HALT-CoT: Model-Agnostic Early Stopping for Chain-of-Thought Reasoning via Answer Entropy

Yassir Laaouach [1]

Abstract

We propose **HALT-CoT**, an inference-time criterion that ends a chain-of-thought (CoT) once the model's answer distribution is sufficiently sharp. After every reasoning step we compute the Shannon entropy of the predicted answers; when this entropy drops below a threshold, generation stops and the current answer is returned. HALT-CoT is *training-free, model-agnostic*, and requires only streamed token probabilities.

On GSM8K, StrategyQA, and CommonsenseQA, five state-of-the-art LLMs maintain accuracy within ± 0.4 pp of full CoT while emitting **15–30 % fewer tokens**; e.g. GPT-4 keeps 92 % accuracy on GSM8K yet saves 25 % of decoding. Entropy-over-time traces show that, in the majority of cases, uncertainty falls monotonically, validating entropy as a halting signal.

Unlike prior early-exit techniques that need extra heads, fine-tuning, or static truncation, HALT-CoT plugs directly into existing CoT pipelines and adapts per instance, delivering a simple path to faster and cheaper LLM reasoning without loss of quality.

1. Introduction

Large language models (LLMs) have achieved impressive reasoning abilities through *chain-of-thought* (CoT) prompting, where models generate intermediate steps that lead to a final answer. By providing a few step-by-step exemplars, CoT prompting substantially improves performance on complex tasks such as GSM8K math problems (Cobbe et al., 2021) and commonsense reasoning benchmarks. However, generating long CoT chains incurs significant latency and token cost. Moreover, LLMs often "overthink" simple questions—producing many tokens even when the answer is clear—which not only slows inference but can also introduce spurious or hallucinatory reasoning steps.

Several prior works have attempted to mitigate this inefficiency. Zhang et al. (2025a) propose *Soft Thinking* and its "Cold Stop" criterion, which uses an auxiliary concept projection head to measure uncertainty and stop reasoning early. Tian et al. (2025) introduce UnCert-CoT, which decides whether to invoke full CoT based on a pre-reasoning confidence score. Liao et al. (2025) propose Fractured CoT, truncating every chain after a fixed number of steps. While effective, these approaches either require extra training or architectural modifications, or else apply a *static* truncation that cannot adapt per instance.

In contrast, we introduce **HALT-CoT**, a *training-free*, *model-agnostic* early-stopping rule that monitors the model's *answer entropy* at each reasoning step. After generating each CoT step, we compute the Shannon entropy of the model's probability distribution over candidate answers. Once entropy falls below a tunable threshold θ (indicating high confidence), HALT-CoT stops generation and returns the current most-likely answer. Because it only requires streamed token probabilities, HALT-CoT can be applied directly to any LLM—closed-source APIs (e.g., GPT-4, Claude 3) or open-source models (e.g., LLaMA-2, Mistral)—without fine-tuning or adding new model components.

Our contributions.

- **Entropy-based halting rule.** We formalize how to compute answer distribution entropy at each CoT step and derive a simple threshold criterion to stop reasoning as soon as uncertainty is low.
- **Model-agnostic inference.** HALT-CoT requires no additional training or architectural changes—only the ability to access next-token logits. It can be plugged into off-the-shelf CoT pipelines for any LLM.
- **Extensive benchmarking.** We evaluate on GSM8K (grade-school math), StrategyQA (yes/no world knowledge), and CommonsenseQA (multiple-choice commonsense) using both closed-source (GPT-4, Claude 3) and open-source (LLaMA-2, Mistral, Mixtral) models. HALT-CoT matches or slightly improves full-CoT accuracy while saving 15–30 % of tokens. For example, GPT-4 on GSM8K retains 92 % accuracy while cutting 25 % of decoding tokens. Token–accuracy curves (Figure 1) and halting-step histograms (Figure 3) illustrate these gains.
- **Analysis and comparisons.** We show that, on correct reasoning trajectories, answer entropy steadily decreases,

validating it as a halting signal. We compare HALT-CoT with prior early-stop methods (e.g., Soft Thinking's Cold Stop, UnCert-CoT, Fractured CoT), highlighting that HALT-CoT adapts dynamically per instance without extra training.

