What Does Another Loop Buy? Attention Reallocation Tracks the Marginal Value of Test-Time Compute in Looped Transformers

Anonymous authors
Paper under double-blind review

Abstract

Looped transformers repeatedly apply shared weights, making iteration count a natural control for test-time compute. However, the benefit of additional iterations is non-monotonic, while existing adaptive stopping methods typically require training or rely on output-level heuristics. We show that the marginal value of a loop is reflected in attention reallocation between successive iterations. Across controlled synthetic tasks and pretrained looped language models, reallocation peaks near the largest marginal gain and decays as performance saturates; alternative signals based on entropy or hidden-state change do not consistently exhibit both behaviors. Intervention experiments further support a functional role for this reallocation: freezing attention updates near the reallocation peak causes substantial accuracy loss, whereas removing attention produces a distinct, position-dependent damage profile. This observation yields a training-free stopping rule that stops after the last iteration whose reallocation exceeds 10% of its peak, followed by one margin iteration. Both constants are fixed on synthetic tasks with known required depths and transferred unchanged to pretrained models. The rule matches maximum-depth accuracy on all eight synthetic runs while using roughly half the computation, and transfers across two model families and three scales, saving 25% of FLOPs on Ouro-2.6B and 48% on Huginn-3.5B. On GSM8K and ARC-Challenge, the selected stops satisfy pre-registered equivalence criteria relative to maximum-depth inference. Reallocation profiles are also highly consistent across inputs on the evaluated short-answer tasks, enabling a shared model-level stopping decision.

1 Introduction

Looped transformers repeatedly apply a shared block of weights at inference time, allowing a model trained with fixed parameters to trade additional computation for greater effective depth. Prior work suggests that this form of test-time depth scaling can substantially improve performance. A 1.4B looped model has been reported to perform comparably to models with up to 12B parameters (Zhu et al., 2025), while recurrent depth architectures exhibit similar benefits(Geiping et al., 2025; Dehghani et al., 2018; Bai et al., 2019). However, the benefit of additional iterations is not monotonic, and it remains unclear how much recurrence should be allocated at inference time. For example, a looped code model improves after a second pass but degrades with three or more (Yang et al., 2026), while on Ouro-2.6B the final iteration yields little additional gain on GSM8K. These observations motivate a criterion for deciding when additional iterations are unlikely to yield meaningful gains.

Existing approaches typically incorporate the stopping decision into training. Adaptive computation time and related methods learn halting policies jointly with the task objective (Graves, 2016; Banino et al., 2021; Ray Chowdhury & Caragea, 2024), while recent analyses suggest that such mechanisms can couple exit selection with trajectory formation and lead to suboptimal computation schedules (Popescu et al., 2026; Kuo et al., 2026). Confidence and entropy based criteria avoid retraining, but do not reliably identify when additional loops cease to provide further benefit on pretrained looped models (Section 5.1). We therefore focus on a setting without additional training or behavioral supervision. *Given a looped model and its internal activations, can the marginal value of an additional loop be estimated directly from the forward computation?*

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.

We characterize loop value using attention reallocation rather than output confidence. For iteration t, reallocation is measured by the ℓ_1 distance and cosine complement between the attention patterns at iterations t and $t-1$, averaged over all layers and heads at the answer position. We evaluate this signal on controlled synthetic tasks with required depths ranging from 2 to 6 loops and on three pretrained looped LLMs spanning two families, three scales, and four benchmarks. Across these settings, the reallocation profile tracks the iteration of largest marginal gain, the range over which additional loops remain beneficial, and the point at which further computation provides little additional benefit.

Our contributions are fourfold. (i) We characterize the relation between attention reallocation and the marginal benefit of recurrent computation. Reallocation peaks near the largest marginal gain and decays as performance saturates, while its *active window* tracks the required depth on synthetic tasks and captures distinct decay patterns across model families (Section 4). (ii) We provide intervention evidence for the functional role of reallocation. Freezing attention updates near the reallocation peak reduces accuracy by up to 18 points and the degradation persists with additional loops, whereas removing attention produces a distinct recovery pattern when further iterations are available (Section 4.4). (iii) On the evaluated short-answer tasks, useful recurrence depth is stable across inputs, and among the candidate signals, only reallocation consistently exhibits both a peak and subsequent decay (Section 5.3). (iv) We derive a training-free stopping rule with two constants fixed on synthetic tasks that transfers across two model families and three scales, saving 25% of FLOPs on Ouro-2.6B and approximately half of the recurrence budget on Huginn-3.5B (Section 4.2).

We report computational savings primarily in FLOPs, which directly measure the reduction in executed computation independently of implementation details. Related work on looped transformers, adaptive halting, and early exit is discussed in Appendix A.

2 Preliminaries

2.1 Looped architectures and test-time depth

A looped transformer repeatedly applies a shared block of layers, allowing inference compute to be varied through the number of iterations. We consider pretrained looped models from two architectural families together with controlled synthetic models. These settings differ in how computation is repeated and whether latent state is propagated across iterations.

Ouro (Zhu et al., 2025) executes its full layer stack once per "UT step", with at most $S{=}4$ steps. The 2.6B and 1.4B variants contain 48 and 24 layers, respectively, and generation may terminate after any step. We denote the one indexed execution depth by d and describe the exit argument semantics in Appendix D. Each step reprocesses the same sequence without reinjecting a latent state.

