Flash Point: Detecting Answer Commitment in Reasoning Models

Anonymous authors
Paper under double-blind review

Abstract

Reasoning LLMs demonstrate a great deal of power through test-time scaling of the length of their generation. However, they often continue reasoning long after they have already effectively committed to an answer. We show that such over-reasoning is not only computationally wasteful but can also reduce accuracy. Across model families ranging from 3B to 20B parameters, we find that commitment to a final answer trajectory is marked by correlated stabilization of attention entropy and feed-forward activations across layers. Building on this observation, we introduce the *Flash Point Probe*, a lightweight causal classifier that detects this commitment point and enables dynamic early stopping. Across scientific reasoning and knowledge-retrieval benchmarks, the probe matches or exceeds full-length reasoning accuracy in most settings. Early stopping improves scientific reasoning accuracy by an average of +18 percentage points across three model families, while reducing reasoning tokens by up to 93% on knowledge-retrieval tasks. Overall, Flash Point reduces token usage by 28% on in-distribution tasks while preserving or improving accuracy, enabling more efficient adaptive reasoning.

1 Introduction

Reasoning-capable LLMs deliberate through long chains of intermediate tokens before committing to a final answer, and this deliberation drives much of their accuracy on complex tasks (Bakouch et al., 2025; OpenAI, 2025). Yet these models have no native stopping criterion: once committed to an answer, they often keep generating, adding computational overhead that is at best wasted and at worst actively erodes the answer's correctness.

Latent-reasoning approaches compress this deliberation into a handful of latent tokens instead of explicit text, but in leading systems such as CODI (Shen et al., 2025) and COCONUT (Hao et al., 2024) the number of latent tokens is fixed at training time and applied uniformly, irrespective of question difficulty or model capacity. Richer access to internal representations does not buy adaptivity: computation still runs to a preset length regardless of the instance. Instantiating these methods is costly by construction, too, requiring architectural changes and a rebuilt supervised fine-tuning (SFT) pipeline before any stopping behavior can even be learned.

We re-frame the stopping decision as a signal-detection problem over the model's internal information-gain dynamics. Without altering the target model's architecture or weights, we propose the ***"Flash Point Probe"***: a lightweight causal sequence classifier that identifies the *"flash point"*, the earliest reasoning step at which the model has already accumulated enough signal to answer correctly, so that everything generated afterward is pure overhead. Unlike latent reasoning methods constrained to a fixed number of latent tokens, our approach probes the model's internal signals to adjust the number of reasoning tokens to both prompt difficulty and model capability. The probe's decision threshold transfers across tasks with as few as 50 calibration samples.

The probe monitors multi-layer activation snapshots at regular intervals to detect when the model has effectively converged on an answer trajectory. Rather than relying on surface text, it operates on a rich internal representation that combines structural shifts in attention, feed-

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.

forward hidden states, and temporal entropy dynamics reflecting the stabilization of internal uncertainty. These complementary signals are streamed through a lightweight recurrent network, which identifies the point at which additional reasoning is unlikely to provide further benefit.

Why Probing over Fine-Tuning? Existing methods control reasoning chain length through costly post-training paradigms (Aggarwal & Welleck, 2025; Arora & Zanette, 2025; Kimi Team, 2025; Muennighoff et al., 2025). Budget-aware reinforcement learning (RL) demands expensive on-policy rollouts, massive memory overhead, and full retraining for new domain or budget (Han et al., 2025). Meanwhile, test-time scaling risks unbounded inference latency without guaranteeing an optimal halt. Additionally, fine-tuning puts existing capabilities at risk: weight updates can catastrophically forget pretrained knowledge and skills (McCloskey & Cohen, 1989; Kirkpatrick et al., 2017; Luo et al., 2023), distort pretrained features in ways that harm out-of-distribution robustness (Kumar et al., 2022), and degrade safety alignment even when the tuning data is benign (Qi et al., 2024).

A probe, in contrast, trains entirely on cached activations from the frozen base model (Alain & Bengio, 2016; Belinkov, 2022), needing only a single forward pass over full-length rollouts. Furthermore, keeping the backbone frozen removes any post-training risk entirely: outputs are drawn from the unmodified model conditioned on a truncated reasoning prefix, so quality can degrade only by stopping too early, collapsing validation to one measurable quantity: the accuracy cost of early exits.

Training the probe: To train the probe, we derive ground-truth labels from completed reasoning trajectories, defining the optimal flash point as the earliest step at which the model has secured task accuracy and its attention dynamics have stabilized. The auto-regressive architecture naturally captures these cross-step temporal dynamics, enabling the probe to identify when the model has reached a stable answer trajectory.

Optimization is guided by a joint *Flash Point Loss* designed for accurate and timely stopping, thereby reducing the number of reasoning tokens generated without sacrificing accuracy. The objective combines binary cross-entropy supervision, which trains the probe to correctly recognize whether a given reasoning step is the true stopping point, with auxiliary penalties that discourage premature stopping before sufficient reasoning evidence has emerged, continued reasoning after internal information gain has stagnated, and over-generation once the entropy trajectory has stabilized. Together, these objectives encourage the probe to identify the earliest reliable stopping point for CoT reasoning, while preserving task correctness.

Results at glance: Evaluations on GPQA-main (Rein et al., 2023), MMLU-Pro (Wang et al., 2024), and GSM8K (Cobbe et al., 2021) show that the Flash Point Probe reliably locates the reasoning termination point across three model families spanning 3B–20B parameters, and that stopping there does not merely preserve accuracy, it improves it. Against the full chain-of-thought baseline, the probe lifts in-distribution GPQA-main accuracy by an average of +28%, removes up to 93% of reasoning tokens while exactly matching full-CoT accuracy, and still cuts in-distribution token usage by 18% on average at the operating point where accuracy is highest. On graduate-level science questions, full-length generation reaches only 38–55% accuracy, while halting at the detected commitment point raises it to 71–80%: extended reasoning is not merely wasteful, it is actively degrading correctness. This probing architecture and the findings on the novel feature of prompt-attention entropy comprise the core of our contributions. By matching inference-time compute to query difficulty at runtime, the Flash Point Probe turns adaptive compute budgeting into a practical, drop-in capability for reasoning-capable LLMs rather than a training-time design choice.