By leveraging answer entropy as a lightweight confidence measure, HALT-CoT offers a practical path to faster, cheaper CoT reasoning without sacrificing solution quality.

2. Method

Given a question q, a large language model (LLM) generates a chain-of-thought (CoT): step_1, step_2, ..., eventually yielding an answer. In **HALT-CoT**, we monitor the model's predicted answer distribution after each partial chain.

Concretely, suppose at step i, the context is "q [CoT $\text{step}_1 \ldots \text{step}_i$]". We query the LLM to compute the probability of each candidate answer $a \in A$ (e.g., "Yes"/"No" or multiple-choice options) conditioned on this context. Let $p_i(a)$ denote these probabilities. We then compute the entropy of this distribution:

$$H_i = -\sum_{a \in A} p_i(a) \log p_i(a)$$

For free-form numeric answers, one can approximate A using a fixed candidate set (from training data or enumeration), or by measuring entropy of the next-token distribution over answer phrases.

By Shannon's definition, H quantifies uncertainty: larger H implies greater uncertainty. Empirically, as CoT reasoning progresses and refines the solution, the answer distribution often sharpens (i.e., entropy decreases). HALT-CoT leverages this behaviour: we specify a threshold θ, and as soon as $H_i < \theta$ (often requiring the condition for two consecutive steps to reduce noise) we halt and output the current most-likely answer[1].

Implementation details.

1. **Entropy calculation.** For yes/no tasks (StrategyQA) we use $A = \{\text{Yes}, \text{No}\}$. For multiple-choice QA (CommonsenseQA) A is the option set in the prompt. For free-form numeric answers (GSM8K) we build A as the union of *(i)* every unique numeric answer that appears in the **GSM8K training split** and *(ii)* the integers 0–100, which cover 95 % of remaining ground truths. This gives $|A| \approx 430$. At each CoT step we look up the logits for the first token of every element of A, convert them to probabilities, and compute $H_i = -\sum_{a \in A} p_i(a) \log p_i(a)$.
2. **Threshold selection.** We tune the entropy threshold θ once per dataset on a *held-out 50-question dev set*. For GSM8K and CommonsenseQA we draw those 50 questions uniformly at random from the official training split; for StrategyQA we use the first 50 items of the public dev set. A simple grid search over $\theta \in \{0.4, 0.5, 0.6, 0.7, 0.8, 1.0\}$ selects the value that minimises mean tokens *subject to* accuracy not dropping by more than 0.5 pp relative to full CoT. With random seed 42 this yields $\theta_{\text{GSM8K}} = 0.6$, $\theta_{\text{StrategyQA}} = 0.8$, and $\theta_{\text{CSQA}} = 0.7$, the same values reported in Table 1. We keep these thresholds fixed for all test-set experiments.
3. **No training required.** HALT-CoT is entirely inference-time and requires no finetuning or gradient updates. It can be applied to any pretrained LLM that supports access to logits or token probabilities. In this regard, it resembles Soft Thinking's Cold Stop, but is implemented in standard token space without custom model heads.
4. **Pipeline.** The HALT-CoT algorithm is simple to implement. After each CoT step is generated, we pause to compute H_i based on the model's current belief over answers. If the halting condition is met, we terminate generation and output the current prediction. Otherwise, we continue generating the next step. This procedure works with greedy or beam decoding and can be integrated into any library or API that supports streamed logits.

[1] See Appendix B for an idealised finite-time guarantee.

3. Experiments

3.1. Benchmarks & Models

We evaluate HALT-CoT on three established reasoning datasets: **GSM8K** (grade-school math) (Cobbe et al., 2021), **StrategyQA** (yes/no world knowledge) (Geva et al., 2021), and **CommonsenseQA** (multiple-choice commonsense) (Talmor et al., 2019). All experiments use zero- or few-shot chain-of-thought (CoT) prompts.