Huginn (Geiping et al., 2025) uses Raven-3.5B, with 2 prelude layers, a 4 layer recurrent core repeated R times, and 2 coda layers. At each recurrence, the hidden state at the answer position is reintroduced through an adapter, while R is sampled from a Poisson-lognormal distribution with mean 32 during training and remains configurable at inference time. We use R for evaluated recurrence counts and r for signal derived locations such as peaks or threshold crossings, and the contrast with Ouro allows us to test whether the observed behavior extends beyond architectures without latent state reinjection.

Synthetic models. We train 4-layer looped cores with 0.22M to 2.4M parameters from scratch on a pointer-chasing task. The hop count K defines a lower bound on the recurrence depth required to solve each instance, providing a controlled reference for evaluating whether the proposed signal reflects the required computation depth. Additional details are provided in Appendix G, and Table 2 summarizes the notation used along the loop axis.

2.2 Tasks, evaluation, and compute accounting

We evaluate GSM8K (Cobbe et al., 2021), SVAMP (Patel et al., 2021), and the non-math subset of ARC-Challenge (Clark et al., 2018). GSM8K and SVAMP provide arithmetic reasoning settings, while ARC-Challenge serves as a transfer setting outside arithmetic. Appendix C details the evaluated subsets, prompts, and loop grids. Predictions are greedily decoded and scored by exact match after answer extraction.

We report relative compute as the fraction of full-loop FLOPs executed at a given stopping point. For Ouro with $S{=}4$, stopping at depth d requires $d/4$ of full-loop FLOPs. For Huginn, R recurrences require $(4{+}4R)/(4{+}4R_{\text{full}})$ of full compute, where the constant 4 accounts for the fixed prelude and coda layers. We use FLOPs as the primary compute measure because they are independent of implementation details. Measured wall-clock results are reported in Appendix C.

All accuracy comparisons use paired predictions obtained with identical prompts and decoding settings. We apply the exact binomial McNemar test to discordant pairs, where b denotes examples solved only by the candidate and c those solved only by the reference. We also report a 95% confidence interval for the paired difference $\hat{\Delta} = (b{-}c)/n$ using the Wald standard error (Appendix F). Since McNemar non-significance does not imply equivalence, we pre-register an equivalence margin corresponding to the smallest practically meaningful loss and report equivalence only when a TOST at this margin rejects. Otherwise, we report the corresponding confidence bound explicitly.

2.3 Operationalizing the value of a loop

Both pretrained model families expose loop count as an inference time control variable. We evaluate a candidate signal according to three properties of recurrent computation. The *benefit peak* is the iteration with the largest marginal gain, $\arg\max_t[\mathrm{acc}(t) - \mathrm{acc}(t{-}1)]$. The *active window* denotes the iterations that yield statistically meaningful marginal gains, and the *stopping point* is the earliest iteration after which accuracy is indistinguishable from full depth inference. These targets are determined from paired evaluation and provide a common reference for comparing candidate signals independently of any specific halting mechanism. The signal must be computable from activations obtained in a single forward pass without labels, retraining, or an additional probe head. We further distinguish model level prediction, where a common stopping point is used across inputs and preserves depth regular batching (Schwethelm et al., 2026), from per sample prediction, where recurrence depth is selected separately for each input. We evaluate the latter using cross validation to account for variation across examples (Section 5.3).

3 Measuring What a Loop Buys

3.1 Reallocation signals

Let $a_t^{\ell,h}$ denote the attention distribution at the answer position for layer ℓ and head h during the tth execution of its recurrent block. We define *attention reallocation* from iteration $t{-}1$ to t as

$$\mathrm{pat}_{\ell_1}(t) = \tfrac{1}{LH}\textstyle\sum_{\ell,h}\|a_t^{\ell,h} - a_{t-1}^{\ell,h}\|_1, \qquad \mathrm{pat}_{\cos}(t) = \tfrac{1}{LH}\textstyle\sum_{\ell,h}\big(1 - \cos(a_t^{\ell,h},\, a_{t-1}^{\ell,h})\big), \quad (1)$$

where the average is taken over the L layers and H heads in the looped block. Both quantities are computed from a single forward pass for $t \geq 2$. The vectors $a_t^{\ell,h}$ are post-softmax attention weights over visible keys. The ℓ_1 distance is computed on the original attention vectors, whereas the cosine complement is computed after ℓ_2 normalization and lies in $[0, 1]$. We average across layers and heads to capture model-wide reallocation. At the peak iteration, the median head has $1 - \cos = 0.87$, while the top 10% of heads ranked by reallocation magnitude account for only 12.3% of the total reallocation (Appendix K).

Reallocation is measured at the answer position, where the representation is used to produce the final prediction. The resulting profiles are consistent under both prompt-only and teacher-forced evaluation (Appendix C). For comparison, we also record the signed and absolute changes in next-token entropy together with the hidden-state change norm $\|\Delta h\|$ (Section 5.1).

We compute Eq. 1 using eager attention and forward hooks that record the final-position attention row at each layer without modifying model weights. The resulting measurement overhead is small relative to the computation saved by early stopping. Implementation details, runtime measurements, and the generation protocol are provided in Appendix C.