2 Related Work

Reasoning (Wei et al., 2022) has gained significant importance across all major LLM families, primarily because it delivers substantial performance improvements under a variety of downstream tasks. Reasoning improves model performance through two mechanisms: first, by enriching the quality of the input context, and second, by supplying additional informa-

tion that steers the model toward the correct answer. In both cases, the computational cost is high if this expensive process is not regulated based on necessity. A fundamental challenge in optimizing reasoning is that it is difficult to determine from the question alone how much computation is actually needed to produce the correct answer (Wang et al., 2023). While research has shown that reasoning consistently improves performance on standard benchmarks, this prompts a more precise question: do models truly require every reasoning token, or only a subset? Identifying the optimal stopping point is an active area of research, as unnecessary reasoning tokens introduce avoidable computational expense.

A second challenge compounds this efficiency problem: existing methods that attempt to reduce token overhead come with steep deployment costs. For instance, several works exploring latent reasoning such as CODI (Shen et al., 2025), COCONUT (Hao et al., 2024), and related approaches have reduced the number of explicit reasoning tokens without sacrificing answer quality. However, these methods require custom decoding and fine-tuning, limiting their applicability.

A complementary line of work takes a probing approach (Wang et al., 2025; Xia et al., 2025; Zhang et al., 2025) to bypass some of these limitations: rather than modifying the model, these methods listen to internal signals from the model and use them to determine when to stop reasoning. Our work follows this latter approach, and our method achieves strong results for reasons we outline below. Our approach is model-agnostic and focuses on measurable internal dynamics within the transformer architecture. Two parameter spaces account for roughly 95% of model computation: the attention space and the feed-forward layer space. We argue that tapping into the dynamics of these spaces is the key to detecting when the model has "settled" on a reliable internal representation. The statistical measure we use to quantify this settling behavior is entropy. By layering multiple entropy-based signals across these spaces, we construct a lightweight probing model that silently monitors the model's internal state during reasoning. When the entropy signals indicate convergence, we treat this as a stopping signal and prompt the model to generate its final answer terminating further reasoning token generation. Our results demonstrate the effectiveness of this approach, as detailed below.

3 Methodology

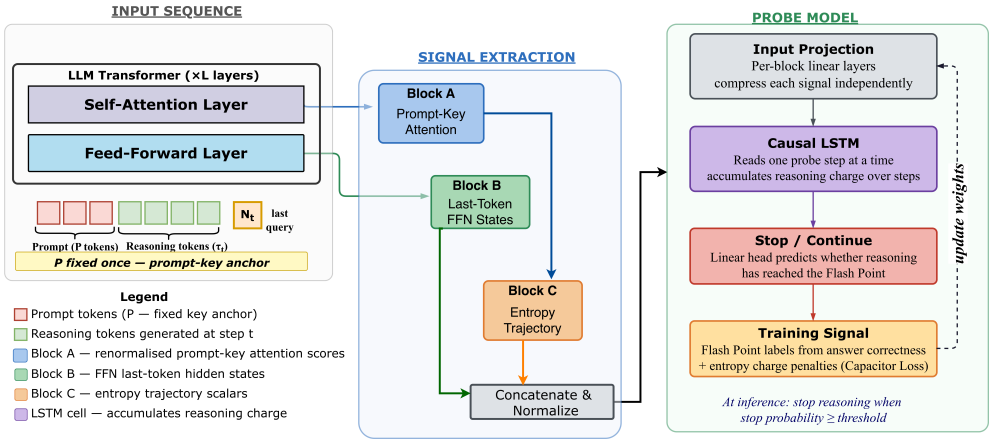


Figure 1: **Flash Point Probe architecture.** At each probe step, three signals are extracted from the frozen LLM: Block A (prompt-key attention), Block B (last-token FFN states), and Block C (entropy trajectory scalars). The concatenated feature vector is streamed through a causal LSTM, which accumulates reasoning signal over steps. A linear head predicts a stop probability at each step; at inference, generation halts when this exceeds threshold τ.

Let M be a decoder-only transformer-based LLM with L layers, H attention heads per layer, and hidden dimension D. Given an input prompt p, let $P = |\,\text{tokenize}(p)| \in \mathbb{Z}^+$ denote the

prompt length. We analyze entropy over these P tokens during generation to produce a core signal for detecting when the model is ready to commit to the correct answer. During reasoning, we probe the model at intervals of Δ tokens. At probe step $t \in \{1, \ldots, T\}$, the total number of reasoning tokens generated so far is $\tau_t = t \cdot \Delta$ and let $N_t = P + \tau_t$ be the total length at step t. Our probe utilizes signals from both entropy accumulation and latent values (see Figure 1) to identify when the model is ready to commit. Below we go into the details on our probing collection strategies: Blocks A, B, and C.

Block A: Attention Signal Extension At probe step t a full forward pass is executed over the token sequence $x^{(t)} \in Z^{N_t}$. For each layer $\ell \in \{1, \ldots, L\}$ the LLM produces a raw attention weight matrix. The probe restricts attention to the last prompt token (position N_t) attending to the prompt: $\hat{a}_k^{(\ell,h,t)} = A_{h,\, N_t,\, k}^{(\ell,t)}$

Because $\sum_{k=1}^{P} \tilde{a}_k^{(\ell,h,t)} \leq 1$ in general (the remaining mass attends to reasoning positions), the slice is **re-normalized**:

$$\hat{a}_k^{(\ell,h,t)} = \frac{\tilde{a}_k^{(\ell,h,t)}}{\sum_{j=1}^{P} \tilde{a}_j^{(\ell,h,t)} + \varepsilon} \tag{1}$$

where ε prevents division by zero. Heads for which $\sum_k \tilde{a}_k^{(\ell,h,t)} = 0$ (masked or GQA-shared heads) are excluded from all subsequent entropy calculations.