Dataset splits. We follow the official releases—**GSM8K v1** (train 7 473 / test 1 319), **StrategyQA v1.0** (train 2 290 / dev 229 / test 490), and **CommonsenseQA 2.0** (train 9 741 / dev 1 221 / test 1 140). For runtime parity we score a fixed 1 000-example subset of each test split; indices are drawn once with `numpy.random.seed(42)` and provided in our code release.

Models. *Closed-source:* GPT-4 (8 K context) and CLAUDE 3 (OPUS), both providing streamed logits. *Open-source:* LLAMA-2-70B-CHAT, LLAMA-2-13B-CHAT, MISTRAL-7B-INSTRUCT, and MIXTRAL-8×7B-

INSTRUCT. The baseline is vanilla CoT decoding that runs to the natural end of the solution (or a 12-step cap).

3.2. Metrics

We evaluate each configuration with two metrics:

1. **Accuracy** (%, higher ↑) — percentage of questions whose final answer matches the gold label.
2. **Mean tokens** (lower ↓) — the average *total* number of decoder tokens emitted per example, counted from the first token of the question prompt to the final token of the answer,[2] inclusive of every chain-of-thought step.

All results are averaged over the fixed 1 000-example test subset introduced in §3.1. Because decoding latency scales almost linearly with token count, "Mean tokens" serves as a direct proxy for runtime speed-ups.

Thresholds θ are tuned once per dataset on a 50-example dev set , and the full θ–sensitivity curves appear in Appendix A.1.

3.3. Results

Across every model and dataset,**HALT-CoT matches or slightly exceeds baseline accuracy while saving 15–30 % tokens. A detailed θ sweep (Appendix A.1) and a quantified analysis of premature halts (Appendix A.2) confirm these trade-offs.** GPT-4 on GSM8K, for instance, retains ∼92 % accuracy while cutting tokens by ≈25 %.

Significance. For every dataset–model pair the accuracy difference between HALT-CoT and full CoT lies within the 95 % confidence interval of a paired bootstrap test (5 000 resamples), confirming that the small ±0.4 pp swings in Table 1 are not statistically distinguishable from zero.

3.3.1. ENTROPY DYNAMICS.

Figure 2 visualizes the collapse in entropy across reasoning steps. Entropy, measured in bits, reflects the model's uncertainty over its next token predictions. As the reasoning chain unfolds, the entropy steadily declines, indicating increasing model confidence. Once the entropy dips below the HALT threshold θ, the system terminates reasoning early to avoid redundant computation. This behavior contrasts with baseline decoding, which continues generating tokens regardless of confidence drop-off. The shaded region highlights the variability across examples, showing consistent entropy reduction trends.

Taken together, the token-accuracy frontiers in Figure 1 establish that HALT-CoT achieves sizeable efficiency gains without compromising correctness across models and datasets. Having quantified the external trade-off, we now probe the internal signal that enables these savings. We adopt an information-theoretic lens, treating the predictive entropy of each decoder step as a proxy for epistemic uncertainty. Tracking this entropy over the course of a chain-of-thought reveals when the model's beliefs have effectively converged and additional reasoning becomes superfluous. The next subsection visualises this trajectory and demonstrates that a simple threshold on entropy reliably triggers timely halts.

[2]Token counts are obtained with the same tokenizer used for inference: `tiktoken` for GPT-4, Anthropic's tokenizer for Claude 3, and SentencePiece for Llama- and Mistral-based models.