3.2 Profiles and the stopping rule

We observe two characteristic forms of reallocation profiles. Ouro and the synthetic models exhibit a sharp peak followed by rapid decay. For Ouro-2.6B on GSM8K, $\mathrm{pat} * \cos$ decreases from 0.810 to

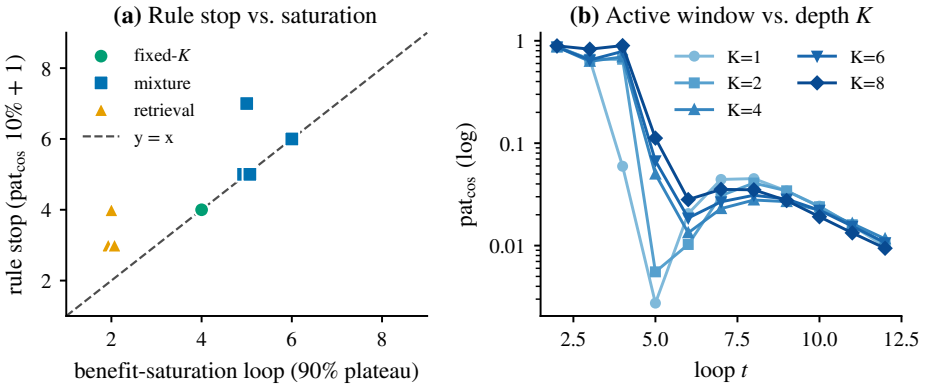


Figure 1: **Synthetic models with controlled depth. (a)** Comparison between the rule stop and the benefit saturation loop across 8 converged runs. The rule does not stop before saturation, adds at most 2 additional loops, and reduces compute by 41.7–62.5% with at most 0.1 point difference in accuracy. **(b)** The active window increases with the required depth K on mix-8 with seed 1. For each K, the profile decays within one loop after the required depth.

0.039 and 0.010 over iterations 2 to 4. Huginn instead shows a more gradual decay, with $\mathrm{pat} * \ell_1$ decreasing by approximately $60\times$ from $r{=}2$ to $r{=}32$. These contrasting profiles are consistent with the architectural differences described in Section 2.1. Ouro repeatedly processes the sequence without latent-state reinjection, whereas Huginn propagates a latent state across recurrences. Accordingly, we use $\mathrm{pat} * \cos$ for rapidly decaying profiles and $\mathrm{pat} * \ell_1$ for gradually decaying profiles. This choice depends only on the reallocation profile and does not use accuracy measurements (Appendix F).

A common stopping rule can be applied to both profile types.

> **Rule.** Let $t^* = \arg\max_t s(t)$ for signal s. Stop one iteration after the last t satisfying $s(t) \geq \theta\, s(t^*)$, with $\theta{=}10\%$.

The additional iteration accounts for residual gains after the signal falls below the threshold. On Ouro, omitting this margin stops one iteration too early across the evaluated settings (Section 4.2), whereas on Huginn the margin provides a more conservative stopping point (Table 10). We therefore retain the same margin across model families. The selected stopping point is stable under bootstrap resampling and perturbations of $\pm 0.5\%$ of the peak signal (Appendix F). The threshold θ controls the tradeoff between compute and stopping conservativeness (Section 5.4), and we use 10% throughout. Both the threshold and the margin are fixed using the synthetic experiments, where the required depth is known, before evaluation on pretrained language models.

We evaluate the main claims using four preregistered criteria defined in Appendix C. These cover peak alignment, zero loss stopping, causal intervention, and per sample robustness.

4 Experiments

4.1 Synthetic tasks with known depth

We first evaluate the signal on controlled tasks whose recurrence-depth requirements are fixed by construction. We consider 8 converged pointer-chase runs covering mixtures over K (mix-8), fixed K, and retrieval-style variants (Appendix G). The threshold $\theta{=}10\%$ and the one-iteration margin are fixed in this controlled setting and held constant in all subsequent pretrained-model experiments.

Peak alignment (G1). The peak of $\mathrm{pat}\ell_1$ or patcos coincides with the benefit peak in 3 of 4 fixed-depth runs and differs by at most one iteration in all 4. In mix-8, the signal window also varies with the required depth. Cases with $K{\leq}5$ require 4 loops, whereas those with $K{\geq}6$ require 5, yielding Spearman $\rho{=}1.0$ on two of three seeds. A control without distractors shows no corresponding

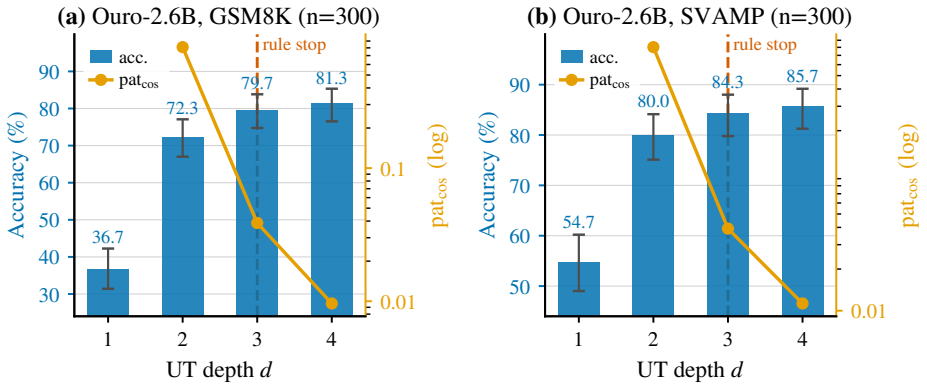


Figure 2: **Ouro-2.6B accuracy and attention reallocation across depth.** Results use greedy decoding and paired evaluation across depths. Bars show accuracy with Wilson 95% confidence intervals, while the orange curves show $\mathrm{pat}_{\cos}$ on the right axis. **(a)** On GSM8K with n=300, $\mathrm{pat}_{\cos}$ decreases to 4.8% and 1.2% of its peak at iterations 3 and 4 as marginal accuracy gains diminish. The stopping rule selects depth 3, corresponding to 75% of full-loop FLOPs. **(b)** SVAMP shows the same qualitative pattern, with reallocation decaying as accuracy saturates and the rule again selecting depth 3.