Block B: FFN Activation Signal Feedforward neural network (FFN) layers hold the majority of parametric knowledge encoded in the model, applying non-linear feature-wise transformations that inject capacity independent of the sequence context handled by attention. Specifically, FFNs prevent the collapse of token embeddings, aiding representation isotropy (Geva et al., 2021; Zhang et al., 2024). We therefore collect FFN output signals at the last token position across a selected subset of K transformer blocks. Rather than reading post-residual hidden states, we capture the output of the MLP sub-module directly i.e., the contribution of the FFN *before* the residual connection via a forward hook registered on the MLP block of each selected layer. Formally, at probe step t, the MLP output of the last token at selected layer ℓ_k is: $f^{(k,t)} = \mathrm{MLP}_{\ell_k}\left(x_{N_t}^{(\ell_k)}\right) \in \mathbb{R}^{D_{\mathrm{mlp}}}$, where D_{mlp} is the MLP output dimension (equal to the hidden dimension D for standard architectures) and $x_{N_t}^{(\ell_k)}$ is the input to block ℓ_k at the last sequence position.

Symbol	Formula	Interpretation
ρ_t	$\frac{H_{\mathrm{pre}} - H_t}{H_{\mathrm{pre}}}$	Accumulation
δH_t	$H_t - H_{t-1}$	Flux
$\delta^2 H_t$	$\delta H_t - \delta H_{t-1}$	Acceleration
$\hat{\tau}_t$	$\frac{\tau_t}{\tau_{\max}}$	Reasoning budget

Table 1: Capturing the dynamics of attention via syntropy across all the Layers via accumulation, flux, and acceleration.

Block C: Per-Head Entropy The goal of this block is to capture the dynamics of entropy across all the attention heads in the selected layers in Block B. Table 1 summarizes the signal information. For every attention head (ℓ, h) at step t ($t \geq 1$) the entropy of the re-normalized attention across the prompt-key distribution (from Block A) is calculated: $\mathcal{H}^{(\ell,h,t)} = -\sum_{k=1}^{P} \hat{a}_k^{(\ell,h,t)} \log \hat{a}_k^{(\ell,h,t)}$, Let $\mathcal{S}^{(t)}$ denote the set of active (ℓ, h) pairs at step t. The mean attention entropy $H_t = \frac{1}{|\mathcal{S}^{(t)}|} \sum_{(\ell,h) \in \mathcal{S}^{(t)}} \mathcal{H}^{(\ell,h,t)}$

$$c^{(t)} = \left[H_t,\ \rho_t,\ \delta H_t,\ \delta^2 H_t,\ \hat{\tau}_t\right]^\top \in \mathbb{R}^5 \tag{2}$$

The pre-reasoning baseline entropy (H_{pre}) is computed identically on the question-only prompt (at $\tau = 0$). From the entropy trajectory $\{H_0, H_1, \ldots, H_t\}$ (where $H_0 \equiv H_{\mathrm{pre}}$) the scalars derived at each probe step are shown in Table 1. The scalar block $c^{(t)}$ is the vector that captures the confusion dynamics inside each attention head across all

the layers. The three blocks are flattened and concatenated into a single feature vector: $x_{\text{feat}}^{(t)} = \left[f^{(k,t)} \| \, c^{(t)} \right] \in \mathbb{R}^{D_{\text{feat}}}$

3.1 Probe Model Architecture

Each trajectory $(\tilde{x}^{(1)}, \ldots, \tilde{x}^{(T)})$ is processed by a causal LSTM classifier whose cell state $c^{(t)}$ accumulates reasoning signal across steps. At every step, the LSTM output is passed through a two-layer classification head: a linear projection, ReLU activation, and dropout, followed by a second linear projection to a scalar, producing a raw stop logit $z^{(t)}$, which is converted to a stop probability via the logistic sigmoid. For a trajectory of valid length T, this process yields a logit sequence of length T; since trajectories in a batch differ in length, shorter sequences are zero-padded and a binary validity mask is applied to exclude padded positions from the loss computation. The probe learns the optimal moment to halt reasoning and force the model to begin producing answer tokens. We define this ***"flash point"*** as the moment where the information gain from reasoning is sufficient to steer the model toward the correct answer. The training of the probe is guided by four distinct penalties, detailed below.

Supervision Penalty ($\mathcal{L}_{\text{BCE}}$): This loss penalizes the probe if the answer generated after pausing the reasoning process is incorrect. Let $m_t \in \{0, 1\}$ denote the binary validity mask at step t (1 for valid, non-padded steps; 0 otherwise), and let $\ell_t \in \{0, 1\}$ denote the ground-truth flash-point label at step t (1 if stopping at t yields a correct answer; 0 otherwise). The loss is defined as:

$$\mathcal{L}_{\text{BCE}} = \frac{\sum_{t=1}^{T} m_t \cdot \text{BCE}\left(z^{(t)}, \ell_t\right)}{\sum_{t=1}^{T} m_t} \qquad \text{where} \quad \text{BCE}(z, \ell) = -\ell \log \sigma(z) - (1 - \ell) \log(1 - \sigma(z)) \tag{3}$$

Weak-Flash Penalty ($\mathcal{L}_{\text{wf}}$): This loss penalizes early stops that occur before sufficient evidence has accumulated, encouraging the probe to terminate only when the reasoning signal is genuinely strong. $\hat{p}_{\text{stop}}^{(t)} = \sigma\left(z^{(t)}\right) \in (0, 1)$ denotes the probe's predicted probability of stopping at step t. The loss is defined as:

$$\mathcal{L}_{\text{wf}} = \frac{1}{\sum_{t=1}^{T} m_t} \sum_{t=1}^{T} \left(m_t \cdot \frac{\hat{p}_{\text{stop}}^{(t)}}{\max(\rho_t, 0) + \epsilon} \right) \tag{4}$$

where $\epsilon > 0$ is a small stability constant that prevents division by zero and controls the maximum penalty magnitude. When $\rho_t \approx 0$, the denominator collapses to ϵ, amplifying the per-step cost by a factor of $1/\epsilon$ and acting as a strong deterrent against premature stopping. As ρ_t grows, the denominator increases and the penalty diminishes, so the loss naturally relaxes once sufficient evidence has accumulated.