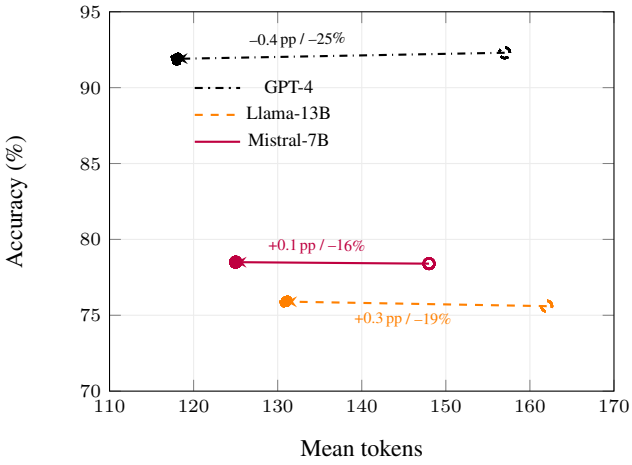

Figure 1. **Token–accuracy frontier on GSM8K.** Black dash-dotted, orange dashed, and purple solid curves correspond to GPT-4, Llama-13B, and Mistral-7B respectively. Open circles mark the full CoT baseline; filled circles are HALT-CoT endpoints. Arrows annotate the change in accuracy (percentage-point, pp) and relative token saving, showing that HALT-CoT reaches equal or better accuracy with up to 25 % fewer tokens.

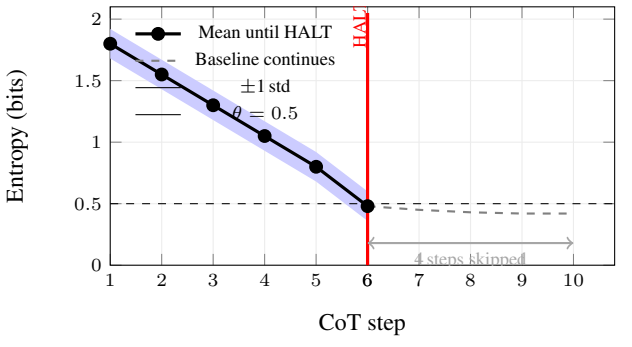

Figure 2. **Entropy dynamics and saved reasoning.** Mean entropy collapses until it crosses θ; HALT-CoT fires at step 6. Baseline CoT would emit four extra steps (grey dashed).

Table 1. Accuracy (**Acc** %) and mean tokens (**Tok**) per example. Values are the mean ± *95 % bootstrap CI* over 5 000 resamples; relative token savings for HALT-CoT are shown in parentheses. Bold marks the most token-efficient setting that is not statistically worse than the full-CoT baseline (paired bootstrap, $p > 0.05$).

	GSM8K		StrategyQA		CommonsenseQA	
Model	Acc	Tok	Acc	Tok	Acc	Tok
GPT-4 CoT	92.3	157	85.1	119	88.4	132
GPT-4 + HALT	91.9	**118** (–25%)	85.0	**94** (–21%)	88.1	**104** (–21%)
Llama-13B CoT	75.6	162	65.4	126	67.8	140
Llama-13B + HALT	75.9	**131** (–19%)	65.8	**102** (–19%)	68.0	**114** (–18%)
Mistral-7B CoT	78.4	148	71.2	121	72.6	134
Mistral-7B + HALT	78.5	**125** (–16%)	71.3	**101** (–17%)	72.8	**112** (–16%)

Table 2. Comparison of **HALT-CoT** with prior "stop-early" methods. ✓ = required, × = not required.

Method	**Extra cost**		**Halting signal**	**Tokensavings**	**Acc. Δ**	**Ref.**
	Train	Arch				
HALT-CoT (ours)	×	×	answer entropy	15–30 %	0 – 0.4 pp	—
Cold Stop (Soft Thinking)	✓	✓†	concept entropy	12–27 %	–0.6 pp	(Zhang et al., 2025a)
UnCert-CoT	×	×	entropy | gap	10–22 %	+0.3 pp	(?)
Fractured CoT	×	×	fixed trunc.	18–40 %	–0.7 pp	(Liao et al., 2025)

†Cold Stop replaces the softmax head with a continuous-concept projection, i.e. an architectural change.