Table 1: **Ouro at each UT depth.** Greedy accuracy per depth, with the paired comparison against full depth below each accuracy row (b/c: discordant counts; p: exact McNemar). The shaded column marks the rule's stop, depth 3 in all four settings, with its accuracy in **bold**. n=300. Confidence intervals and pre-registered equivalence verdicts are in Appendix F.

Setting	Metric	UT depth d			
		d=1	d=2	d=3	d=4 (full)
GSM8K 2.6B	Accuracy (%)	36.7	72.3	**79.7**	81.3
	McNemar b/c (p)	6/140 ($< 10^{-6}$)	14/41 (0.0004)	13/18 (0.47)	–
SVAMP 2.6B	Accuracy (%)	54.7	80.0	**84.3**	85.7
	McNemar b/c (p)	7/100 ($< 10^{-6}$)	6/23 (0.0023)	6/10 (0.45)	–
ARC-C 2.6B	Accuracy (%)	74.0	86.7	**91.0**	91.0
	McNemar b/c (p)	3/54 ($< 10^{-6}$)	4/17 (0.0072)	3/3 (1.0)	–
GSM8K 1.4B	Accuracy (%)	19.7	65.0	**77.3**	77.3
	McNemar b/c (p)	6/179 ($< 10^{-6}$)	14/51 ($< 10^{-5}$)	15/15 (1.0)	–

dependence on K, indicating that the observed variation is associated with task depth rather than sequence length (Figure 1b).

Accuracy-preserving stopping (G2). The rule stops one iteration after the last point above the threshold. Across all 8 converged runs, the selected stop does not precede the benefit saturation point and exceeds it by at most 2 iterations (Figure 1a). Accuracy differs from maximum depth by at most 0.1 point, while the selected stops use 37.5–58.3% of full-loop compute.

4.2 Main result: Ouro on GSM8K

Figure 2 reports a sweep over UT depth for Ouro-2.6B on GSM8K, with detailed results in Table 1. Accuracy increases substantially over the first two iterations and improves by a further 7.3 points at iteration 3. The final iteration adds only 1.7 points and changes 31 of 300 predictions in each direction, resulting in little net improvement. Over the same iterations, $\mathrm{pat}_{\cos}$ decreases to 4.8% and 1.2% of its peak, consistent with the saturation of marginal accuracy gains.

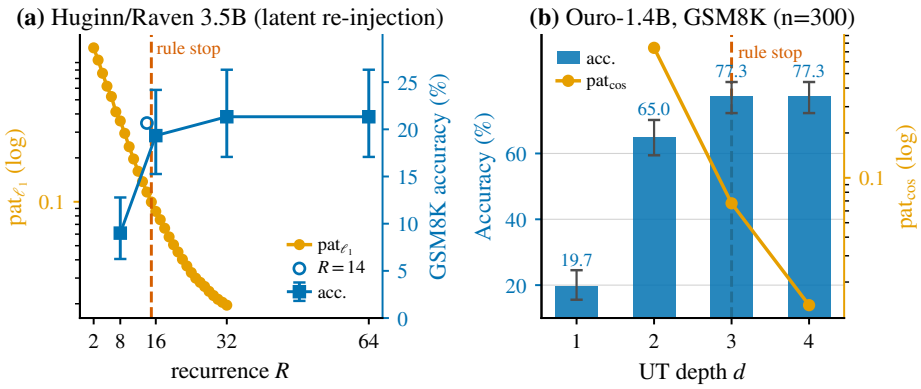


Figure 3: **Replication across model scales and families.** **(a)** Huginn-3.5B with n=300. The pat_{ℓ_1} profile decays gradually, and the selected stop at r=15 lies near the onset of the accuracy plateau. The direct evaluation at R=14 is shown by the open marker. **(b)** Ouro-1.4B with n=300. Reallocation peaks at iteration 2, and the rule selects depth 3, where accuracy matches full depth.

The stopping rule selects depth 3, which uses 75% of full-depth FLOPs. The analysis above uses n=300. On the full 1319-problem test set, the same depth satisfies the preregistered equivalence criterion with δ=3 points. After accounting for the 1.9% measurement overhead, the net FLOP reduction is approximately 24%. The selected depth remains unchanged under sampling with temperature 0.7, whereas stopping one iteration earlier reduces accuracy by 15.3 points (Appendix E).

Effect of the margin. Across the Ouro settings, reallocation at iteration 3 is 4.8–9.0% of its peak, slightly below the 10% threshold. The fixed one-iteration margin moves the selected stop from depth 2 to depth 3 without reference to Ouro accuracy. Without this margin, stopping at depth 2 reduces accuracy by 4.3–14 points, including 72.3% compared with 81.3% on GSM8K. The same threshold and margin are used for all pretrained models, including Huginn. Ouro's four-iteration budget limits the range of possible stopping depths, whereas the synthetic experiments cover required depths from 2 to 6 loops and provide a broader test of whether the signal tracks recurrence depth.

