinks faster than it speaks, and motivate using internal states to trigger early stopping around the reasoning completion point.

4.3 Analysis on the Limitations of Pure Entropy Early-Stopping Baseline

We analyze the pure entropy-based early-stopping baseline, showing that external uncertainty alone is insufficient to achieve good performance. At each logical step, we compute the average token-level Shannon entropy and apply a fixed threshold to decide whether to terminate CoT generation. Figure 3 presents a representative case on the MATH-500 dataset. Although the model reaches the correct answer at Step 3, it continues redundant verification. In such cases, the step entropy oscillates: it spikes during linguistic hesitation (e.g., "Wait ..." or "... Wait") and drops during deterministic re-checks, making fixed thresholds unreliable for distinguishing true completion from redundant continuation. Quantitatively, this baseline reduces CoT tokens by 43% but lowers accuracy from 90.8% to 88.1%. This highlights a fundamental mismatch: step entropy reflects local next-token uncertainty rather than global reasoning completion, motivating the need for internal probe signals.

4.4 Comparison with Baselines

We evaluate our proposed method on four benchmark datasets and compare it against both stan-

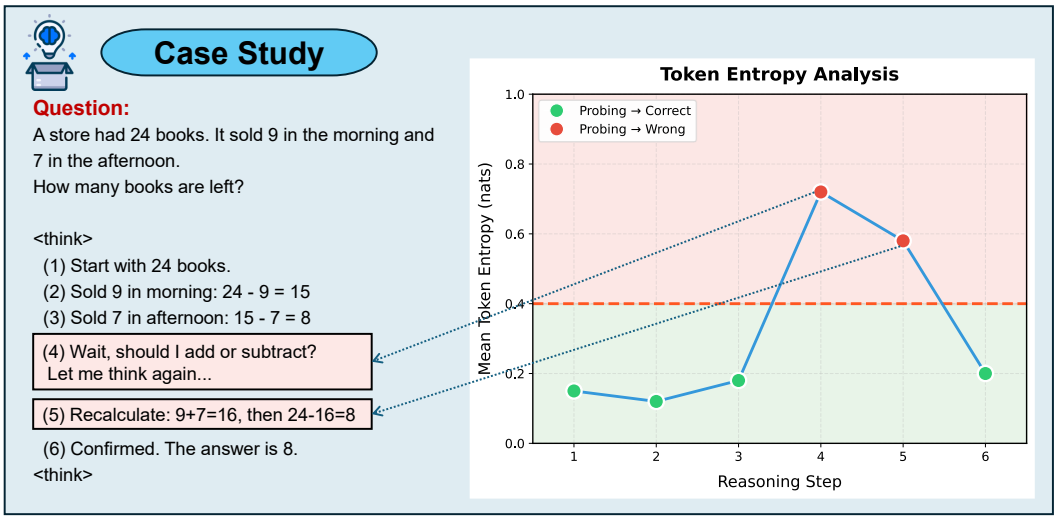

Figure 3: Case analysis on the overthinking phenomenon. The model has obtained the correct answer at Step 3 but continues redundant reasoning. Stepwise entropy is high during hesitation phases and low during confident, deterministic steps, revealing the limitation of simple entropy-based stopping criteria.

