TERMINATOR: Learning Optimal Exit Points for Early Stopping in Chain-of-Thought Reasoning

Alliot Nagle[1] Jakhongir Saydaliev[2] Dhia Garbaya[3] Michael Gastpar[2] Ashok Vardhan Makkuva[4,†] Hyeji Kim[1,†]
[1]UT Austin [2]EPFL [3]ENS Paris-Saclay [4]Télécom Paris (IP Paris)

KEY IDEA The first arrival of an LRM's final answer in a CoT is predictable, so we train **TERMINATOR** to force an exit at that point.

1 Large Reasoning Models (LRMs) keep thinking after they have the answer

MATH-500: How many positive whole-number divisors does 196 have? Qwen3-8B

Thinking (29%) | **Overthinking** (71%)

... So applying this formula to 196, which is $2^2 \times 7^2$, we would take the exponents 2 and 2, add 1 to each, getting 3 and 3, and then multiply them: $3 \times 3 = 9$. Therefore, 196 should have 9 positive divisors. But wait, let me make sure I didn't skip any steps or make a mistake in the prime factorization. Let me double-check the prime factorization of 196. Let's start over: ...

DEFINITION **Hindsight-optimal reasoning length** $\mathrm{HORL} := \min\{\, i \in [M] : \boldsymbol{r}_{\le i} \text{ contains the earliest logical arrival of } \hat{\boldsymbol{a}} \,\}$

A finished Chain-of-Thought (CoT) $\boldsymbol{r}$ of M tokens and its final answer $\hat{\boldsymbol{a}}$ show, in hindsight, where the LRM should have stopped: every step that derives $\hat{\boldsymbol{a}}$ is kept, and the usually redundant rest is skipped. Since $\hat{\boldsymbol{a}}$ is unknown mid-reasoning, HORL is what **TERMINATOR** learns to predict.

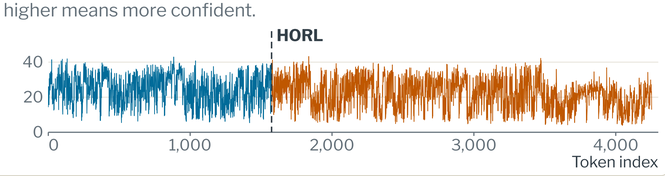
2 LRM behavior shifts at HORL

One CoT looks like noise. However, a sharp transition appears after aligning many CoTs on the HORL position.

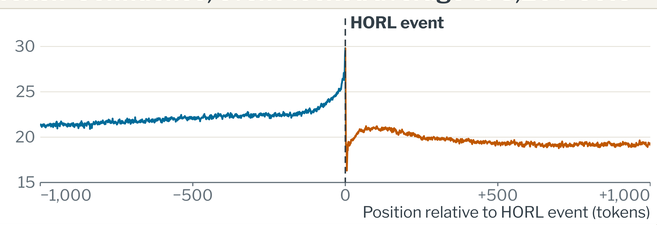
Token-Confidence of a single CoT

Qwen3-8B

Token-Confidence: mean negative log-probability of the Top-20 tokens; higher means more confident.



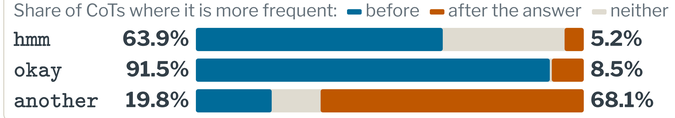
Token-Confidence, event-locked average of 3,200 CoTs



"Thinking token" frequency shift

Qwen3-8B

Share of CoTs where it is more frequent: before | after the answer | neither

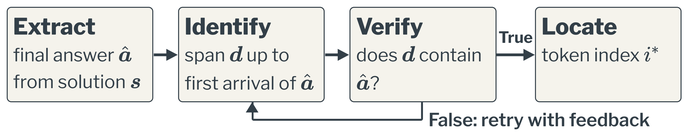


Reading these signals requires the answer position, which is what we need to find. Instead, we use the LRM's hidden states to **learn to exit at the HORL positions**.

3 TERMINATOR learns when to stop

A. Label each CoT with its hindsight-optimal exit

We use an LRM to perform a semantic search to locate HORL in every CoT.