4.3 Replication across benchmarks, scales, and families

SVAMP shows a similar pattern (Table 1), with a reallocation profile of 0.803/0.039/0.011. The final iteration provides little additional accuracy, and the rule selects depth 3, reducing FLOPs by 25% with a 1.3 point decrease in accuracy. The similarity across GSM8K and SVAMP is examined further in Section 5.3.

ARC-Challenge extends the evaluation beyond arithmetic and shows the same rapid decay in reallocation, with $\text{pat}_{\text{cos}} = 0.82 \rightarrow 0.060 \rightarrow 0.032$ over iterations 2 to 4 and a peak CV of 0.008. The rule selects depth 3 and reduces FLOPs by 25%. Reallocation at iteration 3 remains at 7.3% of its peak, and the selected stop satisfies the equivalence criterion with margin δ=2 points (Appendix F).

Ouro-1.4B exhibits the same qualitative behavior at a smaller scale (Figure 3b). The reallocation signal peaks at iteration 2, while the final iteration provides no additional accuracy gain. The rule selects depth 3 because iteration 3 remains at 9.0% of the peak, just below the threshold θ.

Huginn provides the cross family evaluation (Table 10, Appendix J, and Figure 3). Raven-3.5B differs from Ouro by reinjecting latent state and supporting a substantially larger recurrence budget. Accuracy approaches saturation by R=32, while increasing the recurrence count to R=64 provides no further gain at approximately 1.96× the compute. The rule uses pat_{ℓ_1}, whose peak occurs at r=2 and whose threshold crossing gives a stopping location of r=15. On the evaluated grid $\{8, 16, 32, 64\}$, this corresponds to R=16, reducing FLOPs by 48% with a 2.0 point decrease in accuracy. Direct evaluation at R=14, without the margin iteration, reduces FLOPs by 54% with a 0.7 point decrease. The difference between the reported grid point and the signal derived stop therefore results from the recurrence grid used for evaluation (Appendix C). For Huginn, the additional margin gives a more conservative stopping point as the reallocation profile decays gradually toward the accuracy plateau.

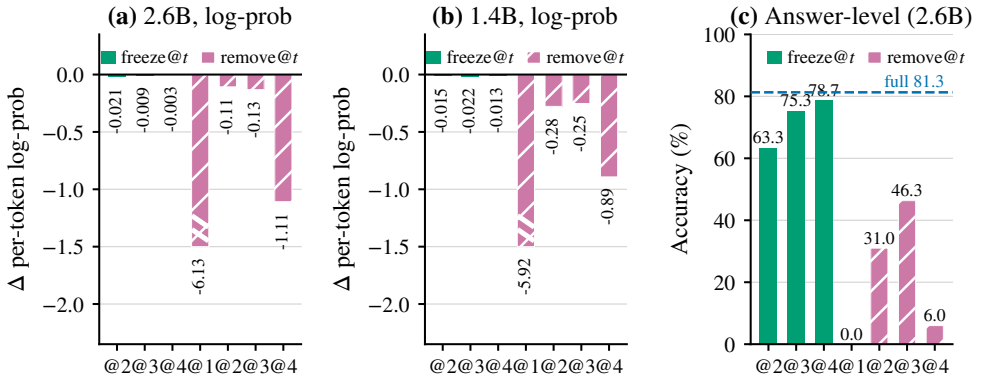


Figure 4: **Effects of freezing and removing loop attention on GSM8K. (a)** Ouro-2.6B and **(b)** Ouro-1.4B results on 100 examples using teacher forced log probability. Freeze damage follows the reallocation profile on 2.6B, whereas remove damage depends more strongly on iteration position. **(c)** Answer level accuracy for Ouro-2.6B with n=300 shows the same distinction.

4.4 Causal ablations of freezing and removing loop attention

To test the functional role of attention reallocation, we perform two interventions under gate G3. **Freeze@**t replaces each head's attention distribution at iteration t with that from iteration $t-1$, suppressing reallocation while preserving the remaining computation at that iteration. **Remove@**t instead sets the attention output at iteration t to zero. We evaluate greedy accuracy at full depth with one modified iteration (n=300). Additional implementation details and teacher forced log probability results are reported in Appendix E.

Synthetic models. Freezing at the reallocation peak produces a persistent loss in accuracy. On mix-8, freeze@2 leaves accuracy at 15% by loop 10, while freezing outside the active window produces no measurable degradation. Removing attention at iteration 2 also reduces accuracy initially, but performance recovers by loop 8. This contrast indicates that preventing the attention update has a different effect from removing the attention output itself.

Ouro models. On Ouro-2.6B, freeze damage follows the reallocation profile, reducing accuracy by 18.0, 6.0, and 2.7 points at iterations 2, 3, and 4, respectively. Remove damage instead depends strongly on iteration position, with the largest effects at the first and final iterations (Figure 4c; Table 3). Freeze@2 is 9 points worse than stopping at depth 2, indicating that preserving a stale attention pattern is more harmful than omitting the corresponding iteration. This difference supports a distinction between disrupting the attention update and removing the attention computation itself. On Ouro-1.4B, the iteration dependence of the freeze effect is weaker, whereas the remove intervention retains a clear position dependence (Figure 4b).

5 Analysis

5.1 Which signals predict loop value?

A candidate signal should identify the iteration of largest marginal gain and decay as additional computation becomes less beneficial. Among the four observables considered, reallocation is the only signal that consistently exhibits both properties.