Method	GSM8K			MATH-500			AIME 24			AIME 25		
	Acc↑	Tokens↓	CR↓	Acc↑	Tokens↓	CR↓	Acc↑	Tokens↓	CR↓	Acc↑	Tokens↓	CR↓
DeepSeek-R1-Distill-Qwen-7B												
Baseline	89.5%	1512.3	100.0%	90.8%	3661.9	100.0%	43.3%	10611.3	100.0%	36.7%	10309.0	100.0%
No-thinking	86.9%	**302.1**	**20.0%**	80.1%	**585.7**	**16.0%**	10.0%	**2289.5**	**21.6%**	10.0%	**1755.3**	**17.0%**
Dynasor	89.5%	1293.3	85.5%	87.7%	2896.3	79.1%	43.3%	10231.6	96.4%	36.7%	9125.4	88.5%
DEER	89.8%	1167.9	77.2%	87.8%	2347.3	64.1%	43.3%	10611.3	100.0%	33.3%	11432.8	111.0%
Ours	**90.6%**	1036.5	68.5%	**91.2%**	2193.5	59.9%	**53.3%**	6583.5	62.0%	**40.0%**	5937.0	57.6%
DeepSeek-R1-Distill-Qwen-14B												
Baseline	**93.2%**	1519.6	100.0%	92.1%	3992.1	100.0%	51.7%	11252.1	100.0%	36.7%	12103.0	100.0%
No-thinking	91.2%	**293.9**	**19.3%**	86.4%	**1321.8**	**33.1%**	26.7%	**5639.4**	**50.1%**	13.3%	**6128.3**	**50.6%**
Dynasor	93.1%	1242.5	81.8%	91.0%	2294.4	57.5%	46.7%	8921.9	79.3%	33.3%	9834.6	81.3%
DEER	93.0%	1183.2	77.9%	91.6%	2364.3	59.2%	53.3%	8214.2	73.0%	**43.3%**	10313.6	85.2%
Ours	**93.2%**	1003.7	66.1%	**92.4%**	2102.6	52.7%	**63.3%**	8219.3	73.0%	**43.3%**	10625.7	87.8%
Qwen3-8B												
Baseline	93.8%	2194.5	100.0%	93.4%	5216.3	100.0%	63.3%	13120.3	100.0%	53.3%	11923.8	100.0%
No-thinking	89.2%	**452.3**	**20.6%**	92.5%	**1269.6**	**24.3%**	26.7%	**3189.7**	**24.3%**	16.7%	**4323.0**	**36.3%**
Dynasor	92.4%	1823.5	83.1%	91.1%	3069.3	58.8%	60.0%	11250.6	85.7%	46.7%	9281.5	77.8%
DEER	95.3%	1662.7	75.8%	89.2%	3064.9	58.8%	63.3%	10383.1	79.1%	53.3%	11284.3	94.6%
Ours	**95.8%**	1513.3	69.0%	**96.3%**	2753.8	52.8%	**66.7%**	8123.3	61.9%	**60.0%**	8039.9	67.4%

Table 1: Accuracy-cost trade-off on the GSM8K, MATH-500, AIME 24, and AIME 25 datasets. **Acc** is the final-answer accuracy, **Tokens** is the average number of CoT tokens generated (excluding the final answer), and **CR** is the compression ratio relative to the full CoT baseline (lower CR indicates fewer tokens retained and thus higher efficiency). The best result for each metric is highlighted in bold.

dard full CoT decoding and existing early-stopping baselines, DEER and Dynasor-CoT. Table 1 summarizes the performance of DeepSeek-R1-Distill-Qwen-7B, DeepSeek-R1-Distill-Qwen-14B, and Qwen3-8B on GSM8K, MATH-500, and AIME 2024/2025. Across models and datasets, the dual-signal controller achieves a favorable balance between efficiency and accuracy. Our method reduces the number of CoT tokens by roughly 31–47%, while maintaining accuracy comparable to or slightly better than full CoT decoding. This counterintuitive improvement suggests that removing redundant late-stage reasoning not only saves computation but can also reduce the risk of logical drift and harmful self-correction in very long CoT trajectories. Compared with existing inference-time early-stopping methods, the dual-signal controller offers both higher accuracy and stronger compres-