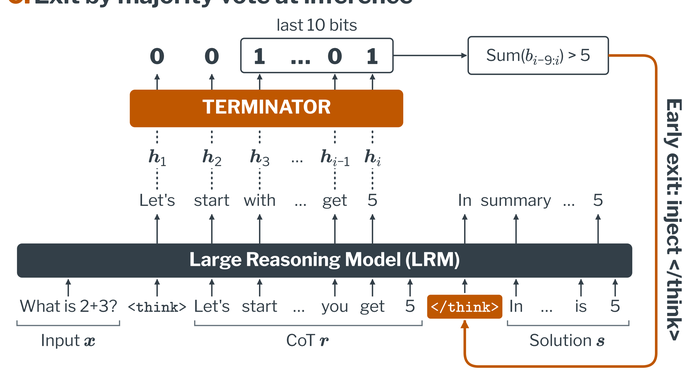


B. Train a probe on every CoT token

TERMINATOR is a single transformer layer and a binary prediction head, taking final-layer hidden states $\boldsymbol{h}_i$. The label $y_i = \mathbf{1}(i \ge i^*)$ marks whether the final answer has arrived.

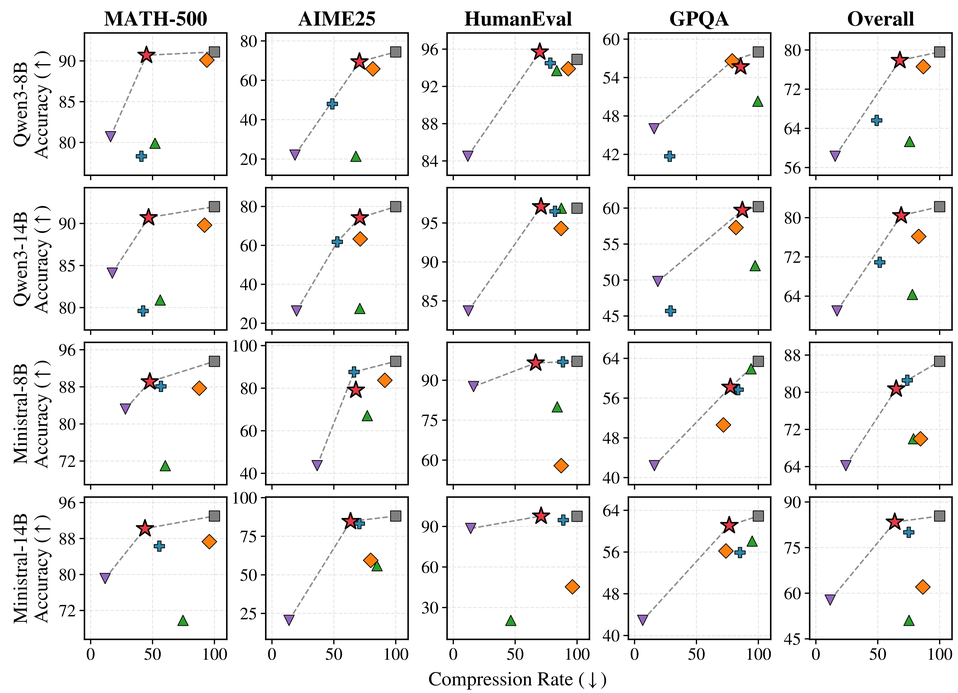
$$L(\boldsymbol{\theta}) = -\frac{1}{M}\sum_{i=1}^{M}\Big[w_1 y_i \log p_i + w_0(1-y_i)\log(1-p_i)\Big]$$

C. Exit by majority vote at inference



4 Experimental results: TERMINATOR defines the Pareto frontier

TERMINATOR is Pareto-optimal on 14 of 16 LRM–benchmark pairs



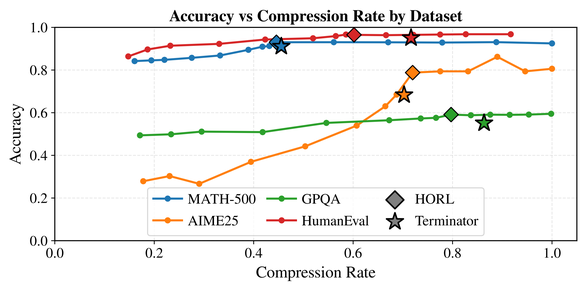
2.3× lower latency

Method	Latency (s)	Tokens/s
Qwen3-8B		
Vanilla	32.68 ± 9.59	151.5 ± 4.4
Terminator	**14.10** ± 6.27	**135.2** ± 2.0
Qwen3-14B		
Vanilla	43.38 ± 13.98	98.0 ± 2.0
Terminator	**18.76** ± 6.52	**90.6** ± 0.8

MATH-500, batch size 1, one GH200. Mean ± 95% CI.

Close to the hindsight optimum

Qwen3-8B



Stars: **TERMINATOR**. Diamonds: HORL, not achievable by any method.