On the synthetic tasks, $|\Delta\text{entropy}|$ matches the benefit peak in all four runs but remains at 27% of its peak at loop 10, after accuracy has saturated. On Ouro-2.6B GSM8K, signed $\Delta\text{entropy}$ peaks at iteration 3 rather than at the iteration 2 benefit peak, while $\|\Delta h\|$ varies only from 13.2 to 16.6 across loops 2 to 10. Similar discrepancies occur on Ouro-1.4B, SVAMP, and Huginn (Appendix I). Across the evaluated settings, reallocation is the only observable whose profile consistently tracks both marginal gain and subsequent saturation.

Entropy reflects changes in the output distribution, whereas reallocation measures changes in the internal attention computation. This distinction is consistent with the observation of Sharma & Vu (2026) that training objectives primarily supervise variables exposed at the readout, while internal recurrent dynamics may be less directly reflected in output logits.

5.2 Mechanistic interpretation of loop value

The marginal value of an additional loop is associated with changes in its attention pattern over the same context. Freeze damage follows the reallocation profile, whereas remove damage depends more strongly on iteration position. The effect is distributed across heads, with the top quarter ranked by reallocation accounting for only 5 of the 18 point loss. Moreover, partial freezing is more damaging than freezing the full attention pattern (Table 3). Beyond the active window, attention patterns change little and additional iterations provide limited benefit. The sharp decay in Ouro and gradual decay in Huginn may therefore reflect manifestations of a common recurrent transition, consistent with the dynamical-phase view of Kim (2026) and the workspace formulation of Wang & Reid (2026).

The internal redistribution of attention differs across model families. A position-resolved analysis of Ouro-2.6B shows that iteration 2 shifts attention mass from the exemplar prefix toward the sink position, whose share increases from 4.4% to 59.8%. The question region changes little, and the position-level changes at iteration 3 are approximately two orders of magnitude smaller. This pattern is related to previously reported attention sink and large activation phenomena (Xiao et al., 2024; Sun et al., 2024). In contrast, SMELT looped MoE models redistribute attention from sink positions toward content tokens (Wang et al., 2026b). The transferable property appears to be the magnitude and temporal profile of reallocation rather than its spatial direction (Appendix K). Excluding sink attention preserves the profile shape, while shifting the stopping point to full depth, suggesting that sink concentration strengthens the observed decay but does not by itself determine the profile.

5.3 model-level stability of the stopping point

When applied independently to each example, the rule selects the same stopping depth for most inputs. On Ouro-2.6B, it selects depth 3 for all 300 GSM8K and SVAMP examples, and for most examples on Ouro-1.4B ($84/100$ at $n{=}100$) and ARC-Challenge ($254/300$). per-sample $\mathrm{pat}_{\cos}$ at the peak iteration shows little variation, with coefficients of variation between 0.009 and 0.013. These results suggest that the stopping point is primarily a model-level property rather than an input-specific one.

We further evaluate whether per-sample variation provides useful stopping information (Figure 5; Appendix H). Reallocation at the decision iteration is similar across examples that are solved, changed, or remain incorrect after the final loop, with limited separation between groups (AUC 0.58). Although per-sample threshold sweeps identify an apparent lossless configuration at 84.5% of full compute, the same effect is reproduced by a permutation null. Split-half cross-validation reduces the apparent gain to -1.2 and -1.3 points on GSM8K and SVAMP, respectively, while the result on Ouro-1.4B is consistent with the population-level effect. These analyses provide little evidence that per sample adaptation improves over a shared stopping point.

The resulting stopping rule can therefore be applied at the model-level in the short-answer settings considered here, without selecting depth separately for each input. This conclusion does not extend uniformly to harder reasoning settings. On MATH, accuracy continues to improve beyond the reallocation window, leaving a 4.2 point gap relative to full-depth inference (Appendix F). Under Huginn's chat template, however, the reallocation profile remains similar and the selected stops match full depth performance (Table 4).

5.4 Thresholds, calibration, and baselines

Threshold sensitivity. On Huginn, $\theta{=}5\%$ selects $r{=}20$, $\theta{=}10\%$ selects $r{=}15$, and $\theta{=}20\%$ selects $r{=}11$, illustrating the tradeoff between compute reduction and proximity to the accuracy plateau. On Ouro, the selected stop remains stable over benchmark-specific ranges of θ (Appendix F). Because the signal itself is label-free, its profile can be examined on unlabeled prompts without reference to task accuracy. A drop below 10% of the peak is consistent with the default threshold, while residual values near the threshold, such as 7.3% on ARC and 9.0% on Ouro-1.4B, illustrate the role of the additional margin iteration. The signal variant in Section 3.2 is selected in the same way from the profile alone.