Stagnation Penalty ($\mathcal{L}_{\text{ic}}$): Reasoning tokens that fail to reduce the entropy of the input prompt are deemed inefficient. This loss penalizes continued generation when the rate of information gain stagnates. We first define the **normalized entropy-reduction rate** at step t: $\Delta Q_t = \frac{-\delta H_t}{H_{\text{pre}}}$, where $\Delta Q_t > 0$ indicates that entropy is actively falling (i.e., the model is resolving uncertainty). The corresponding **penalty cost** is: $\kappa_{\text{ic}}^{(t)} = 1 - \sigma\left(\gamma, \Delta Q_t\right)$, where $\gamma > 0$ is a temperature parameter that controls the sharpness of the transition. The penalty approaches 1 when $\Delta Q_t \approx 0$ (stagnant information gain) and approaches 0 when $\Delta Q_t \gg 0$ (rapid entropy reduction). This penalty is applied only while the probe continues reasoning, weighted by $\hat{p}_{\text{cont}}^{(t)} = 1 - \hat{p}_{\text{stop}}^{(t)} \in (0, 1)$, the predicted probability of continuing at step t. The aggregate **stagnation loss** is computed over the sequence:

$$\mathcal{L}_{\text{ic}} = \frac{\sum_{t=2}^{T} m_t \cdot \hat{p}_{\text{cont}}^{(t)} \cdot \kappa_{\text{ic}}^{(t)}}{\sum_{t=2}^{T} m_t} \tag{5}$$

Saturation Penalty ($\mathcal{L}_{\text{sat}}$): To optimize generation efficiency, we penalize continued reasoning after the entropy reduction of the input prompt tokens has saturated. If the model is no longer gaining information from extended reasoning, it should stop and emit the final answer. The saturation factor at step t is defined as: $\kappa_{\text{sat}}^{(t)} = \exp\left(-\lambda \left|\delta^2 H_t\right|\right)$, where $\lambda > 0$ is a decay rate that controls sensitivity to curvature in the entropy trajectory. When the second derivative of entropy is zero ($\delta^2 H_t = 0$), the entropy trajectory has plateaued and $\kappa_{\text{sat}}^{(t)} = 1$, applying full penalty to continued generation. As the magnitude of the acceleration $|\delta^2 H_t|$ grows indicating the entropy is still actively changing, $\kappa_{\text{sat}}^{(t)} \to 0$ and the penalty is suppressed. A larger λ makes the factor more sensitive, treating even small residual curvature as evidence of saturation. The total saturation loss is then:

$$\mathcal{L}_{\text{sat}} = \frac{\sum_{t=3}^{T} m_t \cdot \hat{p}_{\text{cont}}^{(t)} \cdot \kappa_{\text{sat}}^{(t)}}{\sum_{t=3}^{T} m_t} \tag{6}$$

The sum begins at $t \geq 3$ because computing the second derivative of entropy requires a minimum of two prior steps.

Total Stopping Loss: The four components are combined as a weighted sum:

$$\mathcal{L}_{\text{total}} = \underbrace{w_{\text{bce}}\,\mathcal{L}_{\text{bce}}}_{\text{supervision}} + \underbrace{w_{\text{wf}}\,\mathcal{L}_{\text{wf}}}_{\text{sufficiency}} + \underbrace{w_{\text{ic}}\,\mathcal{L}_{\text{ic}}}_{\text{efficiency}} + \underbrace{w_{\text{sat}}\,\mathcal{L}_{\text{sat}}}_{\text{saturation}} \tag{7}$$

where $w_{\text{bce}} > w_{\text{wf}} > w_{\text{ic}} > w_{\text{sat}} \geq 0$ are scalar hyperparameters. At each training step, the four component losses are computed independently from the current batch and then aggregated via this weighted sum to produce the scalar $\mathcal{L}_{\text{total}}$ on which gradients are back-propagated. The weights encode an explicit priority ordering which ensure that correctness of stop/continue decisions takes precedence over the sufficiency constraint, which in turn outweighs the two continuation penalties.

Inference Decision Rule: At inference time the probe is evaluated incrementally. At each probe step t the model produces $z^{(t)}$ given the hidden state carried over from step $t-1$. Generation switches from reasoning mode to answer mode at

$$t_{\text{pred}}^{*} = \min\left\{\, t \;\middle|\; \sigma(z^{(t)}) \geq \theta_{\text{stop}} \,\right\} \tag{8}$$

where $\theta_{\text{stop}} \in (0, 1)$ is a decision threshold (default 0.5). If no step crosses the threshold before the token budget is exhausted, the model proceeds to answer generation at $t = T$. A trained Flash Point Probe emits a raw logit z_t at each reasoning step t (before the sigmoid). Under the source-domain threshold τ, generation halts at the first step where $\sigma(z_t) \geq \tau$. On a new (transfer) dataset the logit distribution shifts, so a single scalar is learned to recalibrate the stopping probability.