Method	GSM8K			MATH-500			AIME 24			AIME 25		
	Acc↑	Tokens↓	CR↓	Acc↑	Tokens↓	CR↓	Acc↑	Tokens↓	CR↓	Acc↑	Tokens↓	CR↓
DeepSeek-R1-Distill-Qwen-7B												
Pure Probe	90.7%	1092.3	72.2%	91.2%	2308.6	63.0%	53.3%	6732.8	63.5%	40.0%	6436.0	62.4%
Pure Entropy	88.3%	982.9	64.9%	88.1%	2089.3	57.1%	46.7%	6128.3	59.1%	36.7%	4770.0	46.3%
Ours	90.6%	1036.5	68.5%	91.2%	2193.5	59.9%	53.3%	6583.5	62.0%	40.0%	5937.0	57.6%
DeepSeek-R1-Distill-Qwen-14B												
Pure Probe	93.3%	1205.3	79.3%	92.4%	2552.7	63.9%	63.3%	7716.8	68.6%	46.7%	10935.4	90.4%
Pure Entropy	91.9%	983.5	64.7%	88.6%	2076.4	52.0%	46.7%	4993.8	44.4%	33.3%	9939.4	82.1%
Ours	93.2%	1003.7	66.1%	92.4%	2102.6	52.7%	63.3%	8219.3	73.0%	43.3%	10625.7	87.8%
Qwen3-8B												
Pure Probe	95.9%	1683.7	76.7%	95.6%	3512.8	67.3%	66.7%	11073.3	84.4%	60.0%	10935.7	91.7%
Pure Entropy	92.7%	1159.6	52.8%	92.7%	2493.1	47.8%	60.0%	7034.5	53.6%	53.3%	8064.0	67.6%
Ours	95.8%	1513.3	69.0%	96.3%	2753.8	52.8%	66.7%	8123.3	61.9%	60.0%	8039.9	67.4%

Table 2: Ablation study of different signaling mechanisms. Here, "Pure Probe" relies solely on the internal hidden-state probe, "Pure Entropy" relies solely on the external step entropy, and our dual-signal framework combines both to balance efficiency and accuracy.

sion. Under comparable accuracy levels, it consistently achieves lower compression ratios than DEER and Dynasor, indicating more aggressive yet still safe truncation of redundant segments. When contrasted with the single-signal variants, the effect of signal combination becomes clearer: relative to the pure entropy baseline, the dual-signal method recovers most of the lost accuracy, especially on medium- and high-difficulty problems where confident but incorrect reasoning is common; relative to the pure probe baseline, it achieves substantially better compression by using entropy to identify segments where the model is confident at the token level and, therefore, can afford to exit earlier. Overall, these results show that combining internal probe signals with external entropy signals yields a robust and transferable early-stopping mechanism that respects accuracy constraints while significantly mitigating overthinking and high inference costs.

4.5 Ablation Study and Error Analysis

Table 2 reports the results of the ablation study. "Pure Entropy" achieves the strongest compression but consistently degrades accuracy; on the MATH-500 dataset, it reduces accuracy by about 2.7% relative to full CoT, indicating that uncertainty alone cannot distinguish true convergence from confident errors. "Pure Probe" largely preserves accuracy but yields weaker compression, indicating conservative stopping when relying on internal signals alone. Combining both signals yields a better balance: entropy supports earlier exits in low-uncertainty regions, while the probe helps reject semantically unreliable trajectories.

We also analyze why reducing overthinking can improve accuracy. We split baseline failures into reasoning errors, in which the model never reaches the correct answer, and overthinking errors, in which a correct intermediate conclusion is later overwritten by redundant self-correction or exploration. We find that overthinking errors account for a clear majority (roughly 55–65%), while the remaining 35–45% are reasoning errors. This suggests that many failures arise after the model has already reached a correct state. By detecting the moment of reasoning convergence and truncating promptly, our controller reduces overwriting and improves both efficiency and final-answer quality.