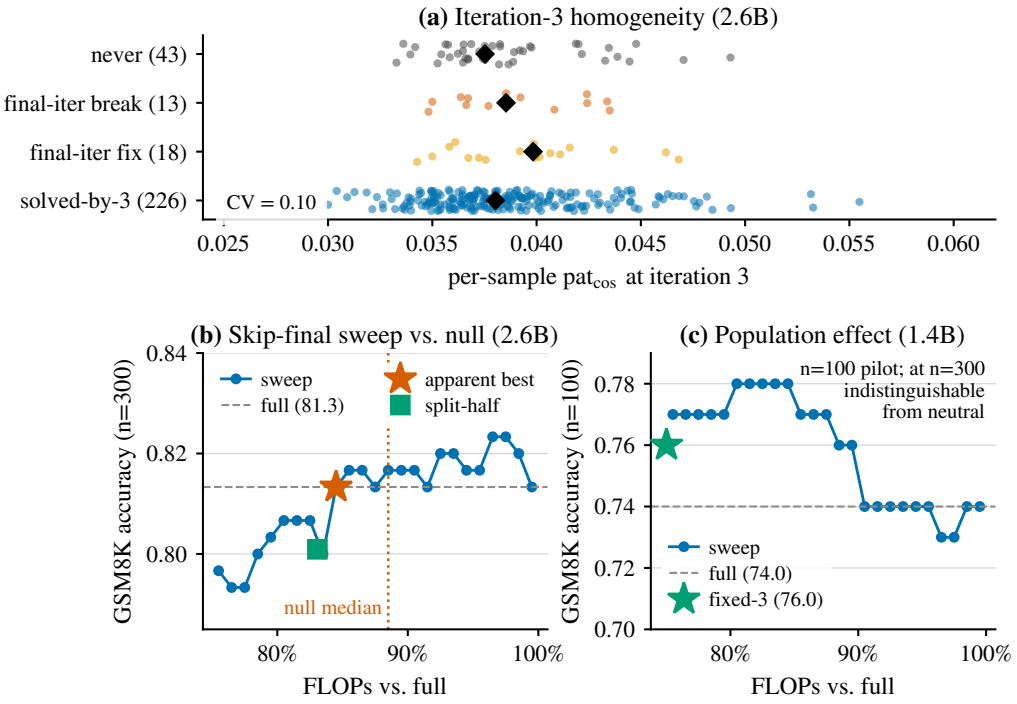


Figure 5: **The per-sample resolution boundary** (Ouro, GSM8K). **(a)** Iteration-3 reallocation by outcome group (solved-by-3 n=226, final-iteration benefit n=18 and regression n=13, never-solved n=43): all 300 samples fall in $[0.030, 0.056]$, and the group medians (diamonds) are indistinguishable (AUC 0.58). **(b)** 2.6B: per-sample threshold sweeps (skip the final iteration for samples with $\text{pat}_{\cos,2} < \tau$) produce an apparent lossless skip point at 84.5% compute, which the permutation null (dotted) reproduces; split-half validation gives -1.2pt. **(c)** 1.4B: the apparent per-sample gain equals the population effect, and the null selects the same configuration.

The peak-to-next-iteration ratio is at least 11 for all rapidly decaying profiles, compared with 1.30 for Huginn. This separation distinguishes the two profile types without using accuracy measurements.

Baselines. A labeled calibration sweep on the same evaluation sets recovers the same stopping point in all five settings. Behavioral exit criteria yield less favorable accuracy and compute tradeoffs. On Ouro, output stability uses 81% of full compute with a 0.3 point accuracy decrease, while on Huginn the KL-based exit achieves 16.0% accuracy at matched compute compared with 20.7% for the reallocation rule (Appendix J). These criteria can determine whether to exit only after the corresponding recurrent computation has been executed, whereas reallocation is available from the prefill and can support depth-batched scheduling (Schwethelm et al., 2026).

6 Conclusion

We show that attention reallocation provides an internal signal of the marginal value of recurrent computation in looped transformers. The signal identifies the iterations over which additional computation remains useful and supports a training-free stopping rule that reduces FLOPs by 25% on Ouro-2.6B. The selected stops on GSM8K and ARC-Challenge satisfy preregistered equivalence criteria, while the same two constants transfer without modification across two model families and three scales. On synthetic tasks with required depths from 2 to 6 loops, the rule remains within 0.1 point of maximum-depth accuracy across all eight converged runs while using roughly half the computation. Freeze and removal interventions further support a functional role for reallocation, and the selected stopping depth remains highly stable across inputs on the evaluated short-answer tasks.

AI Use Statement

Generative AI tools assisted with experiment orchestration, analysis code, and language editing in parts of the manuscript. We reviewed and verified all AI-assisted outputs and take responsibility for the final content of this work.

Reproducibility Statement

The signal definitions, stopping rule, causal-ablation protocol, and validation procedure are described in the main paper and appendix; a minimal implementation of the rule is given in Appendix C. We will release the evaluation code, per-sample signal dumps, and processed experimental artifacts needed to reproduce the reported results.

References

Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. Deep equilibrium models. In *Advances in Neural Information Processing Systems*, 2019.

Andrea Banino, Caswell Barry, Adrià Badia, Kai Bundesen, Gedas Bertasius, et al. PonderNet: Learning to ponder. *arXiv preprint arXiv:2107.05407*, 2021.

Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try ARC, the AI2 reasoning challenge. *arXiv preprint arXiv:1803.05457*, 2018.

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. *arXiv preprint arXiv:2110.14168*, 2021.

Róbert Csordás, Kazuki Irie, Jürgen Schmidhuber, Christopher Potts, and Christopher D. Manning. MoEUT: Mixture-of-experts universal transformers. In *Advances in Neural Information Processing Systems*, 2024.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Lukasz Kaiser. Universal transformers. *arXiv preprint arXiv:1807.03819*, 2018.

Maha Elbayad, Jiatao Gu, Douwe Kiela, and Emmanuel Grave. Depth-adaptive transformer. In *International Conference on Learning Representations*, 2020.

Angela Fan, Edouard Grave, and Armand Joulin. Reducing transformer depth on demand with structured dropout. In *International Conference on Learning Representations*, 2020.