4 Experiments

Our core research question is: ***Using a model's own internal signals, how early can we stop reasoning without damaging performance?*** To answer this, we collect attention entropy signals across multiple model layers and train three independent Flash Point Probe instances, one on GSM8K (Cobbe et al., 2021), a free-form mathematical reasoning dataset in which each item is a word problem with a step-by-step solution ending in a single numeric answer; one on MMLU-Pro (Wang et al., 2024), a broad multiple-choice benchmark spanning academic and professional subjects with up to ten answer options per question; and one on GPQA-main (Rein et al., 2023), a graduate-level, expert-authored multiple-choice science benchmark (biology, physics, and chemistry) in four-option form. Experiments span three model families and three benchmark datasets covering mathematics, mixed academic knowledge, and science, and together vary along two orthogonal axes: subject domain (math $\leftrightarrow$ mixed $\leftrightarrow$ science) and answer format (open-ended numeric $\leftrightarrow$ multiple-choice). A complete overview of the experimental setup is provided in Table 2 in the Appendix. Each probe is evaluated under three transfer regimes defined by the relationship between training and evaluation data.

In-domain evaluation: Each probe is tested on a held-out split of its own training benchmark, establishing an upper bound on performance when the training and evaluation reasoning distributions match.

Cross-domain transfer: Each probe is tested on datasets differing in both subject matter and question format: the GSM8K-trained probe is applied to GPQA-main and the full MMLU-Pro; the GPQA-main-trained probe is applied to GSM8K and the full MMLU-Pro; and the MMLU-Pro-trained probe is applied to GSM8K and GPQA-main. This bidirectional setup stress-tests the entropy-stabilization signal across domains and difficulty levels, including whether a probe trained on multiple-choice science questions can locate Flash Points in open-ended mathematical reasoning, and vice versa.

5 Results and Discussion

In-distribution performance: The probe's largest gains appear on the harder benchmarks GPQA-main and MMLU-PRO, and the signal structure explains why: science- and reasoning-heavy traces show a long settlement period followed by a sharp, unambiguous collapse at the point of commitment, a structure the probe learns easily and detects reliably. The results (see Figure 2) show that the probe offers both substantial decreases in reasoning tokens at similar accuracy across all 3 models and all 3 datasets as well as a more modest but still substantial reduction in reasoning with accuracy improvements for the multiple-choice datasets (differences are the result of varying the probe's threshold for early termination). This reflects the already-strong math baselines, where the greater headway for the probe is in recovering compute rather than accuracy. In all cases the probe waits for genuine settlement rather than truncating prematurely.

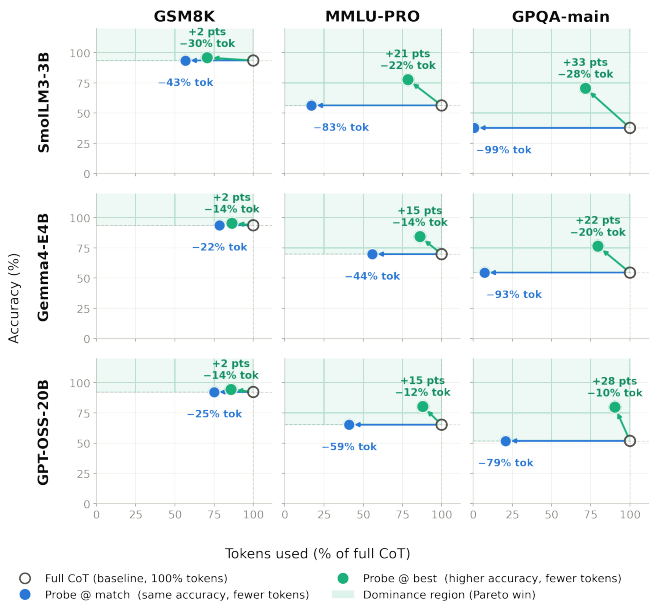


Figure 2: Performance on in-distribution settings. Each probe evaluated on the benchmark it was trained on.

Generalization follows task structure: A GSM8K-trained probe, calibrated to the slow settlement of multi-step arithmetic, preserves accuracy under transfer but recovers little compute when the target task settles at a different pace. On the mixed MMLU-PRO benchmark it stays within a point of the full-CoT baseline (−0.3 for SmolLM3-3B, −1.2 for GPT-OSS-20B) yet saves under 3% of tokens: tuned to slow arithmetic, the probe reads MMLU-PRO's faster, more heterogeneous settlement as a signal to keep going. Transfer to GPQA-main is strongly model-dependent. SmolLM3-3B gains +11.0 while cutting 92.7% of tokens, whereas GPT-OSS-20B and Gemma-4-E4B barely move (−0.7 and +0.0) and save under 4%.

The GPQA-main-trained probe reverses the risk profile. It holds accuracy for SmolLM3-3B on both computation targets (GSM8K +0.0, MMLU-PRO +0.0), but on the larger models a stopping signal tuned to the long commitment latency of open-ended science reasoning fires too aggressively on structured computation: GPT-OSS-20B drops −3.5 on GSM8K

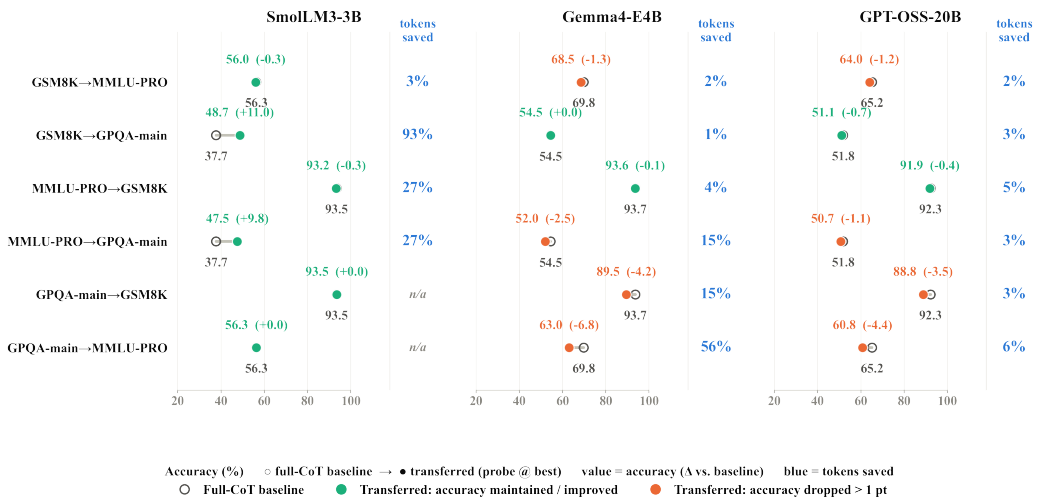


Figure 3: Accuracy and mean tokens saved when the probe is applied outside its training set. "Cross-bench" = each probe evaluated on the other's home benchmark.

