ChatGPT

Early Stopping in Chain-of-Thought (CoT) Reasoning

Researchers have proposed various inference-time strategies to halt CoT generation once a model has effectively "thought enough," cutting verbosity without hurting accuracy. One class uses *answer consistency*: if a model's predicted answer stabilizes, stop further steps. For example, Liu & Wang (2025) find that LLMs often reach their final answer early. They introduce **Answer Convergence** methods: (1) *Answer-Consistency* halts when two successive reasoning chunks yield the same answer, and (2) *Think-Token Adjustment* encourages the model to emit an explicit end-of-reasoning token ("`</think>`") sooner [1]. They also train a supervised **Learn-to-Stop** probe on hidden activations to predict the stopping point. Across math and question-answering benchmarks (NaturalQuestions, GSM8K, etc.), these methods cut token usage dramatically (e.g. ~30–48% fewer tokens on NaturalQuestions) with little or no accuracy loss [2] [3]. In fact, on NQ their unsupervised "answer consistency" rule alone saved 40% of tokens *and* slightly improved accuracy [3].

Probing Hidden States for Early Exit

Several works train lightweight classifiers ("probes") on the model's internal states to judge whether the current partial reasoning is sufficient. Afzal et al. (2025) ("Knowing Before Saying") show that a probe on a frozen LLM's early-layer representations can predict success of a CoT chain *before any token is generated*, achieving ~60–76% accuracy [4]. This implies much of the final-answer information is encoded early. Similarly, Zhang et al. (2025) ("Self-Verification") train a probe on hidden states at each intermediate answer to classify its correctness. This probe is well-calibrated and can "look ahead" – it often knows the final outcome before the answer is fully written. Using the probe's confidence score as a threshold for early exit, they cut inference tokens by ~24% without losing accuracy [5]. Fu et al. (2025) introduce **Dynasor-CoT**, a related *certainty-probing* approach: a "Probe-In-The-Middle" evaluates hidden states mid-chain to decide if reasoning should end. This training-free method yields up to ~29% token savings on math benchmarks (AMC, AIME, MATH500) while keeping full accuracy [6]. These methods confirm that hidden-state probes can reliably signal when a correct answer is already within reach, enabling early stopping of CoT.

Entropy- and Confidence-Based Heuristics

Other methods monitor the model's answer distribution or confidence as a stopping signal. Laaouach et al. (2025) propose **HALT-CoT**, which after each reasoning step computes the *Shannon entropy* of the model's answer distribution. When the entropy falls below a preset threshold, generation stops. This simple, training-free rule (just using the streamed token probabilities) is model-agnostic. In practice, HALT-CoT saved ~15–30% of decoding (tokens) on GSM8K, StrategyQA, and CommonsenseQA with state-of-the-art LLMs, while the final accuracy remained within ~0.4% of full CoT [7]. The authors note that answer uncertainty typically drops monotonically, validating entropy as a halting signal [7].

Likewise, Yang et al. (2025) introduce **DEER**, a dynamic-early-exit heuristic using model confidence and special "Wait" tokens. In DeepSeek-style prompts, the model emits keywords like "Wait" to chain reasoning paths. DEER treats each "Wait" as a potential exit point: it temporarily replaces "Wait" with a "final answer" token to generate a trial answer, then measures that answer's confidence. If the confidence is above a threshold, DEER stops; otherwise it continues reasoning. This plug-and-play method needs no extra training and yields large gains: CoT lengths shrink by ~31–43% on tasks (AIME, GPQA, etc.) *and* accuracy actually rises by ~1.7–5.7% (since it avoids overthinking) [8] .

Confidence can also improve *multi-sample* decoding. Li et al. (2024) proposed **Early-Stopping Self-Consistency (ESC)** to speed up self-consistency decoding: rather than drawing a fixed large number of CoT samples, ESC samples in small "windows" and stops when all answers in a window agree. They monitor the entropy of the sampled answers, and when a window's entropy hits zero (all answers identical), ESC halts further sampling [9] . On benchmarks like MATH, GSM8K, StrategyQA, etc., ESC slashed the number of samples by huge margins (e.g. –33.8% on MATH, –80.1% on GSM8K) while matching full self-consistency performance [10] . Finally, Taubenfeld et al. (2025) propose **Confidence-Informed Self-Consistency (CISC)**: after drawing multiple CoT samples, they weight each answer by the model's own confidence in that answer and take a weighted vote. This prioritizes high-confidence chains and finds the correct answer with fewer samples. Across nine models and four datasets, CISC matched or beat vanilla self-consistency while requiring ~40% fewer sampled chains [11] .

Prompt Signals and Tokens

Some approaches use explicit "stop" tokens or prompt tricks. For instance, in Liu & Wang's Answer Convergence work, inserting an end-of-thought token (e.g. "</think>") after each partial chain and training the model to emit it sooner (Think-Token Adjustment) helped truncate the reasoning [1] . Yang et al.'s DEER relies on detecting "Wait" or "double-check" phrases (inserted by the model) as branching points to evaluate a trial answer. In general, these methods exploit textual cues in the generated CoT (keywords or special tokens) as signals that a reasoning phase is complete. By inducing the model to output such tokens (or by monitoring them), one can trigger an early stop.

In summary, a range of methods have been experimentally validated: probes on hidden states can predict answer correctness, entropy or confidence thresholds can halt generation, and consensus checks (answer consistency or weighted voting) can end multi-sample reasoning early. All report substantial efficiency gains (often tens of percent fewer tokens or samples) with negligible accuracy loss [2] [5] [7] [10] . These inference-time strategies – from simple heuristics to supervised probes – show that LLMs often "know" their answer before exhausting a full chain-of-thought, and can thus stop sooner without sacrificing performance.

Sources: Recent research has systematically evaluated these techniques on benchmarks. Key references include Liu & Wang (2025) on answer convergence [2] [1] , Afzal et al. (2025) and Zhang et al. (2025) on hidden-state probes [4] [5] , Yang et al. (2025) on dynamic exit via confidence [8] , Laaouach et al. (2025) on entropy-based halting [7] , Li et al. (2024) on Early-Stopping Self-Consistency [9] [10] , and Taubenfeld et al. (2025) on confidence-weighted self-consistency [11] . Each of these reports empirical improvements in decoding efficiency.

---

[1] [2] [3] Answer Convergence as a Signal for Early Stopping in Reasoning
https://arxiv.org/html/2506.02536v1

[4] [2505.24362] Knowing Before Saying: LLM Representations Encode Information About Chain-of-Thought Success Before Completion
https://ar5iv.labs.arxiv.org/html/2505.24362v2

[5] Reasoning Models Know When They're Right: Probing Hidden States for Self-Verification
https://arxiv.org/html/2504.05419v1

6 Reasoning Without Self-Doubt: More Efficient Chain-of-Thought Through Certainty Probing | OpenReview
https://openreview.net/forum?id=wpK4IMJfdX&referrer=%5Bthe%20profile%20of%20Zheyu%20Fu%5D(%2Fprofile%3Fid%3D~Zheyu_Fu1)

7 ICML HALT-CoT: Model-Agnostic Early Stopping for Chain-of-Thought Reasoning via Answer Entropy
https://icml.cc/virtual/2025/50550

8 Dynamic Early Exit in Reasoning Models
https://arxiv.org/html/2504.15895v1

9 10 [2401.10480] Escape Sky-high Cost: Early-stopping Self-Consistency for Multi-step Reasoning
https://ar5iv.org/html/2401.10480v1

11 [2502.06233] Confidence Improves Self-Consistency in LLMs
https://arxiv.org/abs/2502.06233