# LATENTHALT: REUSING TERMINATION GEOMETRY FOR ADAPTIVE LATENT REASONING

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

Latent chain-of-thought (CoT) reduces reasoning cost by replacing textual rationales with continuous states, but fixed latent budgets leave the model without a per-problem stopping rule. We introduce LATENTHALT, which reuses the termination geometry already present in an explicit-CoT model. In a supervised fine-tuned (SFT) 1B model, last-layer states that predict $<$/think$>$ concentrate in a narrow angular cone, separable from other generation states with held-out AUROC $0.99995$. We freeze the cone's center and quantile radii, then train latent reasoning through a single joint curriculum. An auxiliary decoder supervises the content of each block, while a linear terminal hinge trains the final state of a fully latent sequence to enter the cone. At inference, the same region determines when to stop generating latent blocks and decode the answer, without a separately trained halting head. Across GSM8K-Aug, GSM-Hard, MultiArith, and SVAMP, all $3{,}818$ problems halt adaptively, using $2.43$ blocks on average. LATENTHALT is on par with fixed-length supervised latent CoT on GSM-Hard and improves SVAMP and MultiArith accuracy by $10.5$ and $7.3$ points. It uses $4.2\times$ fewer estimated reasoning-phase FLOPs than explicit CoT, and its best model comes from a single $20$-epoch run rather than the two-phase, $53$-epoch Coconut$\rightarrow$SIM-CoT pipeline. The code is available at `https://anonymous.4open.science/r/LatentHalt-EA19/`.

## 1 INTRODUCTION

Chain-of-thought (CoT) prompting enables a language model to work through intermediate steps before giving an answer (Wei et al., 2022; Kojima et al., 2022). These textual rationales improve reasoning, but their cost grows with every generated token. The rationale can be much longer than the answer it supports. Latent CoT reduces this cost by carrying intermediate computation in continuous states: the model feeds its hidden state back as the next input, bypassing the vocabulary during reasoning (Hao et al., 2024; Shen et al., 2025; Wei et al., 2026).

This change raises a stopping problem. In explicit CoT, the model can end its rationale by emitting an end-of-reasoning token. Latent reasoning produces no tokens, so that decision is no longer available through the usual decoding process. Coconut, CODI, and SIM-CoT use a fixed latent budget, commonly six blocks. A fixed budget is simple but allocates the same number of reasoning steps to every problem. It cannot recognize that a short problem is already complete or that a harder one needs more work. An adaptive latent model therefore needs a stopping signal that can be read directly from its continuous states. Figure 1 contrasts these three regimes.

Our starting point is the explicit-CoT model used to initialize latent training. Before this model emits $<$/think$>$, its last-layer hidden state already encodes the decision to end the rationale. We examine these *terminal states* and find that, after normalization, they concentrate in a narrow angular cone. This *termination region* is almost perfectly separable from other generation states (held-out AUROC $0.99995$). The model thus provides an existing geometric signal for its own stopping decision. Reading that signal requires only one normalization and one dot product against a fixed center.

LATENTHALT turns this observation into an adaptive stopping rule. We extract the region's center and angular quantile radii from SFT generations on a separate validation set, freeze them, and progressively replace explicit reasoning steps with latent blocks. An auxiliary decoder supervises each block against its corresponding explicit step from the first curriculum stage, while a region loss



Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

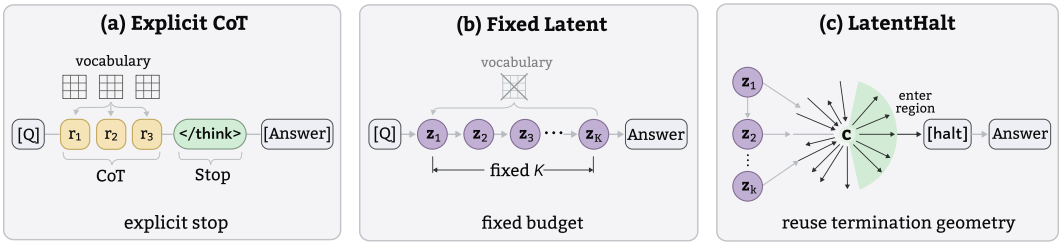



Figure 1: **Three ways to end reasoning.** **(a)** Explicit CoT decodes each step $r_i$ through the vocabulary and stops by emitting $<$/think$>$. **(b)** Fixed-latent methods (Coconut, CODI, SIM-CoT) reason in continuous states $z_i$ but spend a fixed budget of $K$ blocks. **(c)** LATENTHALT halts adaptively when a block-end state enters the frozen termination region around center $c$, then decodes the answer. Different problems enter the region after different numbers of blocks.

guides terminal latent states into the frozen cone. At inference, the model checks the region after each block. Entry triggers the pre-trained $<$/think$>$ boundary and answer decoding, so latent length is chosen per problem without a separately trained halting head.

Preserving the stopping signal during latent training is the main difficulty. Recursive continuous inputs change the model's hidden-state distribution, and the language-modeling and decoder objectives alone do not keep terminal states near the original region. The shape of the terminal penalty also matters: a linear hinge keeps a constant pull toward the cone where a squared hinge weakens near the target. Our ablations connect this difference to the model's ability to halt, and distinguish it from the decoder's role in preserving reasoning content.

The paper makes the following contributions:

- Termination geometry. We identify a concentrated, separable region of explicit-CoT terminal states that recovers the model's own end-of-reasoning decision (Section 4.4, Table 2).
- Adaptive latent reasoning. LATENTHALT reuses this region as a per-sample stopping rule, combining step supervision and geometric constraints in a single curriculum from the SFT checkpoint (Section 3).
- Preserving the stopping signal. We explain how the terminal hinge behaves near its boundary and test the separate contributions of penalty shape and step supervision to halting and accuracy (Sections 3.4 and 4.6).
- Empirical evidence. Across four benchmarks, LATENTHALT achieves $100\%$ adaptive halting at $2.43$ blocks on average, matches SIM-CoT on GSM-Hard, and improves MultiArith and SVAMP while cutting estimated reasoning-phase FLOPs by $76\%$ relative to explicit CoT (Section 4).

## 2 RELATED WORK

**Latent and Implicit Chain-of-Thought.** Coconut establishes a direct route from explicit to continuous reasoning: it feeds the last hidden state back as the next input and gradually replaces textual steps through a training curriculum (Hao et al., 2024). CODI transfers explicit reasoning into latent states through self-distillation (Shen et al., 2025). SIM-CoT introduces an auxiliary decoder that reconstructs the explicit step associated with each latent block (Wei et al., 2026). LATENTHALT builds on this step supervision, but introduces it at the start of the curriculum and trains it jointly with a geometric stopping constraint.

Related approaches internalize explicit rationales by progressively removing tokens during fine-tuning (Deng et al., 2023; 2024). Others explore mixed latent and text tokens (Su et al., 2025), compressed reasoning states (Cheng & Van Durme, 2024), continuous concept spaces (Zhang et al., 2025; Xu et al., 2025), or transitions between latent and explicit reasoning (Shi et al., 2026). Li et al. (2026) study the causal structure of latent CoT. These works examine how reasoning can be represented without a full textual trace. Our focus is the complementary question of how a continuously reasoning model can recognize when to stop.

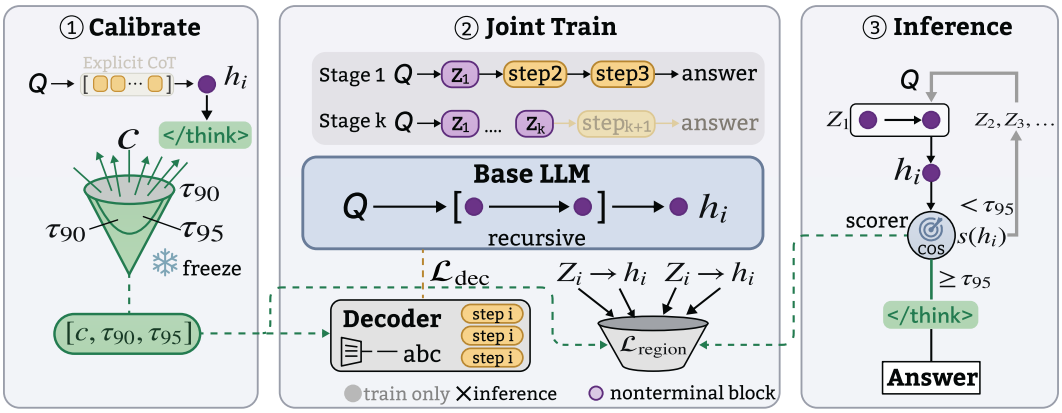



Figure 2: **LATENTHALT in three stages. (1) Calibrate:** on the explicit-CoT SFT model, fit the termination region, freeze its center $c$ and radii $\tau_{90}, \tau_{95}$. **(2) Joint train:** a prefix-replacement curriculum turns steps into latent blocks in the recursive base LLM, supervised by a step decoder ($\mathcal{L}_{\text{dec}}$) and the region loss ($\mathcal{L}_{\text{region}}$). **(3) Inference:** score each block end against $c$ and halt once it reaches $\tau_{95}$, then force `</think>` and decode the answer.