and −4.4 on MMLU-PRO, and Gemma-4-E4B drops −4.2 on GSM8K versus the near-lossless GSM8K-trained transfers. The MMLU-PRO-trained probe sits between the two, transferring to GSM8K with negligible loss and moderate savings (−0.3 at 26.5% saved for SmolLM3-3B, −0.4 at 5.2% for GPT-OSS-20B) and improving SmolLM3-3B on GPQA-main (+9.8 at 26.8% saved).

Transfer follows entropy similarity: Transfer is most reliable when the target's settlement profile is easy to read: either accuracy is preserved because the probe defers (as when the GSM8K probe meets MMLU-PRO and simply keeps generating), or the target settles sharply enough that stopping is safe regardless of calibration (as in SmolLM3-3B's GSM8K→GPQA-main transfer, +11.0 at 92.7% saved). The failure mode is the science-trained probe applied to structured computation, whose settlement profile sits awkwardly between rapid recall and extended deliberation, causing premature stops and accuracy loss on the larger models. This motivates a simple deployment rule: use a single in-distribution probe for homogeneous evaluation; expect far-domain settings to need only minimal recalibration (T fit on 50 samples); and treat the science-to-computation direction as the case requiring joint training or a domain-adaptive threshold search.

5.1 Ablation Study

Sparsity is universal: In every combination, for the vast majority of layers, ablation shows near-zero change. That is, nearly all relevant information is spread across multiple layers. GPT-OSS-20B and Gemma-4-E4B each localize to 3–4 "critical" layers out of 24 and 42 respectively; SmolLM-3 shows the flattest, most diffuse profile of the three.

FFN (Block B) ablation is inert: Ablating the FFN block never triggers a meaningful change in token-savings. Given the smaller number of FFN layers sampled, the entropy (Block A) signal seems to be more central to detecting commitment.

Critical-layer identity is model-specific, not dataset-specific: For a fixed model, the same layers are critical regardless of training set. This is a key point: the location of the relevant signals is an intrinsic property of the model, not an artifact of the dataset.

The probe trained on GPQA learns a notion of "commitment" that transfers almost uniformly well to everything downstream: reasoning-heavy (math/physics) and memory-heavy (history/anatomy) datasets show correlated per-layer ablation, and mean relative accuracy after ablation stays high for GPT-OSS-20B and Gemma, with tokens saved in the 17–28%

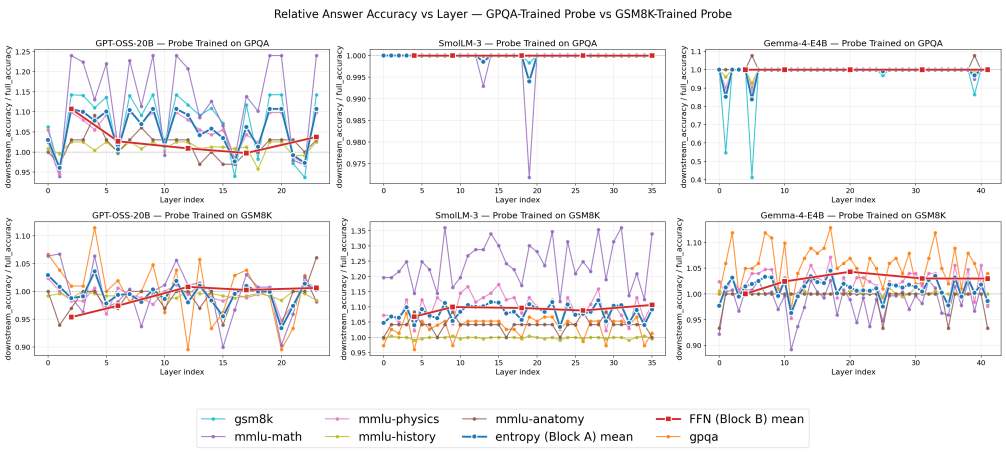


Figure 4: Per-layer ablation shows answer-commitment signal is sparse and model-specific.

range. In contrast, GSM8K's probe learns a narrower notion of commitment. It transfers well to other reasoning tasks (math/physics correlation of 0.68–0.89) but transfers much more weakly and noisily to memory-heavy tasks (history/anatomy correlation as low as 0.43). SmolLM-3 shows the most extreme version of this asymmetry: its GPQA-trained probe essentially never fires when transferred anywhere. The probe fails to find a flash point, while a GSM8K-trained probe transfers with large, accuracy-improving savings on math (50% tokens saved, 1.24× relative accuracy) and physics (39%, 1.10×). For the smaller model, transfer within reasoning-heavy domains (e.g. math/physics) is far more reliable than transfer from a broad-knowledge source like GPQA.

Conclusion

We introduced a mechanism to force-stop reasoning at the ***"flash point"***, the earliest step at which a model has accumulated sufficient information gain to commit to a correct answer, making all subsequent reasoning redundant. Rather than relying on fixed token budgets or heuristic truncation, our probe reads models' attention and entropy-settlement patterns during generation and identifies commitment points on a per-instance basis. The results show this is not merely a compute-saving trick: stopping at the flash point consistently improves accuracy over full-budget reasoning, in some cases by tens of percentage points, while simultaneously eliminating a large fraction of generated tokens: up to 98% on knowledge-retrieval tasks. This confirms that extended chain-of-thought is not neutral; past the flash point, continued reasoning actively degrades the model's answer rather than simply wasting compute. Critically, this pattern replicates across three model families spanning 3B to 20B parameters and across math, science, and knowledge-recall datasets, indicating that the flash point is a real, structural property of how these models reason rather than an artifact of any single model or benchmark.