5 Conclusion

In this paper, we address inefficiencies and overthinking in large reasoning models by proposing the shortcut decoding framework. Grounded in the hypothesis that models think faster than they speak, our framework employs a dual-signal adaptive controller that combines internal hidden-state probes with external step-level entropy to accurately detect the point at which reasoning converges. Experimental results demonstrate that our method reduces token consumption by approximately 35% while maintaining or improving accuracy, thereby reducing semantic drift due to redundant generation. These results establish our proposed framework as a robust paradigm for efficient reasoning.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 62477012 and 52308250), the AI for Science Program of the Shanghai Municipal Commission of Economy and Informatization, China (Grant No. 2025-GZL-RGZN-BTBX-01014), the Brain Science and Brain-like Intelligence Technology National Science and Technology Major Project (Grant No. 2022ZD0208900), and the Guangdong Basic and Applied Basic Research Foundation (Grant No. 2026A1515012965).

Limitations

Our framework still has several limitations: it requires training task-specific probes, which limits zero-shot generalization to new tasks without additional data, and it relies on white-box access to model internals, making it incompatible with closed-source LLMs where intermediate representations are unavailable.

References

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

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2025. Do NOT think that much for 2+3=? On the overthinking of long reasoning models. In *Proceedings of the 42nd International Conference on Machine Learning*, pages 9487–9499.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. FlashAttention: Fast and memory-efficient exact attention with IO-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359.

DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *Preprint*, arXiv:2501.12948.

Roy Eisenstadt, Itamar Zimerman, and Lior Wolf. 2025. Overclocking LLM reasoning: Monitoring and controlling thinking path lengths in LLMs. *Preprint*, arXiv:2506.07240.

Sebastian Farquhar, Jannik Kossen, Lukas Kuhn, and Yarin Gal. 2024. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630:625–630.

Yichao Fu, Junda Chen, Yonghao Zhuang, Zheyu Fu, Ion Stoica, and Hao Zhang. 2025. Reasoning without self-doubt: More efficient chain-of-thought through certainty probing. In *ICLR 2025 Workshop on Foundation Models in the Wild*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35:22199–22213.

Jannik Kossen, Jiatong Han, Muhammed Razzak, Lisa Schut, Shreshth Malik, and Yarin Gal. 2024. Semantic entropy probes: Robust and cheap hallucination detection in LLMs. *Preprint*, arXiv:2406.15927.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, pages 611–626.

Yassir Laaouach. 2025. HALT-CoT: Model-agnostic early stopping for chain-of-thought reasoning via answer entropy. In *4th Muslims in ML Workshop co-located with ICML 2025*.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *Proceedings of the 40th International Conference on Machine Learning*, pages 19274–19286.

Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. 2024. Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning. In *The Twelfth International Conference on Learning Representations*.

Zeju Li, Jianyuan Zhong, Ziyang Zheng, Xiangyu Wen, Zhijian Xu, Yingying Cheng, Fan Zhang, and Qiang Xu. 2026. Making slow thinking faster: Compressing LLM chain-of-thought via step entropy. In *The Fourteenth International Conference on Learning Representations*.

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

Xiaoou Liu, Tiejin Chen, Longchao Da, Chacha Chen, Zhen Lin, and Hua Wei. 2025. Uncertainty quantification and confidence calibration in large language models: A survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pages 6107–6117.

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.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 305–329.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017.

Minjia Mao, Bowen Yin, Yu Zhu, and Xiao Fang. 2025. Early stopping chain-of-thoughts in large language models. *Preprint*, arXiv:2509.14004.

OpenAI. 2024. OpenAI o1 system card. *Preprint*, arXiv:2412.16720.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Tran, Yi Tay, and Donald Metzler. 2022. Confident adaptive language modeling. *Advances in Neural Information Processing Systems*, 35:17456–17472.

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. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 677–693.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36:8634–8652.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, Hanjie Chen, and Xia Hu. 2025. Stop overthinking: A survey on efficient reasoning for large language models. *Transactions on Machine Learning Research*.

Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. 2023. Language models don't always say what they think: Unfaithful explanations in chain-of-thought prompting. *Advances in Neural Information Processing Systems*, 36:74952–74965.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Zihao Wei, Liang Pang, Jiahao Liu, Wenjie Shi, Jingcheng Deng, Shicheng Xu, Zenghao Duan, Fei Sun, Huawei Shen, and Xueqi Cheng. 2025. The evolution of thought: Tracking LLM overthinking via reasoning dynamics analysis. *Preprint*, arXiv:2508.17627.

Ji Xin, Raphael Tang, Jaejun Lee, Yaoliang Yu, and Jimmy Lin. 2020. DeeBERT: Dynamic early exiting for accelerating BERT inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2246–2251.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. 2025. SoftCoT: Soft chain-of-thought for efficient reasoning with LLMs. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 23336–23351.

Yuhui Xu, Hanze Dong, Lei Wang, Doyen Sahoo, Junnan Li, and Caiming Xiong. 2026. Scalable chain of thoughts via elastic reasoning. In *The Fourteenth International Conference on Learning Representations*.

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

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36:11809–11822.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. 2023b. ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Ping Yu, Jing Xu, Jason Weston, and Ilia Kulikov. 2024. Distilling system 2 into system 1. *Preprint*, arXiv:2407.06023.

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

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*.

Level	B. Acc	B. Tokens	O. Acc	O. Tokens	CR
L_1	97.7%	1,210	97.7%	665	54.9%
L_2	96.7%	1,930	96.7%	1,150	59.6%
L_3	94.3%	2,910	94.7%	1,748	60.1%
L_4	89.3%	4,310	90.6%	2,580	59.9%
L_5	83.3%	5,780	83.3%	3,460	59.9%
All	90.8%	3,662	91.2%	2,194	59.9%

Table 3: Results stratified by official difficulty levels on MATH-500 for DeepSeek-R1-Distill-Qwen-7B. Here, "B" and "O" denote the full-CoT baseline and our method, respectively.

Dataset	Subset	B. Acc	B. Tokens	O. Acc	O. Tokens	CR
MATH-500	T	96.0%	2,902	96.0%	1,700	58.6%
	NT	70.0%	6,700	72.0%	4,170	62.2%
	All	90.8%	3,662	91.2%	2,194	59.9%
AIME 24	T	100.0%	3,000	100.0%	1,200	40.0%
	NT	37.0%	11,457	48.1%	7,182	62.7%
	All	43.3%	10,611	53.3%	6,584	62.0%

Table 4: Results for trivial (T) vs. non-trivial (NT) subsets on DeepSeek-R1-Distill-Qwen-7B. Here, "B" and "O" denote the full-CoT baseline and our method, respectively.

Dataset	$C\to C$	$C\to I$	$I\to C$	$I\to I$	Net Gain
MATH-500	446	8	10	36	+2
AIME 24	12	1	4	13	+3

Table 5: Flip matrix between the full-CoT baseline and our method on DeepSeek-R1-Distill-Qwen-7B. "Net Gain" denotes the net gain in correct problems (i.e., $I\to C$ minus $C\to I$).

A Additional Experiments

A.1 Difficulty-Stratified Analysis

To complement the overall results in Table 1, we further analyze where the compression gains come from by stratifying the MATH-500 dataset along two axes: (i) the official difficulty levels and (ii) whether a direct no-thinking answer is already correct. All results are reported for DeepSeek-R1-Distill-Qwen-7B.

Results for Different Difficulty Levels MATH-500 inherits the five official difficulty levels (L_1–L_5) from the Hendrycks MATH dataset, with per-level question counts $|L_1| = 43$, $|L_2| = 90$, $|L_3| = 105$, $|L_4| = 128$, and $|L_5| = 134$. Table 3 reports the accuracy and compression ratio at each level. The compression ratio remains within a narrow range (54.9%–60.1%) across all five levels, and even at the hardest level L_5, we still save roughly 40% of the reasoning tokens with no loss in accuracy. This indicates that the efficiency gains are distributed across difficulty levels rather than being concentrated on easy problems.