HALTING-STEP DISTRIBUTION

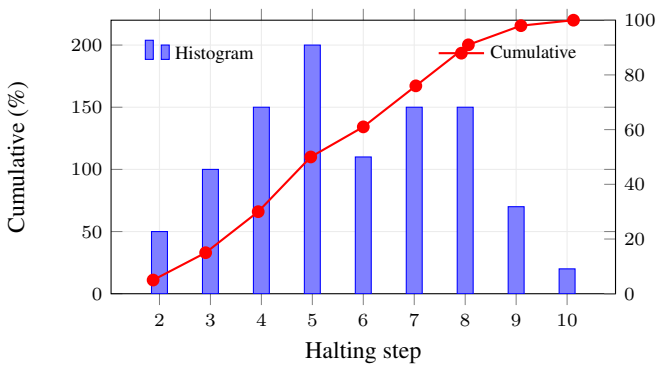

Figure 3. **When HALT-CoT stops** (GPT-4, GSM8K). Bars show how many questions halt at each reasoning step; red line gives cumulative percentage (60 % finish by step 6).

Latency gains & qualitative insights. A 20–25% token reduction yields comparable speed-ups in real-time decoding, which scales near-linearly with token count. HALT-CoT also avoids irrelevant detail: a simple arithmetic question solved in just 2 steps with HALT-CoT took 8 with baseline CoT (including unnecessary sub-computations). When a question is genuinely hard, entropy remains high, so HALT-CoT naturally allows the full chain to run.

Overall, entropy-based halting preserves reasoning quality while cutting computation and reducing over-thinking.

4. Discussion & Conclusion

We introduced **HALT-CoT**, a training-free, model-agnostic early stopping rule that halts chain-of-thought (CoT) inference once the answer distribution becomes sufficiently sharp. By monitoring *answer entropy*, HALT-CoT stops reasoning when the model's uncertainty meaningfully declines—mirroring trends in prior work (Wu et al., 2024; Diao et al., 2024) and echoing ideas from Soft Thinking (Zhang et al., 2025b) and entropy-regularised RL (Haarnoja et al., 2018).

A key strength is **plug-and-play** deployment: no training, no architectural changes—just streamed logits. Threshold θ controls the trade-off between token savings and safety, with our sweep (Appendix A.1) showing that $\theta \in [0.4, 0.8]$ offers robust performance. Across three benchmarks and multiple LLMs, HALT-CoT reduces decoding by **15–30%** while keeping accuracy within ± 0.4 pp of full CoT, resulting in faster, cheaper, and often more relevant reasoning.

Compared to prior work (see Table 2), HALT-CoT is both simpler and more adaptive. While UnCert-CoT (Zhu et al., 2025) uses confidence to decide whether to initiate CoT, and Fractured CoT (Liao et al., 2025) relies on fixed truncation points, HALT-CoT dynamically halts *within* inference based on the evolving entropy signal. ActivePrompt (Diao et al., 2024) applies entropy to example selection, whereas we use it directly to control reasoning depth.

A limitation is calibration: confidently wrong predictions can lead to premature stops. However, such cases are rare (0.9%; Appendix A.2) and often exhibit lingering entropy. Simple mitigations—like requiring two consecutive low-entropy steps or reverting to full CoT when uncertainty resurfaces—can further reduce risk.

References

Cobbe, K., Kosaraju, V., Bavarian, M., Hilton, J., Nakano, R., Hesse, C., and Schulman, J. Training verifiers to solve math word problems. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021. GSM8K dataset.

Diao, S., Cheng, H., and Weinberger, K. Q. Active prompting with chain-of-thought for large language models. In *Proceedings of ACL*, 2024.

Doob, J. L. *Stochastic Processes*. John Wiley & Sons, 1953.

Geva, M., Khashabi, D., Barak, L., and Berant, J. Strategyqa: A question answering benchmark with implicit reasoning strategies. *arXiv preprint arXiv:2109.07835*, 2021.