**Allocating Computation at Inference Time.** Pause and filler tokens allocate additional forward passes without requiring linguistic content at every position (Goyal et al., 2024; Pfau et al., 2024). Adaptive computation time (Graves, 2016) and PonderNet (Banino et al., 2021) address how much computation to allocate to an input by learning when to halt. Confident adaptive language modeling (Schuster et al., 2022) adjusts computation along a different axis, using confidence to exit before all transformer layers are evaluated. The distinction in LATENTHALT is the source of the stopping criterion: it reads a region already present in the SFT model and trains latent states to reach that fixed target. No additional halting module is learned.

**Efficient Reasoning and Early Stopping.** Reducing the cost of explicit reasoning has motivated work on test-time budgeting (Muennighoff et al., 2025), compression and pruning of reasoning traces (Xia et al., 2025; Hong et al., 2025), and early stopping based on answer convergence or confidence (Liu & Wang, 2025; Mao et al., 2025; Sun et al., 2026). These methods use information available during token generation. LATENTHALT pursues the same efficiency goal in a setting where intermediate tokens and answer distributions are not decoded. By reading the current hidden state, it can decide whether reasoning should continue without decoding a textual trace.

## 3 METHOD

LATENTHALT starts from an explicit-CoT SFT checkpoint and learns continuous reasoning while retaining access to its termination signal. We first extract and freeze a termination region from the checkpoint's hidden states. A curriculum then replaces textual steps with latent blocks; an auxiliary decoder supervises what those blocks represent, and a region loss constrains where they end. The same frozen region serves as the stopping rule at inference. Figure 2 shows the three stages together.

### 3.1 TERMINATION GEOMETRY AND CALIBRATION

Explicit-CoT terminal states concentrate in a narrow angular cone. We use this structure to define a termination region; Section 4.4 evaluates its concentration and separability.

**Setup.** We fine-tune Llama-3.2-1B-Instruct (Meta, 2024b) with explicit CoT using a completion-only objective and an explicit reasoning protocol: `<think>\n`, then reasoning steps, then `</think>\n`, then the answer and EOS. The boundary markers `<think>` and `</think>` are added as two special tokens. This SFT model is both our accuracy reference and the source of the termination geometry, and latent training starts from its weights.

**Calibration Data and Terminal States.** We measure termination geometry on a separate validation set. During greedy SFT generation on these calibration questions, we record the last-layer hidden state $h$ that predicts each next token. We call $h$ a *terminal state* when its vocabulary projection selects </think> as the greedy argmax. The state is recorded before </think> is fed back as an input, so it captures the decision to terminate rather than the response to a termination marker.

**The Termination Region.** We $\ell_2$-normalize each state, $z = h/\|h\|_2$, and define the region *center* as the normalized mean direction of the terminal states, $c = \big(\sum_i z_i\big)/\|\sum_i z_i\|_2$. The *halt score* of any state is its cosine to the center,

$$s(h) \;=\; \cos(\hat{h}, c) \;=\; \hat{h}^\top c, \qquad \hat{h} = h/\|h\|_2. \tag{1}$$

For terminal states in the validation set, we measure the angles $\theta_i = \arccos(z_i^\top c)$ and set $\tau_p = \cos(\mathrm{Quantile}_{p/100}(\{\theta_i\}))$ for $p \in \{90, 95\}$. The 90th angular percentile gives cosine threshold $\tau_{90} = 0.97824$, and the 95th gives $\tau_{95} = 0.97463$. A larger angular radius corresponds to a smaller cosine threshold, so $\tau_{90}$ defines the tighter cone. We use $\tau_{90}$ as the terminal training target and $\tau_{95}$ as the inference stopping threshold. The center and both thresholds are computed from the SFT model's greedy validation generations, using only its hidden states and generated </think> decisions. This calibration requires no reference answers, correctness labels, or accuracy measurements. The center and thresholds remain frozen throughout latent training and final testing.

### 3.2 Prefix-Replacement Latent Curriculum

We transfer reasoning from text to continuous states gradually, following the prefix-replacement curriculum of Coconut (Hao et al., 2024). At stage $k$, the first $k$ reasoning steps are replaced by $k$ *latent blocks*, while the remaining steps stay explicit:

stage 1: question + <think> $+Z_1+$ step$_2$ $+\cdots+$ </think> + answer
stage $k$: question + <think> $+Z_1+\cdots+Z_k+$ step$_{k+1}$ $+\cdots+$ </think> + answer

Each block spans $c_{\text{thought}} = 2$ continuous positions. At each position, the model receives the previous position's last-layer hidden state as its input, so the latent trajectory evolves recursively without decoding an intermediate token. Samples with fewer than $k$ reasoning steps replace only the steps they contain; longer samples retain an explicit continuation after the latent prefix.

Latent positions are excluded from the language-modeling loss. The </think> \n boundary remains in the input as a forced marker and is also excluded from supervision. The question prefix, remaining explicit steps, answer, and EOS retain the language-modeling loss. We train for 20 epochs, advancing to the next curriculum stage every three epochs. Appendix B describes how the added tokens are initialized from the SFT embeddings.

### 3.3 Auxiliary Step Decoder

Replacing text with hidden states removes the direct token supervision of each reasoning step. We retain a step-level learning signal through an auxiliary decoder in the SIM-CoT style (Wei et al., 2026). For each latent block, the decoder receives its two injected continuous states as a prefix and is trained with cross-entropy to reconstruct the corresponding explicit step and EOS. This objective encourages the block to retain the content of the step it replaces.

We attach the decoder from the first curriculum stage and optimize it jointly with the main model. Step supervision can therefore guide the latent representation as it forms, without waiting for a separate phase to interpret an already-trained latent space. The auxiliary decoder is a Llama-3.2-1B module with a lower learning rate than the main model, intended to limit how quickly it adapts to the evolving latent states. It is used only for training and discarded at inference.

### 3.4 Terminal-Region Loss

Step supervision alone does not preserve a stopping rule. As recursive hidden states replace token embeddings, the model encounters a different input distribution and its hidden geometry can drift. The language-modeling and decoder objectives may improve even as terminal states move away from the frozen region. A region loss constrains these last-layer states at latent block ends.

A sequence has a *terminal* block end only when all of its reasoning steps have been replaced, so that the final latent block immediately precedes </think>. We denote this end's halt score by $s_T$ and the scores of the remaining, *nonterminal* ends by $\{s_j\}$. When explicit steps still follow the latent prefix, every block end is nonterminal. With these definitions, the joint objective is

$$\mathcal{L} = \mathcal{L}_{\mathrm{LM}} \; + \; \lambda_+ \underbrace{\big[\tau_{90} - s_T\big]_+}_{\text{terminal, linear hinge}} \; + \lambda_- \underbrace{\big[s_j - \tau_{95}\big]_+^2}_{\text{nonterminal, squared hinge}} \; + \lambda_{\mathrm{dec}}\, \mathcal{L}_{\mathrm{dec}}, \tag{2}$$

where $[\cdot]_+ = \max(0, \cdot)$ and the weights are $\lambda_+ = 5$, $\lambda_- = 1$, and $\lambda_{\mathrm{dec}} = 0.5$. The terminal term is averaged over sequences with a terminal end. The nonterminal term is first averaged over each sequence's nonterminal ends, then over the batch.

The two region terms serve different purposes. The terminal term pulls the final state into the tighter $\tau_{90}$ cone, while the nonterminal term discourages intermediate ends from entering the wider $\tau_{95}$ cone. Their cosine thresholds differ by $\approx 0.0036$ ($\approx 1^\circ$), placing the terminal training target beyond the inference boundary. Terminal supervision expands with the curriculum: at stage $k$, it applies to problems with at most $k$ reasoning steps. By stage seven, problems with at most seven steps are fully latent, while longer problems still retain explicit steps. The nonterminal term applies from stage one.