Results for Trivial vs. Non-Trivial Subsets To further isolate the contribution of CoT compression from the mere skipping of easy problems, we split each benchmark by whether the *no-thinking* baseline (direct answer generation without CoT) already answers correctly. We refer to the first subset as *trivial* and the second as *non-trivial* since the latter genuinely requires multi-step reasoning. Table 4 shows the breakdown of MATH-500 and AIME 24. On both datasets, the non-trivial subsets still exhibit strong compression (CR of 62.2% on MATH-500 and 62.7% on AIME 24), accompanied by accuracy improvements of $+2.0\%$ and $+11.1\%$, respectively. This suggests that the efficiency gains of shortcut decoding do not come primarily from skipping problems for which CoT is unnecessary.

A.2 Overthinking Flip Analysis

Overall accuracy rates in Table 1 do not, on their own, reveal *why* shortcut decoding can outperform the full-CoT baseline. To clarify the underlying mechanism, we compute a per-example *flip matrix*: for every problem, we record whether the full-CoT baseline and our method each produce a correct or incorrect final answer, yielding four cases: $C\to C$ (both correct), $C\to I$ (baseline correct, ours incorrect; our early exit degrades the performance), $I\to C$ (baseline incorrect, ours correct; our early exit improves the performance), and $I\to I$ (both incorrect).

Table 5 reports the flip matrix on AIME 24 and MATH-500. First, beneficial flips ($I\to C$) consistently outnumber harmful flips ($C\to I$) on both benchmarks (4:1 on AIME 24 and 10:8 on MATH-500), so the accuracy gain is not an artifact of random fluctuation. Second, the existence of $C\to I$ cases confirms that the controller does occasionally cut too early on some problems; this is an expected failure mode. Qualitatively, the $I\to C$ cases typically exhibit a "solved-then-overwritten" pattern: the model reaches a correct intermediate conclusion relatively early in the CoT, but subsequent redundant self-verification, alternative attempts, or reformatting steps eventually override that correct answer with an incorrect one. Our dual-signal controller terminates generation once the hidden-state probe and the step-level entropy jointly indicate convergence, thereby preventing late-stage seman-

Prompt Style	B. Acc	O. Acc	CR
Natural CoT	91.4%	92.0%	60.5%
Numbered Steps	90.8%	91.2%	59.9%
Soft Delimiter	91.2%	91.8%	60.3%

Table 6: Prompt-style sensitivity analysis on MATH-500 with DeepSeek-R1-Distill-Qwen-7B.

tic drift. This provides a mechanism-level explanation for the counterintuitive observation that removing redundant reasoning tokens can improve final-answer accuracy.

A.3 Prompt-Style Sensitivity for Step-Boundary Detection

Our controller operates at step boundaries segmented by paragraph-level cues (e.g., "\n\n" followed by a step indicator), rather than by model-specific lexical markers. To assess whether this design relies on a particular prompt format, we examine the behavior of shortcut decoding under three commonly used CoT prompt styles: (i) *Numbered Steps*, i.e., an explicit numbered style "(1)…(2)…(3)…" (used in the experiments by default), (ii) *Natural CoT*, i.e., simply asking the model to reason step by step without any formatting constraints, and (iii) *Soft Delimiter*, i.e., encouraging the model to separate each reasoning step with a blank line but without explicit numbering.

Table 6 reports the results on MATH-500 with DeepSeek-R1-Distill-Qwen-7B under the three styles. The compression ratio varies by less than 1% across all three styles (with a standard deviation of $\approx 0.3\%$), and final-answer accuracy also remains very stable (with a standard deviation of O. Acc $\approx 0.4\%$). In all these cases, the paragraph-level step segmentation continues to identify well-formed reasoning steps without any style-specific tuning. This supports our point that the framework is largely insensitive to the specific phrasing of the CoT prompt, provided that paragraph boundaries are reasonably preserved.