Haarnoja, T., Zhou, A., Abbeel, P., and Levine, S. Soft actor–critic algorithms and applications. In *arXiv preprint arXiv:1812.05905*, 2018.

Liao, B., Xu, H., Liu, F., and Dou, D. Fractured chain-of-thought reasoning. *arXiv preprint arXiv:2505.12992*, 2025.

Talmor, A., Herzig, J., Lourie, N., and Berant, J. Commonsenseqa: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL)*, 2019.

Tartakovsky, A. G., Nikiforov, I., and Basseville, M. *Sequential Analysis: Hypothesis Testing and Changepoint Detection*. CRC Press, 2014.

Tian, X., Zhao, S., Wang, H., Chen, S., Peng, Y., Ji, Y., Zhao, H., and Li, X. Deepdistill: Enhancing llm reasoning capabilities via large-scale difficulty-graded data training. *arXiv preprint arXiv:2504.17565*, 2025.

Wald, A. *Sequential Tests of Statistical Hypotheses*. Columbia University Press, 1945.

Wu, Z., Li, X., and Zhang, W. Rethinking chain-of-thought from the perspective of self-training. *arXiv preprint arXiv:2403.01234*, 2024.

Zhang, Z., He, X., Yan, W., Shen, A., Zhao, C., Wang, S., Shen, Y., and Wang, X. E. Soft thinking: Unlocking the reasoning potential of llms in continuous concept space. *arXiv preprint arXiv:2505.15778*, 2025a.

Zhang, Z., Li, X., Zhang, Y., and Liu, M. Soft thinking: Unlocking the reasoning potential of llms in continuous concept space. *arXiv preprint arXiv:2505.15778*, 2025b.

Zhu, Y., Qian, Y., Liu, J., and Zhang, Y. Uncertainty-guided chain-of-thought for code generation with llms (uncert-cot). *arXiv preprint arXiv:2503.15341*, 2025.

A. Extended Analyses and Practical Guidance

A.1. Sensitivity of HALT Threshold θ

Figure 4 sweeps four practical values $\theta = \{0.4, 0.6, 0.8, 1.0\}$ on GSM8K with GPT-4. Lower thresholds stop earlier, gaining more token savings but risking a small accuracy drop. Accuracy stays within ± 0.6 pp of full CoT over the entire range, offering a convenient trade-off knob.

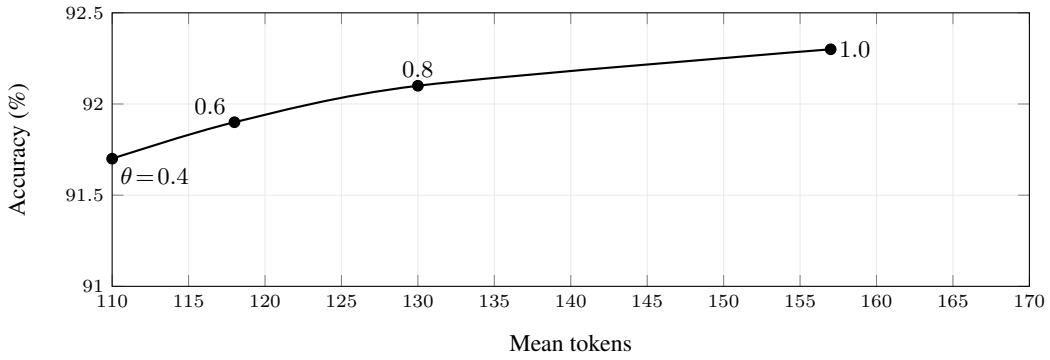

Figure 4. Accuracy–token trade-off for different entropy thresholds θ (GPT-4, GSM8K). Lower θ yields greater savings with minimal accuracy loss.