**Why Use a Linear Terminal Hinge?** A squared terminal penalty $\lambda_+[\tau_{90} - s_T]_+^2$ has derivative $-2\lambda_+(\tau_{90} - s_T)$ with respect to $s_T$ whenever $s_T < \tau_{90}$. Its pull toward the target becomes weaker as the score approaches the threshold. Gradients from the language-modeling and decoder losses can then counteract this signal before the terminal state reaches its target.

The linear hinge maintains derivative $-\lambda_+$ with respect to the score throughout $s_T < \tau_{90}$. It continues to penalize a small remaining gap to the target; once the score exceeds the threshold, both the penalty and its derivative vanish. This motivates using a linear terminal term to preserve pressure near the boundary. Section 4.6 tests this explanation by changing only the hinge exponent. For nonterminal ends, the squared penalty discourages premature entry into the halting region.

### 3.5 Adaptive-Halt Inference

Inference uses the main model alone. After appending <think>\n to the question, it recursively generates one latent block and evaluates the halt score of the block-end state. If $s(h) \geq \tau_{95}$, it ends the latent phase; otherwise, it generates another block and repeats the check. Every sequence produces at least one block, with a maximum of ten.

After stopping, the model forces the pre-trained </think>\n boundary and greedily decodes the answer. Only answer positions map to vocabulary logits. In batched decoding, later cache positions of halted sequences are masked out. A sequence that never enters the region answers at the ten-block maximum; we report budget fallbacks separately from geometric halts (Section 4.3).

## 4 Experiments

### 4.1 Setup

**Model and Data.** Unless stated otherwise all models use Llama-3.2-1B-Instruct (Meta, 2024b) ; Section 4.5 repeats the entire pipeline on 3B and 8B backbones. We fine-tune the explicit-CoT SFT model and train LatentHalt on GSM8K-Aug (Deng et al., 2024), an augmentation of the GSM8K grade-school math benchmark (Cobbe et al., 2021) with short symbolic reasoning steps. The separate validation set supplies the geometry calibration data (Section 3.1); final test questions do not enter this calibration. Final evaluation follows the SIM-CoT protocol on four math benchmarks: GSM8K-Aug (Deng et al., 2024) (1,319 test problems), GSM-Hard (Gao et al., 2023) (1,319), MultiArith (Roy & Roth, 2015) (180), and SVAMP (Patel et al., 2021) (1,000). We use GSM8K as shorthand for the GSM8K-Aug test split when reporting results. Accuracy is exact numeric match. For CoT SFT and LatentHalt, we extract the answer only from text after the first </think>; outputs without this marker are scored as wrong. Appendix B gives the baseline-specific scoring for Coconut and No-CoT.

Table 1: Accuracy and estimated inference cost on the four-benchmark protocol. * marks out-of-distribution benchmarks; only GSM8K is in-domain. "Avg." is the unweighted mean accuracy across benchmarks. GFLOPs are per question, excluding the shared question prefill; "All" averages over 3,818 problems (Appendix D). Best latent result per column in **bold**; green shading identifies our method. All rows use the same 1B backbone; accuracy is exact numeric match.

| | Accuracy (%) | | | | | GFLOPs / question | |
|---|---|---|---|---|---|---|---|
| Method | GSM8K | GSM-Hard* | MultiArith* | SVAMP* | Avg. | GSM8K | All |
| No-CoT | 29.2 | 6.7 | 59.4 | 38.4 | 33.4 | 3.1 | 3.8 |
| CoT SFT | 56.9 | 13.2 | 97.2 | 61.4 | 57.2 | 76.7 | 73.4 |
| Coconut | 31.0 | 7.9 | 90.6 | 39.8 | 42.3 | 29.0 | 29.4 |
| SIM-CoT | **42.2** | **9.3** | 87.7 | 43.9 | 45.8 | 29.0 | 29.4 |
| ours | 40.0 | 9.0 | **95.0** | **54.4** | **49.6** | **18.3** | **17.6** |

**Training.** We train LATENTHALT jointly from the SFT checkpoint on a single A800 GPU in bf16, using the 20-epoch, seven-stage curriculum of Section 3.2. The decoder loss is normalized per block so that supervision is organized at the level of the reasoning steps being replaced. After warmup, learning rates remain fixed at $10^{-4}$ for the main model and $10^{-5}$ for the decoder. Evaluation uses the same frozen halt threshold and one-to-ten-block inference rule on every benchmark, with no per-dataset tuning. Appendix B gives the full hyperparameters.

**Baselines.** Our comparison covers direct answering, explicit reasoning, and fixed-length latent reasoning (Table 1). *No-CoT* answers without an intermediate rationale, while *CoT SFT* generates the full textual trace and provides the explicit-reasoning accuracy reference. Among latent methods, *Coconut* (Hao et al., 2024) uses six fixed continuous blocks in our 1B run; *Coconut → SIM-CoT* (Wei et al., 2026) adds step-decoder supervision in a second training phase and uses a fixed latent budget. We refer to this chain as SIM-CoT. For models trained on GSM8K-Aug, GSM8K measures in-domain performance, while GSM-Hard, MultiArith, and SVAMP measure transfer.

### 4.2 MAIN RESULTS

Table 1 shows that adaptive latent reasoning remains competitive with SIM-CoT in domain and improves the average across the transfer sets. (1) LATENTHALT reaches $40.0\%$ on GSM8K, within $2.2$ points of SIM-CoT, and is on par on GSM-Hard ($9.0\%$ vs. $9.3\%$). Its largest gains are on MultiArith, where accuracy reaches $95.0\%$ ($+7.3$ points), and SVAMP, where it reaches $54.4\%$ ($+10.5$ points). Together, these results raise the three-benchmark transfer average by $5.8$ points. (2) No-CoT reaches $33.4\%$ on the four-benchmark average at $3.8$ GFLOPs per question. LATENTHALT increases this average to $49.6\%$ at $17.6$ GFLOPs, placing adaptive latent reasoning between direct answering and explicit CoT in computation cost. (3) Coconut reaches $42.3\%$ on the four-benchmark average at $29.4$ GFLOPs per question with its fixed six-block budget. LATENTHALT improves this average by $7.3$ points while using $17.6$ GFLOPs per question. The gains on MultiArith and SVAMP coincide with shorter latent trajectories on these benchmarks, as examined below. This pattern is consistent with the motivation for allowing reasoning length to vary with the problem. Explicit CoT remains the strongest accuracy reference, including $56.9\%$ on GSM8K; the latent methods trade some of that accuracy for substantially shorter reasoning sequences.

### 4.3 ADAPTIVE HALTING AND EFFICIENCY

All $3{,}818$ evaluation problems halt inside the frozen region. The halt rate is $100\%$ on every benchmark, with no maximum-budget fallbacks, so the accuracies in Table 1 are obtained through the geometric stopping rule. The model uses $2.43$ blocks per problem on average, and the block count tracks problem difficulty: GSM8K and GSM-Hard draw the most blocks and SVAMP the fewest, under the same threshold and decoding settings (Table 7). Variation also appears within each benchmark: GSM8K trajectories range from one to seven blocks, with most problems using two or three. Appendix C reports the full statistics and block-count distribution.

**Inference Compute.** Adaptive halting shortens the thinking phase, and the last two columns of Table 1 turn that into FLOPs. On GSM8K, LATENTHALT produces on average 2.79 latent blocks plus the </think> boundary and $\approx 2.2$ answer tokens. The explicit-CoT model instead emits 31.9 rationale and answer tokens. Under a KV-cache-aware estimate of useful decoding FLOPs (Appendix D), and counting only the continuation after the shared question prefill, this is 18.3 versus 76.7 GFLOPs per question: a 76.1% reduction, or $4.2\times$ less reasoning-phase compute. The saving holds across all four benchmarks (68.8–78.8%) and is largest on GSM-Hard, where explicit rationales are longest. Two effects compound: LATENTHALT generates $3.3\times$ fewer positions, and it applies the vocabulary projection only to answer tokens, never to latent blocks. Relative to *fixed-length* latent CoT, the additional saving comes from problems that halt in one or two blocks (most of SVAMP, where LATENTHALT averages 1.48 blocks against a fixed budget of six).

### 4.4 GEOMETRY ANALYSIS

We evaluate the explicit-CoT termination geometry underlying the calibration procedure in Section 3.1. Terminal predictor states form the positive class, and all non-terminal predictor states form the negative class. These measurements concern the SFT model; the following ablations examine how latent training preserves access to its frozen termination region.