Our ablation study reinforces this robustness from a mechanistic angle: the commitment signal is carried by a small, model-specific subset of attention layers, and this same subset of layers is identified regardless of which dataset the probe was trained on. This consistency (the same layers mattering across training conditions) is strong evidence that we are detecting a genuine computational event in the model's own reasoning process, not overfitting to surface statistics of a particular dataset. Together, the accuracy gains, the token savings, and the ablation evidence establish the flash point as a robust, model-intrinsic signal for adaptively halting reasoning at the moment it has already done its job.

References

Pranjal Aggarwal and Sean Welleck. L1: Controlling how long a reasoning model thinks with reinforcement learning. *arXiv preprint arXiv:2503.04697*, 2025.

Guillaume Alain and Yoshua Bengio. Understanding intermediate layers using linear classifier probes. *arXiv preprint arXiv:1610.01644*, 2016.

Daman Arora and Andrea Zanette. Training language models to reason efficiently. *arXiv preprint arXiv:2502.04463*, 2025.

Elie Bakouch, Loubna Ben Allal, Anton Lozhkov, Nouamane Tazi, Lewis Tunstall, Carlos Miguel Patiño, Edward Beeching, Aymeric Roucher, Aksel Joonas Reedi, Quentin Gallouédec, Kashif Rasul, Nathan Habib, Clémentine Fourrier, Hynek Kydlicek, Guilherme Penedo, Hugo Larcher, Mathieu Morlon, Vaibhav Srivastav, Joshua Lochner, Xuan-Son Nguyen, Colin Raffel, Leandro von Werra, and Thomas Wolf. SmolLM3: smol, multilingual, long-context reasoner. `https://huggingface.co/blog/smollm3`, 2025.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL `https://arxiv.org/abs/2110.14168`.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. Transformer feed-forward layers are key-value memories. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 5484–5495, 2021.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. Token-budget-aware LLM reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025. arXiv:2412.18547.

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.

Kimi Team. Kimi k1.5: Scaling reinforcement learning with LLMs. *arXiv preprint arXiv:2501.12599*, 2025.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022.

Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747*, 2023.

Michael McCloskey and Neal J. Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, volume 24, pp. 109–165. Academic Press, 1989.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling. *arXiv preprint arXiv:2501.19393*, 2025.

OpenAI. gpt-oss-120b & gpt-oss-20b model card, 2025. URL `https://arxiv.org/abs/2508.10925`.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *International Conference on Learning Representations*, 2024.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*, 2023.

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 Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 677–693, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.36. URL https://aclanthology.org/2025.emnlp-main.36/.

Boshi Wang, Sewon Min, Xiang Deng, Jiaming Shen, You Wu, Luke Zettlemoyer, and Huan Sun. Towards understanding chain-of-thought prompting: An empirical study of what matters. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2717–2739, 2023.

Yongjie Wang, Yibo Wang, Xin Zhou, and Zhiqi Shen. Response uncertainty and probe modeling: Two sides of the same coin in llm interpretability?, 2025. URL https://arxiv.org/abs/2505.18575.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark, 2024. URL https://arxiv.org/abs/2406.01574.

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.

Zhiqiu Xia, Jinxuan Xu, Yuqian Zhang, and Hang Liu. A survey of uncertainty estimation methods on large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 21381–21396, 2025.

Anqi Zhang, Yulin Chen, Jane Pan, Chen Zhao, Aurojit Panda, Jinyang Li, and He He. Reasoning models know when they're right: Probing hidden states for self-verification. *arXiv preprint arXiv:2504.05419*, 2025.

Yang Zhang, Yanfei Dong, and Kenji Kawaguchi. Investigating layer importance in large language models. In Yonatan Belinkov, Najoung Kim, Jaap Jumelet, Hosein Mohebbi, Aaron Mueller, and Hanjie Chen (eds.), *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 469–479, Miami, Florida, US, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.blackboxnlp-1.29. URL https://aclanthology.org/2024.blackboxnlp-1.29/.

A Appendix

Table 2: Experiment configuration summary.

Component	**Parameter**	**Value**
Models	SmolLM3-3B	HuggingFaceTB/SmolLM3-3B
	GPT-OSS-20B	OpenAI/GPT-OSS-20B
	Gemma-4-E4B	Google/Gemma-4-E4B-IT
Datasets	Training	GSM8K, GPQA-main, MMLU-PRO
	Evaluation	GSM8K, MMLU-PRO, GPQA-main
Phase 1 — Collection	CoT token budget	2,048
	Answer token budget	64
	Probe interval	10 tokens
	FFN extraction targets	{0.1, 0.25, 0.5, 0.75, 1.0}
	Drop-ratio thresholds	{0.10, 0.15, 0.20, 0.25, 0.30}
Phase 1 — Features	Block A (attention entropy)	$L \times H$ scalars
	Block B (FFN states)	$K \times d_{\text{inter}}$
	Block C (temporal dynamics)	5 scalars
Phase 2 — Probe Architecture	Hidden dim (swept)	{128, 256, 512}
	LSTM layers	2
	Activation (swept)	{ReLU, GELU, SiLU, SwiGLU}
	Norm order (swept)	{pre, post}
	Dropout (swept)	{0.2, 0.3, 0.4}
	Scalar projection dim	16
	Head projection divisor	2
	Loss	Capacitor (BCE + IC + WF + SAT)
Phase 2 — Training	Learning rate (swept)	{5e-4, 1e-3, 2e-3}
	Batch size	8
	Validation fraction	0.20
	Weight decay	5e-4
	Gradient clip norm	1.0
Phase 2 — Sweep Schedule	Rounds	4
	Initial configs per round	{128, 64, 32, 16}
	Epoch budget per round	{1, 2, 3, 4}
	Elimination ratio	0.25
	Total configs evaluated	240
Loss Weights (swept)	w_{IC} (inefficient charging)	{0.1, 0.3, 0.5}
	w_{WF} (weak flash)	{0.1, 0.2, 0.3}
	w_{SAT} (saturation)	{0.1, 0.2, 0.3}
Inference	Operating threshold τ	0.70
	Context length	8,192 tokens
	Decoding	Greedy (temperature = 0)
Threshold Transfer	Method	Temperature scaling
	Calibration samples	50
	Optimization	Bounded 1-D BCE minimization