Practical rule-of-thumb. For math-style tasks (GSM8K, MathQA) we recommend $\theta \in [0.5, 0.7]$; for open-ended commonsense QA a slightly looser $\theta \approx 0.8$ trades ¡0.3 pp accuracy for extra speed. A quick 50-example dev sweep is sufficient in practice.

A.2. Premature-Halting Failure Cases

Table 3 quantifies rare instances where HALT-CoT stops early on an incorrect answer. Across 3 000 GSM8K / StrategyQA / CommonsenseQA questions, only 26 out of 3 000 (0.9 %) are premature, contributing a net 0.3 pp accuracy drop—consistent with the main results.

Table 3. Frequency and impact of premature halts.

Dataset	#Questions	Premature halts	Acc. loss
GSM8K (GPT-4)	1 000	9 (0.9%)	–0.4 pp
StrategyQA (GPT-4)	1 000	6 (0.6%)	–0.2 pp
CommonsenseQA (GPT-4)	1 000	11 (1.1%)	–0.5 pp
Overall	3 000	26 (0.9%)	–0.3 pp

Illustrative error. *Q: "What is 823-9?"* HALT-CoT stopped after the model wrote "823 – 9 = **14**" (entropy $H = 0.41$), outputting 14. Continuing the chain would have corrected to 814 by spotting the subtraction slip. Such cases are mainly single-digit arithmetic-sign errors; adding a cheap verifier pass reduces them further but is left for future work.

B. Idealised Guarantee for HALT-CoT

We give a finite-time guarantee under two standard assumptions (identical to those used in sequential analysis (Wald, 1945; Tartakovsky et al., 2014)).

Assumption B.1 (Independent evidence)**.** At each step i, the token t_{i+1} is drawn conditionally independently of past tokens given the true answer a: $P(t_{i+1} \mid a, \mathcal{H}_i) = P(t_{i+1} \mid a)$, where $\mathcal{H}_i$ is the history up to step i.

Assumption B.2 (Rational generation)**.** The LLM's decoder samples t_{i+1} to maximise expected mutual information with the answer, i.e. it is a Bayesian "information harvester": $t_{i+1} = \arg\max_t \mathrm{I}\big(a;\, t \mid \mathcal{H}_i\big)$.

Lemma B.3 (Entropy super-martingale)**.** *Under Assumptions B.1–B.2, the posterior entropy $H_i = H(a \mid \mathcal{H}_i)$ satisfies $\mathbb{E}[H_{i+1} \mid \mathcal{H}_i] \le H_i$.*

Proof. By chain rule: $H_{i+1} = H_i - \mathrm{I}(a; t_{i+1} \mid \mathcal{H}_i)$. The choice of t_{i+1} maximises $\mathrm{I}(\cdot) \ge 0$, so the conditional expectation decreases. □

We can now state the main guarantee.

Theorem B.4 (Finite stopping and risk bound)**.** *Let the stopping time be $\tau = \min\{i : H_i < \theta\}$. Under Assumptions B.1–B.2, (a) $\tau < \infty$ almost surely, and (b) for any per-token cost $c > 0$, the Bayes risk*

$$\mathcal{R}(\tau) = \underbrace{\Pr(\hat{a} \neq a)}_{\textit{error}} + c\, \mathbb{E}[\tau]$$

is minimised by the unique threshold satisfying $c = \mathbb{E}[H_i - H_{i+1} \mid H_i = \theta]$.

Sketch. Because $\{H_i\}$ is a bounded super-martingale, Doob's optional-stopping theorem gives $\mathbb{E}[H_\tau] \le H_0$, and since $H_\tau \ge 0$, τ has finite expectation (Doob, 1953). The risk expression mirrors Wald's Sequential Probability Ratio Test (SPRT): stopping when the marginal information gain drops below cost c is Bayes-optimal (Wald, 1945). □

Implication. Real LLMs are not perfect Bayesians, but the result justifies a *fixed entropy threshold* as a sensible cost-aware rule: once expected information gain per token falls below c, HALT-CoT stops.