Table 2: Geometry of the </think> termination region measured on the validation set with the explicit-CoT SFT model. Separability uses the cosine halt score (Eq. 1) on a held-out, prompt-disjoint partition within this set, with all non-terminal predictor states as negatives.

| Quantity | Value |
|---|---|
| Mean resultant length | 0.9862 |
| Mean pairwise cosine | 0.9727 |
| Radius (50/90/95%) | 8.98°/11.98°/12.93° |
| $\tau_{90}$ / $\tau_{95}$ threshold | 0.97824/0.97463 |
| Held-out AUROC / FPR @ 95% recall | $0.99995/1.5 \times 10^{-4}$ |
| $\cos(c, w_{\texttt{</think>}})$ | 0.4047 (66.1°) |

**The Region Is Concentrated and Separable.** The terminal states point in nearly the same direction: their mean resultant length is 0.986 and mean pairwise cosine is 0.973 (Table 2). Concentration alone would not make the region useful if non-terminal states occupied it too. On a held-out, prompt-disjoint partition within the validation set, the halt score separates the two classes with AUROC 0.99995 and AUPRC 0.99859. At 95% terminal recall, only $1.5 \times 10^{-4}$ of non-terminal states fall inside the region. Together, these measurements show why a cosine comparison against one fixed center can recover the explicit model's stopping decision.

**The Region Is a Residual-Stream Direction, Stable across Domains.** Because every terminal state predicts the same token, one might expect the center to recover the </think> unembedding row $w_{\texttt{</think>}}$; instead the two are 66.1° apart ($\cos(c, w_{\texttt{</think>}}) = 0.405$; Table 2), so the halt score reads a characteristic terminal-state direction in the residual stream rather than the </think> logit magnitude. The region is also stable across the four math domains: fitting on three and detecting explicit-CoT terminations on the fourth gives 94.5% macro recall at a $1.0 \times 10^{-4}$ false-positive rate and AUROC 0.99996 (Appendix A), leaving the deployed region unchanged.

### 4.5 SCALING TO LARGER BACKBONES

We test whether the termination geometry and the halting rule built on it generalize beyond a 1B backbone, rather than being an artifact of small scale. We repeat the *entire* pipeline (explicit-CoT SFT, region fit, joint latent curriculum, adaptive-halt inference) on Llama-3.2-3B-Instruct (Meta, 2024c) and Llama-3.1-8B-Instruct (Meta, 2024a), and re-run the No-CoT, CoT SFT and Coconut baselines at each scale under the same protocol and four benchmarks.[1] With no per-scale tuning, each scale re-fits its own region from its own SFT checkpoint and deploys its own frozen $\tau_{95}$; Appendix E gives the per-scale settings, including the 8B latent block width matched to Coconut.

[1] Coconut's original paper and released code report hyperparameters only for its own backbone, not for 3B or 8B (Hao et al., 2024); we therefore train the 3B and 8B Coconut baselines with the same hyperparameters as LATENTHALT, for a fair comparison.

Table 3: Backbone scaling: the full pipeline is re-run at 3B and 8B with no per-scale tuning, re-fitting the region from each scale's own SFT checkpoint (Appendix E). Columns, the *out-of-distribution marking, and the FLOPs estimator follow Table 1. GFLOPs compare within a scale, not across. Bold marks the best latent result per column within each scale; green shading identifies our method.

| | Accuracy (%) | | | | | GFLOPs / question | |
|---|---|---|---|---|---|---|---|
| Method | GSM8K | GSM-Hard* | MultiArith* | SVAMP* | Avg. | GSM8K | All |
| *Llama-3.2-3B-Instruct* | | | | | | | |
| No-CoT | 37.1 | 9.0 | 90.6 | 52.2 | 47.2 | 8.2 | 10.0 |
| CoT SFT | 73.2 | 21.2 | 99.4 | 75.3 | 67.3 | 201.1 | 193.8 |
| Coconut | 37.8 | 8.6 | **92.2** | 45.5 | 46.0 | 82.4 | 84.0 |
| ours | **40.2** | **9.9** | 90.0 | **61.8** | **50.5** | **54.0** | **51.3** |
| *Llama-3.1-8B-Instruct* | | | | | | | |
| No-CoT | 42.8 | 11.6 | 93.9 | 61.8 | 52.5 | 18.5 | 24.2 |
| CoT SFT | 77.7 | 26.2 | 98.9 | 79.0 | 70.5 | 471.9 | 457.2 |
| Coconut | 16.9 | 4.0 | 42.2 | 21.1 | 21.1 | 202.2 | 203.0 |
| ours | **47.8** | **13.6** | **97.2** | **70.2** | **57.2** | **90.8** | **90.6** |

Table 3 reports the result against the 1B reference of Table 1. LATENTHALT reaches $47.8\%$ GSM8K accuracy at 8B, but the explicit-CoT ceiling also rises: the latent-to-explicit gap grows from $16.9$ points at 1B to $29.9$ at 8B. Against fixed-length Coconut on the same backbone, LATENTHALT leads the four-benchmark average at both 3B and 8B. The primary result is the halting behavior: the halt rate stays $100\%$ with no ten-block fallbacks at any scale, at 2.4–2.7 latent blocks per problem. Because dense parameters and hidden size grow with scale, only the within-scale GFLOPs saving against each scale's own CoT SFT is comparable across rows, and it remains substantial at every scale. The mechanism itself strengthens: the SFT terminal states stay as concentrated as at 1B, and the separability the halting rule relies on improves with scale, the held-out false-positive rate at $95\%$ recall falling to $0$ at 8B. Appendix E gives the per-scale geometry, GFLOPs, and halting statistics.

### 4.6 ABLATIONS

The loss ablations separate the ability to halt from the quality of reasoning performed before halting (Figure 3). We also vary the inference threshold, where no tested value beats the geometric default on the four-benchmark average (Appendix C). Appendix B reports additional ablations on loss normalization, learning-rate scheduling, and curriculum stage length.

**Removing the Terminal-Region Loss ($\lambda_+ = \lambda_- = 0$).** We remove both region terms while retaining the frozen region for evaluation. The resulting model never halts geometrically: all $3{,}818$ problems exhaust the ten-block budget. Its terminal scores average $0.111$ and peak at $0.170$, far below $\tau_{95} = 0.9746$ and the full model's mean terminal score of $0.985$. GSM8K accuracy falls from $40.0\%$ to $8.4\%$. The score gap shows an explicit-CoT initialization does not by itself preserve access to its termination region. Without the geometric constraint, the language-modeling and decoder objectives allow the latent states to move far from the stopping signal we intend to reuse, so preserving that signal requires the region loss during training, not the frozen region alone.

**Shape of the Terminal Hinge (Linear vs. Squared).** Replacing only the terminal hinge with a squared penalty produces a different failure: trajectories approach the halting region but often fail to enter it. For each problem, we record the highest halt score reached anywhere along its rollout. These maxima average $0.9748$, almost at the inference threshold $\tau_{95} = 0.9746$; $89\%$ fall within $0.01$ of the boundary, and the largest is $0.9870$. This diagnostic gives non-halting problems the benefit of their best state across all ten blocks, yet the scores remain concentrated near the boundary. These rollout maxima are distinct from this variant's mean terminal-state score of $0.972$.

The resulting halt rate is $59.4\%$, leaving $40.6\%$ of problems to exhaust the budget, with $5.86$ blocks per problem on average. Estimated cost rises to $34.5$ GFLOPs per question from $17.6$, while GSM8K accuracy falls to $32.6\%$ from $40.0\%$. This behavior is consistent with the gradient argument in

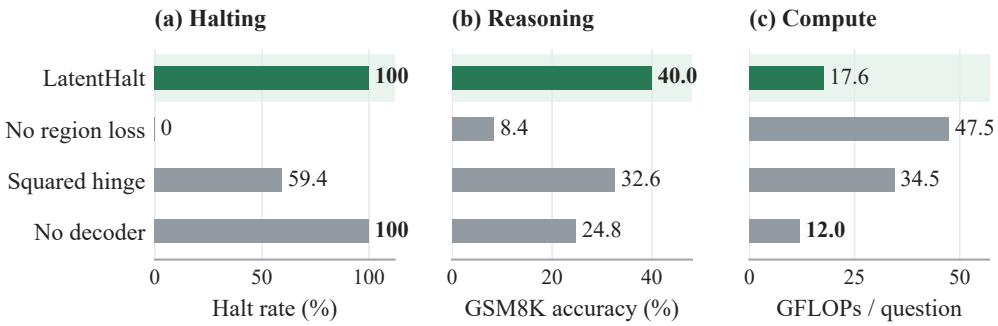



Figure 3: **Loss-component ablations.** The training budget, schedule, and evaluation protocol are held fixed. **(a)** Geometric halt rate. **(b)** GSM8K accuracy. **(c)** Estimated inference GFLOPs per question. Halting and GFLOPs cover all 3,818 problems. Green identifies the full model; bold values mark the highest halt rate and accuracy and the lowest cost, including ties. Accuracies pooled across all problems are 35.7%, 5.6%, 27.2%, and 27.2% in row order.