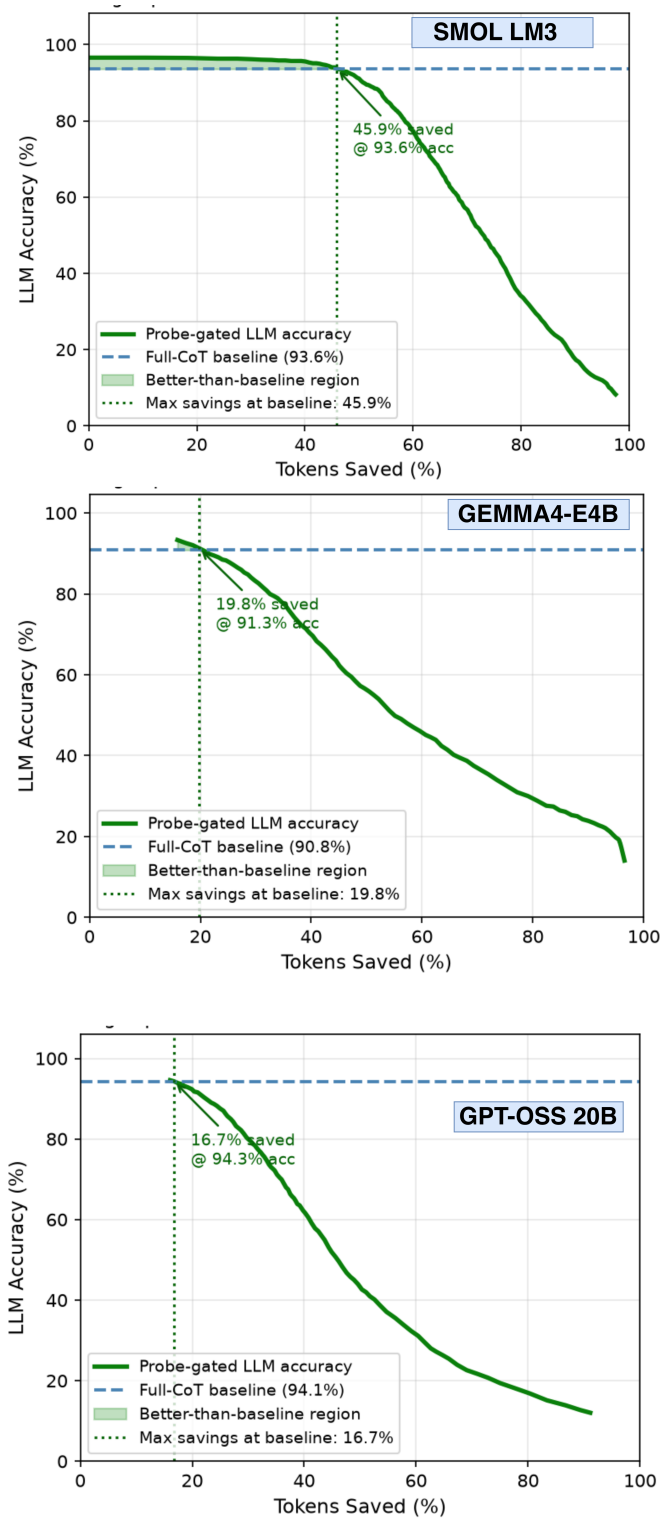


Figure 5: **Accuracy–efficiency tradeoff on GSM8K.** Each curve shows probe-gated accuracy as the stopping threshold is swept from conservative to aggressive. The dashed line marks full chain-of-thought baseline; the shaded region marks where the probe matches or exceeds it. All three models sustain baseline accuracy while saving 16-46% of tokens. N.B. The differences between the numbers reported here and those in the body of the paper are a result of separate runs and their broad similarity is a reflection of the robustness to initialization and run-to-run LLM generation variance.

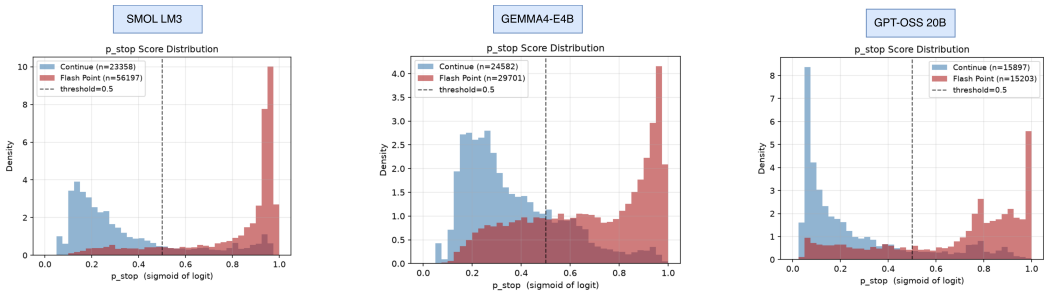


Figure 6: **Predicted stop probability distributions on GSM8K.** Each panel shows $\hat{p}_{\text{stop}}$ density for Flash Point (red) and continue (blue) steps. SmolLM3-3B and GPT-OSS-20B exhibit strong bimodal separation, confirming a discriminative internal signal. Gemma-4-E4B shows greater overlap, consistent with softer entropy settlement dynamics.