Section 3.4: a penalty that becomes weak near its target may bring trajectories close to a region without making entry reliable under the joint objective.

**Removing the Auxiliary Step Decoder ($\lambda_{\text{dec}} = 0$).** Removing step supervision leaves the geometric stopping behavior intact: halting remains at 100%, with mean terminal score 0.979. What changes is the trajectory leading to that decision. The model stops after one block on 98.3% of problems, averaging just 1.02 blocks and 12.0 GFLOPs per question. Accuracy falls from 40.0% to 24.8% on GSM8K and from 95.0% to 56.1% on MultiArith, while SVAMP changes little (53.1% vs. 54.4%). A high halt rate alone is therefore insufficient evidence of effective latent reasoning. The region loss makes termination possible, while the decoder helps preserve the intermediate reasoning content that gives the stopping decision its value.

**Training Cost.** Because the decoder is attached from stage one, LATENTHALT trains in a *single* joint run of seven curriculum stages (20 epochs from the SFT checkpoint). Adding step supervision the SIM-CoT way instead takes two sequential runs from the same SFT checkpoint: a full Coconut curriculum and *then* a separate decoder-supervised phase. In its released GSM8K configuration this chain is 21 Coconut epochs followed by 32 decoder epochs, 53 in total. That is about 2.7× more training than LATENTHALT, at comparable in-distribution accuracy and higher out-of-distribution accuracy (Table 1). Folding the two phases into one also removes the representation re-mapping the decoder must otherwise perform on an already-drifted latent space (Section 3.3).

## 5 DISCUSSION AND CONCLUSION

LATENTHALT shows that a model's own end-of-reasoning geometry is precise enough to serve as a halting rule, but several limitations bound the claims. (i) We study backbones of up to 8B parameters (Section 4.5, Appendix E) and math benchmarks with short symbolic rationales. Whether the termination region stays as concentrated well beyond 8B, for longer or free-form rationales, or outside the math and symbolic tasks where CoT is most effective (Sprague et al., 2025), is open. (ii) The region is frozen from an explicit-CoT SFT model, which is what makes halting training-free but also requires such a model and ties halting quality to how cleanly it learned `</think>`.

The decision to stop explicit reasoning leaves a measurable structure in a model's hidden states. LATENTHALT reuses that structure to set the length of latent reasoning per problem: its joint curriculum gives latent blocks step-level supervision, while a linear terminal hinge preserves their access to the frozen termination region. More broadly, an explicit model's decision geometry can provide a starting point for controlling continuous computation, without learning a separate halting head.

### REPRODUCIBILITY STATEMENT

The termination geometry (Section 3.1, Table 2) is fit by a deterministic procedure over greedy SFT generations on the separate validation set. The center and thresholds are frozen before latent training; final test data are excluded from fitting the deployed region and selecting its stopping threshold. Test-split refits in Appendix A and threshold sweeps in Appendix C are post-hoc diagnostics. The exact halt score (Eq. 1), thresholds, training objective (Eq. 2), curriculum, and inference loop (Sections 3.2–3.5) are specified in the main text. Appendices B–D give all hyperparameters, dataset protocols, and the FLOPs-estimation formula. Every reported LATENTHALT number (accuracy, halt rate, block statistics) comes from a single released evaluation summary over 3,818 problems. Code for geometry calibration, joint latent training, adaptive-halt inference, and evaluation is available in the anonymous repository at `https://anonymous.4open.science/r/LatentHalt-EA19/`.

### AI USE STATEMENT

In this work, we used generative AI tools to implement and debug parts of our method, training, and inference code. We have not used generative AI tools to propose or refine our hypotheses, to design or give feedback on our methodology or experiments, or to interpret our results; these are the authors' own. The remaining required-disclosure tasks, including generating synthetic datasets, cleaning or reformatting datasets, developing theoretical models, formulating or proving mathematical claims, assisting with translation, and supporting qualitative or thematic data analysis, are not applicable to this work. Additionally, we used generative AI tools to draft and edit parts of the manuscript, improve its readability, and to identify, source, and format bibliographic references. We have reviewed all AI-assisted work: all AI-assisted code was verified and tested for correctness by the authors, and all AI-drafted text and citations were checked against primary sources. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

### ETHICS STATEMENT

This work studies the training and inference efficiency of latent chain-of-thought reasoning on public grade-school math word-problem benchmarks with a small open-weight model. It involves no new data collection, human subjects, or personally identifiable information, and we foresee no direct ethical concerns beyond those common to language models. Reducing the compute required for reasoning may lower the energy cost of deployment.

## REFERENCES

Andrea Banino, Jan Balaguer, and Charles Blundell. PonderNet: Learning to ponder. *arXiv preprint arXiv:2107.05407*, 2021.

Jeffrey Cheng and Benjamin Van Durme. Compressed chain of thought: Efficient reasoning through dense representations. *arXiv preprint arXiv:2412.13171*, 2024.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Yuntian Deng, Kiran Prasad, Roland Fernandez, Paul Smolensky, Vishrav Chaudhary, and Stuart Shieber. Implicit chain of thought reasoning via knowledge distillation. *arXiv preprint arXiv:2311.01460*, 2023.

Yuntian Deng, Yejin Choi, and Stuart Shieber. From explicit CoT to implicit CoT: Learning to internalize CoT step by step. *arXiv preprint arXiv:2405.14838*, 2024.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. PAL: Program-aided language models. In *International conference on machine learning*, pp. 10764–10799. PMLR, 2023.

Sachin Goyal, Ziwei Ji, Ankit Singh Rawat, Aditya Krishna Menon, Sanjiv Kumar, and Vaishnavh Nagarajan. Think before you speak: Training language models with pause tokens. In *International Conference on Learning Representations*, volume 2024, pp. 27896–27923, 2024.

Alex Graves. Adaptive computation time for recurrent neural networks. *arXiv preprint arXiv:1603.08983*, 2016.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. *arXiv preprint arXiv:2412.06769*, 2024.

Colin Hong, Xu Guo, Anand Chaanan Singh, Esha Choukse, and Dmitrii Ustiugov. SLIM-SC: Thought pruning for efficient scaling with self-consistency. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 34488–34505, 2025.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213, 2022.

Zirui Li, Xuefeng Bai, Kehai Chen, Yizhi Li, Jian Yang, Chenghua Lin, and Min Zhang. Dynamics within latent chain-of-thought: An empirical study of causal structure. *arXiv preprint arXiv:2602.08783*, 2026.

Xin Liu and Lu Wang. Answer convergence as a signal for early stopping in reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 17907–17918, 2025.

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

Meta. Llama 3.1 8B. `https://huggingface.co/meta-llama/Llama-3.1-8B`, 2024a. Hugging Face model card, accessed 2026-09-05.

Meta. Llama 3.2 1B. `https://huggingface.co/meta-llama/Llama-3.2-1B`, 2024b. Hugging Face model card, accessed 2026-04-22.

Meta. Llama 3.2 3B. `https://huggingface.co/meta-llama/Llama-3.2-3B`, 2024c. Hugging Face model card, accessed 2026-09-05.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori B Hashimoto. s1: Simple test-time scaling. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 20286–20332, 2025.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 conference of the North American chapter of the association for computational linguistics: human language technologies*, pp. 2080–2094, 2021.

Jacob Pfau, William Merrill, and Samuel R Bowman. Let's think dot by dot: Hidden computation in transformer language models. *arXiv preprint arXiv:2404.15758*, 2024.

Subhro Roy and Dan Roth. Solving general arithmetic word problems. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, pp. 1743–1752, 2015.

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

Zhenyi Shen, Hanqi Yan, Linhai Zhang, Zhanghao Hu, Yali Du, and Yulan He. CODI: Compressing chain-of-thought into continuous space via self-distillation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 677–693, 2025.

Dachuan Shi, Abedelkadir Asi, Keying Li, Xiangchi Yuan, Leyan Pan, Wenke Lee, and Wen Xiao. SwiReasoning: Switch-thinking in latent and explicit for Pareto-superior reasoning LLMs. In *International Conference on Learning Representations*, volume 2026, pp. 137060–137093, 2026.

Zayne Sprague, Fangcong Yin, Juan Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. To CoT or not to CoT? Chain-of-thought helps mainly on math and symbolic reasoning. In *International Conference on Learning Representations*, volume 2025, pp. 94118–94162, 2025.

DiJia Su, Hanlin Zhu, Yingchen Xu, Jiantao Jiao, Yuandong Tian, and Qinqing Zheng. Token assorted: Mixing latent and text tokens for improved language model reasoning. *arXiv preprint arXiv:2502.03275*, 2025.

Renliang Sun, Wei Cheng, Dawei Li, Haifeng Chen, and Wei Wang. Stop when enough: Adaptive early-stopping for chain-of-thought reasoning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 27250–27268, 2026.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

Xilin Wei, Xiaoran Liu, Yuhang Zang, Xiaoyi Dong, Yuhang Cao, Jiaqi Wang, Xipeng Qiu, and Dahua Lin. SIM-CoT: Supervised implicit chain-of-thought. In *International Conference on Learning Representations*, volume 2026, pp. 56721–56742, 2026.

Heming Xia, Chak Tou Leong, Wenjie Wang, Yongqi Li, and Wenjie Li. TokenSkip: Controllable chain-of-thought compression in LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 3351–3363, 2025.

Yige Xu, Xu Guo, Zhiwei Zeng, and Chunyan Miao. 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)*, pp. 23336–23351, 2025.

Zhen Zhang, Xuehai He, Weixiang Yan, Ao Shen, Chenyang Zhao, and Xin Wang. Soft Thinking: Unlocking the reasoning potential of LLMs in continuous concept space. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 168990–169012. Curran Associates, Inc., 2025. doi: 10.52202/085713-5629. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/f7396d1c54d51416958d63e285377103-Paper-Conference.pdf.

# A TERMINATION-REGION TRANSFER AND STABILITY

This appendix tests the stability and cross-domain transfer of the label-free fitting procedure in Section 3.1. All region refits and held-out detection measurements here use the `test` split for post-hoc analysis. The resulting centers and quantile thresholds are diagnostic quantities; they do not replace the deployed region calibrated on the separate validation set before latent training. We extract states by teacher-forced replay of greedy SFT trajectories on these test questions, using the state definition in Section 3.1. The positive label is the model's own generated `</think>` token, so these measurements require no reference answers or correctness annotations. The acceptance criteria were fixed before the run: minimum cross-domain center cosine $\geq 0.95$, leave-one-domain-out macro recall $\geq 90\%$, and macro false-positive rate $\leq 1\%$. All three hold.

**The Center Direction Is a Property of the Model, Not of the Fitting Set.** Table 4 fits a separate region on each domain's `test` questions alone. Let $c_{\text{test}}$ denote the center refit on pooled test trajectories from all four domains. The four domain centers agree with $c_{\text{test}}$ to cosine $0.9963$ or better and with each other to at least $0.9924$. Resampling half of a domain's terminal states $200$ times moves its center by at most cosine $0.9997$ (5th percentile). Concentration is likewise a domain-independent property: the mean resultant length is between $0.985$ and $0.991$ everywhere. What does vary is the *radius*: the per-domain $\tau_{95}$ threshold spans $0.9726$ to $0.9858$, with the tightest cone on MultiArith, whose $180$ terminal states are the fewest and come from the shortest rationales. For this analysis, the region refit on pooled test terminal states from all four domains serves as the in-fit reference.

Table 4: Diagnostic termination-region refits per domain on the `test` split with the explicit-CoT SFT model. The pooled $c_{\text{test}}$ is the reference center for this analysis. Split-half (q05) is the 5th percentile of center cosines from random halves of each domain's terminal states over 200 repeats.

| Fitting domain | Terminal states | MRL | Domain $\tau_{95}$ | cos to $c_{\text{test}}$ | Split-half (q05) |
|---|---|---|---|---|---|
| GSM8K | 1,319 | 0.9853 | 0.9726 | 0.99957 | 0.99993 |
| GSM-Hard | 1,318 | 0.9874 | 0.9759 | 0.99910 | 0.99992 |
| MultiArith | 180 | 0.9910 | 0.9858 | 0.99628 | 0.99968 |
| SVAMP | 1,000 | 0.9884 | 0.9772 | 0.99916 | 0.99992 |

**A Region Fit without a Domain Still Detects That Domain's Terminations.** Table 5 withholds one domain's `test` questions, fits the center and the $\tau_{95}$ radius on the pooled terminal states of the other three domains, and applies the result to the held-out domain with no tuning. Held-out recall is $89.9$–$99.4\%$ at a false-positive rate of order $10^{-4}$, for a macro recall of $94.5\%$. The same quantities for the pooled test-region refit are $93.9$–$99.4\%$ and $96.0\%$. Withholding a domain from the fit therefore costs $1.5$ points of recall on it, and costs nothing in false positives. This diagnostic shows that the region detects terminations in a math domain excluded from its fit almost as well as in a domain included in it.

Table 5: Diagnostic explicit-state termination detection on the `test` split: a region refit on the other three domains (left) and a pooled test-region refit on all four (right, in-fit reference). Recall and FPR are the fractions of the target's terminal and non-terminal states inside the respective $\tau_{95}$ regions. "Macro" averages benchmarks equally.

| | Fit on the other three | | | Pooled test refit (in-fit) | | |
|---|---|---|---|---|---|---|
| Target | Recall | FPR ($10^{-4}$) | AUROC | Recall | FPR ($10^{-4}$) | AUROC |
| GSM8K | 89.9% | 1.5 | 0.99993 | 94.1% | 1.2 | 0.99996 |
| GSM-Hard | 92.3% | 1.3 | 0.99993 | 93.9% | 2.1 | 0.99993 |
| MultiArith | 99.4% | 0 | 1.00000 | 99.4% | 0 | 1.00000 |
| SVAMP | 96.2% | 1.2 | 0.99997 | 96.7% | 1.2 | 0.99997 |
| Macro | 94.5% | 1.0 | 0.99996 | 96.0% | 1.1 | 0.99997 |

## B HYPERPARAMETERS AND TRAINING PROTOCOL

LATENTHALT is trained from the explicit-CoT SFT checkpoint of Llama-3.2-1B-Instruct (Meta, 2024b) .

**Data.** Training uses GSM8K-Aug (Deng et al., 2024) (385,610 training examples). Geometry calibration uses a separate validation set (Section 3.1). Final evaluation uses the GSM8K-Aug test split (Deng et al., 2024) (1,319), GSM-Hard (Gao et al., 2023) (1,319), MultiArith (Roy & Roth, 2015) (180), and SVAMP (Patel et al., 2021) (1,000; the protocol merges the 700 train and 300 test items).

**Baseline Scoring.** Coconut uses its original generation procedure with <|start-latent|> and <|end-latent|> boundaries; we score the last numeric value in its output by exact decimal equality. For No-CoT, we match the generated answer directly.

**Curriculum.** Prefix replacement with $c_{\text{thought}} = 2$ latent positions per logical block, starting at stage 1 and advancing to the next stage every three epochs. Samples with fewer steps than the current stage replace only their available steps; longer samples retain their remaining explicit steps.

**Token Initialization.** The <|latent|> token is added for book-keeping and initialized from the embedding of <<. Continuous positions receive recursive hidden states rather than its static embedding. The <think> and </think> boundary tokens reuse their SFT embeddings.

**Optimization.** Single NVIDIA A800 GPU, bf16 (tf32 enabled), effective batch size 64 (32 per device $\times$ 2 gradient-accumulation steps). AdamW, weight decay 0.01, gradient clipping 2.0, warmup ratio 0.03. After warmup, the main model uses a fixed learning rate of $10^{-4}$ and the auxiliary decoder $10^{-5}$ for the remainder of training. Training runs for 20 epochs from the SFT checkpoint, covering seven curriculum stages.

**Losses.** The total objective (Eq. 2) sums the language-model loss, a terminal linear hinge (weight $\lambda_+ = 5$) pulling each sequence's final block-end state to cosine $\geq \tau_{90} = 0.97824$, a nonterminal squared hinge (weight $\lambda_- = 1$) keeping intermediate block ends at cosine $\leq \tau_{95} = 0.97463$, and a step-decoder cross-entropy (weight $\lambda_{\text{dec}} = 0.5$, normalized per block). The auxiliary decoder is a separate Llama-3.2-1B module used only to reconstruct the explicit step for each latent block and is discarded at inference.

**Inference.** A minimum of one and a maximum of ten latent blocks; halt when a block-end halt score reaches $\tau_{95}$; force the pre-trained </think> boundary; greedily decode at most 64 answer tokens.

**Training Settings.** Table 6 varies one setting at a time to examine how the joint curriculum depends on its optimization choices. Replacing per-block decoder-loss normalization with per-token normalization lowers GSM8K accuracy by 3–7 points. Replacing the fixed learning rates with decay reduces accuracy to approximately 32%, an 8-point drop. Shortening the stage interval from three epochs to two costs approximately 10 points. Under this training recipe, the results favor supervision normalized by block, fixed learning rates after warmup, and more time at each curriculum stage.

Table 6: Training-choice ablations, changing only the listed setting. Differences are GSM8K accuracy points relative to the 40.0% main result. The normalization comparison matches training steps and keeps $\lambda_{\text{dec}} = 0.5$. LR denotes learning rate.

| Variant | Changed setting | $\Delta$ GSM8K (pp) |
|---|---|---|
| LATENTHALT | Main-experiment settings | 0 |
| Token-normalized decoder | Per-block $\rightarrow$ per-token loss | $-3$ to $-7$ |
| Learning-rate decay | Fixed $\rightarrow$ decaying LR | $\approx -8$ |
| Shorter curriculum stages | 3 $\rightarrow$ 2 epochs per stage | $\approx -10$ |

## C Adaptive Halting Details

Table 7 and Figure 4 report per-benchmark halting statistics and GSM8K block counts on the `test` split. This appendix also gives the full inference-threshold sensitivity analysis summarized in Section 4.6.

**Threshold Sweep Setup.** The deployed threshold is the cosine of the 95th percentile of SFT terminal-state angles on the validation set. To assess this choice, we repeat inference on the `test` split with the trained model over 112 thresholds: $[0, 1]$ in steps of $0.01$, the two frozen geometric values, and a refinement grid over $[0.981, 0.989]$. Figure 5 shows how these choices change the width of the halting cone and the resulting accuracy–compute trade-off, and Table 8 lists ten representative settings. This sweep is a post-hoc sensitivity analysis of the trained model; the deployed threshold remains the $\tau_{95}$ calibrated on the separate validation set before latent training, and test-set results do not enter its selection.

**GSM8K Tuning versus Transfer.** On GSM8K alone, a tighter threshold does improve accuracy: $0.982$ reaches $41.4\%$, $1.4$ points above the default $\tau_{95}$, at $6\%$ more compute. The gain comes at the expense of transfer accuracy, however, reducing the four-benchmark average from $49.6$ to $49.3$. Across the refinement grid, which we evaluate on all four benchmarks, the best average is $49.4$ at $0.981$. Thus, the alternatives tested across all four benchmarks do not improve on the geometric default, even though tuning for GSM8K alone would suggest a different threshold.

**Behavior Away from the Default.** The broader sweep explains why tighter stopping rules need not improve reasoning. A looser threshold ends trajectories earlier, reducing both compute and accuracy; a sufficiently tight threshold causes problems to miss the region and exhaust the maximum budget. At the fixed-budget endpoints, one block gives $18.1\%$ GSM8K accuracy at $11.3$ GFLOPs, while ten blocks give $9.8\%$ at $46.9$, compared with $40.0\%$ at $18.3$ for adaptive halting. More latent computation is therefore not uniformly beneficial for this model. At threshold $0.982$, SVAMP accuracy falls to $52.6$ from $54.4$, MultiArith to $94.4$ from $95.0$, and GSM-Hard to $8.7$ from $9.0$. Away from the GSM8K peak, looser thresholds reduce accuracy and cost together: $0.95$ gives $35.2\%$ at $17.1$ GFLOPs and $0.90$ gives $28.9\%$ at $15.5$. Below $\approx 0.57$, every problem halts after one block. The first fallbacks appear at $0.983$; at $0.987$, halting falls to $91.0\%$ and accuracy to $35.6\%$ with $33\%$ more compute than the default. At $1.00$, all problems exhaust the ten-block budget.

Table 7: Adaptive halting statistics at the frozen $\tau_{95}$ threshold. "Answer tokens" counts greedily decoded answer tokens after the forced `</think>` boundary.

| Dataset | # problems | Halt rate | Avg. blocks | Block range | Avg. answer tokens |
|---|---|---|---|---|---|
| GSM8K | 1,319 | 100% | 2.79 | 1–7 | 2.20 |
| GSM-Hard | 1,319 | 100% | 2.83 | 1–6 | 3.08 |
| MultiArith | 180 | 100% | 2.01 | 1–3 | 2.01 |
| SVAMP | 1,000 | 100% | 1.48 | 1–3 | 2.12 |
| All | 3,818 | 100% | 2.43 | 1–7 | 2.47 |

## D Inference FLOPs Estimation

We estimate *useful* decoding FLOPs with a KV cache; this is an analytical estimate of the dominant matrix operations, not a hardware measurement, and excludes padding, normalization, activation functions, sampling, and kernel overhead. For a prompt of $P$ tokens and $T > 0$ generated tokens, with dense (non-embedding) parameter count $D$, $L$ transformer layers, hidden size $H$, vocabulary size $V$, and query-projection size $Q$:

$$\mathrm{FLOPs}_{\mathrm{dense}} = 2\,D\,(P + T - 1), \tag{3}$$

$$\mathrm{FLOPs}_{\mathrm{attn}} = 4\,L\,Q\left(P^2 + (T-1)P + \tfrac{T(T-1)}{2}\right), \tag{4}$$

$$\mathrm{FLOPs}_{\mathrm{head}} = 2\,H\,V\,T, \tag{5}$$

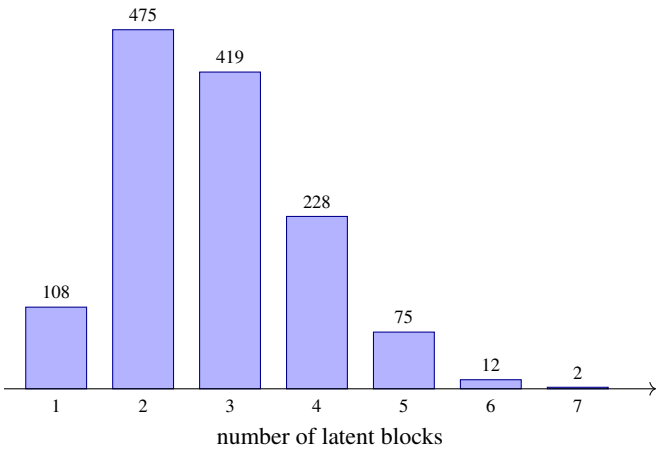



Figure 4: Latent blocks per GSM8K problem at $\tau_{95}$. Labels give problem counts.

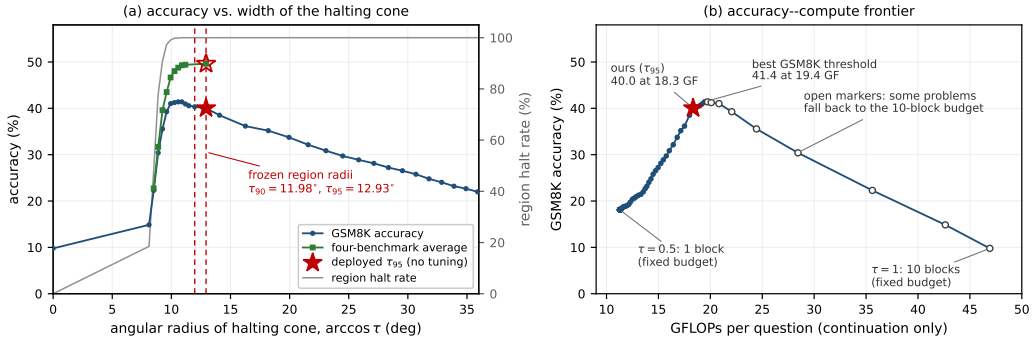



Figure 5: Post-hoc inference-threshold sensitivity on the `test` split; the deployed $\tau_{95}$ is fixed on the validation set. **(a)** GSM8K accuracy and halt rate versus the cone's angular radius $\arccos \tau$; the green curve is the four-benchmark mean accuracy over $[0.981, 0.989]$, the hollow star marks its value at $\tau_{95}$, and dashed lines mark the frozen SFT radii. **(b)** GSM8K accuracy versus GFLOPs per question; open markers have ten-block fallbacks, and $\tau = 0.5, 1$ give fixed one- and ten-block controls.

and the total is their sum. Prefill processes the prompt once; producing $T$ tokens then requires $T - 1$ cached decode forwards (the final sampled token is not fed back). For LATENTHALT the generated positions are the latent block positions, the forced `</think>` boundary, and the answer tokens; for explicit CoT they are the full textual rationale and the answer. Because both methods receive the same question, we report the *continuation* only, i.e., the expression above with the prefill terms $2\,D\,P$ and $4\,L\,Q\,P^2$ removed. The prompt still contributes to the KV-cache context of every subsequent forward. For LATENTHALT the vocabulary projection is charged on answer tokens alone, since latent and boundary positions are never mapped to logits. Table 9 gives per-benchmark GFLOPs for No-CoT, Coconut, and SIM-CoT at 1B, and for CoT SFT and LATENTHALT at 1B and 3B. The 1B No-CoT summary and the 3B and 8B baseline summaries in Table 3 also exclude the first token's vocabulary projection, which is performed during question prefill.

Coconut estimates cover twelve latent positions, one `<|end-latent|>` boundary, and answer decoding, including terminal EOS. We count vocabulary projections only for sampled answer tokens and assume cached decoding, excluding repeated prefix forwards and unused projections in the original implementation.

Table 8: Post-hoc inference-threshold sweep on the GSM8K `test` split with the trained LATENTHALT model: ten representative settings of the 112 thresholds plotted in Figure 5. The deployed threshold is fixed from the separate validation set. GFLOPs are per question; fallbacks reach the ten-block maximum without entering the region. Green shading marks the deployed threshold; bold marks the highest GSM8K accuracy among these diagnostic settings.

| Halt threshold | Accuracy | Halt rate | Avg. blocks | Fallbacks | GFLOPs |
|---|---|---|---|---|---|
| 0.50 (halts after one block) | 18.1 | 100% | 1.00 | 0% | 11.3 |
| 0.95 | 35.2 | 100% | 2.50 | 0% | 17.1 |
| $\tau_{95} = 0.9746$ (ours) | 40.0 | 100% | 2.80 | 0% | 18.3 |
| $\tau_{90} = 0.9782$ | 40.3 | 100% | 2.90 | 0% | 18.7 |
| 0.982 (best on GSM8K) | **41.4** | 100% | 3.07 | 0% | 19.4 |
| 0.985 | 41.0 | 99.1% | 3.42 | 0.9% | 20.8 |
| 0.987 | 35.6 | 91.0% | 4.34 | 9.0% | 24.4 |
| 0.989 | 22.3 | 48.4% | 7.13 | 51.6% | 35.6 |
| 0.99 | 14.9 | 18.5% | 8.91 | 81.5% | 42.6 |
| 1.00 (never halts) | 9.8 | 0% | 10.00 | 100% | 46.9 |

Table 9: Estimated inference FLOPs per question (GFLOPs; lower is better) by benchmark at 1B and 3B. "All" is the per-question mean over all 3,818 problems; reductions compare LATENTHALT with CoT SFT within each backbone.

| Method | GSM8K | GSM-Hard | MultiArith | SVAMP | All |
|---|---|---|---|---|---|
| **Llama-3.2-1B-Instruct** | | | | | |
| No-CoT | 3.1 | 5.3 | 2.6 | 2.8 | 3.8 |
| CoT SFT | 76.7 | 97.7 | 48.2 | 41.6 | 73.4 |
| Coconut | 29.0 | 30.4 | 28.4 | 28.7 | 29.4 |
| SIM-CoT | 29.0 | 30.4 | 28.4 | 28.7 | 29.4 |
| LATENTHALT (ours, adaptive) | 18.3 | 20.7 | 14.8 | 13.0 | 17.6 |
| Reduction | 76.1% | 78.8% | 69.3% | 68.8% | 76.1% |
| **Llama-3.2-3B-Instruct** | | | | | |
| CoT SFT | 201.1 | 261.3 | 124.2 | 107.7 | 193.8 |
| LATENTHALT (ours, adaptive) | 54.0 | 62.0 | 41.5 | 35.5 | 51.3 |
| Reduction | 73.1% | 76.3% | 66.6% | 67.0% | 73.5% |

## E BACKBONE SCALING: PER-SCALE GEOMETRY AND HALTING

This appendix supports the backbone-scaling study of Section 4.5. The termination region is a direction in one model's hidden space and cannot be carried across backbones, so for each scale we re-run the explicit-CoT SFT stage of Section 3.1 and re-fit the center $c$ and the radii $\tau_{90}, \tau_{95}$ from that checkpoint's own greedy generations, by the identical procedure and on the same validation questions, before any latent training begins. Everything else follows Appendix B: the same GSM8K-Aug training set, the same reasoning protocol (<think> and </think> added as two special tokens), the same 20-epoch prefix-replacement curriculum across seven stages, the same loss weights $\lambda_+ = 5$, $\lambda_- = 1$, $\lambda_{\text{dec}} = 0.5$, and the same inference loop with a one-block minimum, a ten-block maximum and that scale's own frozen $\tau_{95}$ as the halt threshold (0.97463 at 1B, 0.95169 at 3B and 0.95394 at 8B; Table 10). The single per-scale difference is the number of continuous positions per latent block: $c_{\text{thought}} = 2$ at 1B and 3B as in the main results, and $c_{\text{thought}} = 1$ at 8B, matching the configuration Coconut uses in its own released setup (Hao et al., 2024) so that the 8B comparison against Coconut is at a matched per-block cost. Block counts are therefore comparable within a scale but not between the 8B rows and the others.

Table 10: Termination geometry of the 3B and 8B explicit-CoT SFT models, measured exactly as in Table 2 and on the same validation questions, with the region fit separately per scale. Rows below the separator measure the *trained* latent model; terminal halt scores are measured at inference on the test split. The 1B reference values appear in Table 2 and Section 4.6.

| Quantity | Llama-3.2-3B | Llama-3.1-8B |
|---|---|---|
| Mean resultant length | 0.9777 | 0.9785 |
| Mean pairwise cosine | 0.9558 | 0.9575 |
| Angular radius 50/90/95% (deg.) | 11.01/15.77/17.88 | 10.89/14.96/17.46 |
| $\tau_{90}$ / $\tau_{95}$ threshold | 0.96238/0.95169 | 0.96612/0.95394 |
| Held-out AUROC | 0.99998 | 0.99999 |
| FPR @ 95% recall | $9 \times 10^{-6}$ | 0 |
| Cosine of $c$ to </think> unembedding (SFT) | 0.4366 | 0.3125 |
| Cosine of $c$ to </think> unembedding (trained) | 0.1909 | −0.3396 |
| Mean terminal halt score (test) | 0.9722 | 0.9885 |

Table 10 extends the SFT geometry measurements of Table 2 to 3B and 8B. It also reports the trained latent model's unembedding alignment and mean terminal halt score on the test split. At 3B, the cosine between the frozen center and the </think> unembedding falls from 0.4366 in the SFT model to 0.1909 after latent training, while the mean terminal halt score is 0.9722 (standard deviation 0.0063). Only states that predict </think> under greedy decoding contribute to each region fit; all other predictor states form the negative class.

Table 11: Per-benchmark adaptive halting for LATENTHALT at the three backbone scales, in the format of Table 7. "Halt rate" and "Fallbacks" are over all 3,818 problems; fallbacks are the problems that never enter the region and spend the ten-block maximum. The Llama-3.2-1B row is Table 7.

| | Avg. latent blocks | | | | | All problems | |
|---|---|---|---|---|---|---|---|
| Backbone | GSM8K | GSM-Hard | MultiArith | SVAMP | All | Halt rate | Fallbacks |
| Llama-3.2-1B | 2.79 | 2.83 | 2.01 | 1.48 | 2.43 | 100% | 0% |
| Llama-3.2-3B | 3.00 | 3.03 | 2.01 | 1.42 | 2.55 | 100% | 0% |
| Llama-3.1-8B | 3.11 | 3.18 | 2.03 | 1.50 | 2.66 | 100% | 0% |

Table 11 gives the block counts and halt rates the scaling paragraph quotes, per benchmark and per scale, in the format of Table 7. At 3B, all 3,818 problems halt geometrically, with no budget fallbacks on any benchmark. Trajectories use one to seven blocks on GSM8K and GSM-Hard and one to three on MultiArith and SVAMP, under the same frozen threshold $\tau_{95} = 0.95169$.