Jonas Geiping, Sean McLeish, Neel Jain, John Kirchenbauer, Siddharth Singh, Brian R. Bartoldson, Bhavya Kailkhura, Abhinav Bhatele, and Tom Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. *arXiv preprint arXiv:2502.05171*, 2025.

Angeliki Giannou, Shashank Rajput, Jy-yong Sohn, Kangwook Lee, and Dimitris Papailiopoulos. Looped transformers as programmable computers. *arXiv preprint arXiv:2301.13196*, 2023.

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

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, 2017.

Gunn Kim. Dynamical phase selection controls compute scaling in looped transformers. *arXiv preprint arXiv:2608.26556*, 2026.

Hsun-Yu Kuo, El Mahdi Chayti, Patrik Reizinger, Wieland Brendel, and Martin Jaggi. Stabilizing extrapolation in looped transformers via learned stochastic stopping. *arXiv preprint arXiv:2606.29983*, 2026.

Guanghao Li, Wenhao Jiang, Li Shen, Ming Tang, and Chun Yuan. Zero token-driven deep thinking in LLMs: Unlocking the full potential of existing parameters via cyclic refinement. *arXiv preprint arXiv:2502.12214*, 2025.

Shuzhen Li, Yifan Zhang, Jiacheng Guo, Quanquan Gu, and Mengdi Wang. DeepLoop: Depth scaling for looped transformers. *arXiv preprint arXiv:2607.13491*, 2026.

Jialin Liu, Lisang Ding, Stanley Osher, and Wotao Yin. Expressive power of implicit models: Rich equilibria and test-time scaling. *arXiv preprint arXiv:2510.03638*, 2025.

James O'Neill and Fergal Reid. Looped latent attention: Cross-loop KV compression for looped transformers. *arXiv preprint arXiv:2607.15456*, 2026.

Francesco Pappone, Donato Crisostomi, and Emanuele Rodolà. Two-scale latent dynamics for recurrent-depth transformers. *arXiv preprint arXiv:2509.23314*, 2025.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. Are NLP models really able to solve simple math word problems? *arXiv preprint arXiv:2103.07191*, 2021.

Andrei Cristian Popescu, Haitz Sáez de Ocáriz Borde, and Pietro Liò. Adaptive depth in looped transformers: Diagnosing learned halting gates and trajectory readouts. *arXiv preprint arXiv:2607.20519*, 2026.

Hayden Prairie, Zachary Novack, Taylor Berg-Kirkpatrick, and Daniel Y. Fu. Parcae: Scaling laws for stable looped language models. *arXiv preprint arXiv:2604.12946*, 2026.

Jishnu Ray Chowdhury and Cornelia Caragea. Investigating recurrent transformers with dynamic halt. *arXiv preprint arXiv:2402.00976*, 2024.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dheeraj Rajagopal, Huaixiu Zheng, Antonio Torralba, and Yejin Choi. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems*, 2022.

Kristian Schwethelm, Daniel Rueckert, and Georgios Kaissis. Depth-adaptive inference of looped language models via continuous depth batching. *arXiv preprint arXiv:2608.09444*, 2026.

Rituraj Sharma and Tu Vu. Dense supervision is not enough: The readout blind spot in looped language models. *arXiv preprint arXiv:2606.24898*, 2026.

Mingjie Sun, Xinlei Chen, J. Zico Kolter, and Zhuang Liu. Massive activations in large language models. *arXiv preprint arXiv:2402.17762*, 2024.

Shaowen Wang, Bingrui Li, Ge Zhang, Wenhao Huang, Shen Yan, and Jian Li. On the residual scaling of looped transformers: Stability and transferability. *arXiv preprint arXiv:2606.18524*, 2026a.

Shaowen Wang, Ge Zhang, Kairong Luo, Yuhao Wu, Shaofan Liu, et al. SMELT: Scaling laws for compute-matched MoE looped transformers. *arXiv preprint arXiv:2609.01343*, 2026b.

Wenlong Wang and Fergal Reid. Looped transformers under the Jacobian lens: Does the global workspace survive recurrence? *arXiv preprint arXiv:2609.01924*, 2026.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. *arXiv preprint arXiv:2201.11903*, 2022.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations (ICLR)*, 2024.

Jian Yang, Shawn Guo, Wei Zhang, Tianyu Zheng, Yaxin Du, et al. LoopCoder-v2: Only loop once for efficient test-time computation scaling. *arXiv preprint arXiv:2606.18023*, 2026.

Liu Yang, Kangwook Lee, Robert Nowak, and Dimitris Papailiopoulos. Looped transformers are better at learning learning algorithms. *arXiv preprint arXiv:2311.12424*, 2023.

Rui-Jie Zhu, Zixuan Wang, Kai Hua, Tianyu Zhang, et al. Scaling latent reasoning via looped language models. *arXiv preprint arXiv:2510.25741*, 2025.

A Related Work

Looped and recurrent-depth transformers. Weight sharing across depth dates to the Universal Transformer (Dehghani et al., 2018), with expressivity established by programmability results (Giannou et al., 2023; Yang et al., 2023), implicit-model analysis showing more iterations enlarge the expressible function class (Liu et al., 2025), and fixed-point variants that train through the infinite-depth limit (Bai et al., 2019). The line has scaled to practical language models—MoEUT (Csordás et al., 2024), recurrent transformers with dynamic halting (Ray Chowdhury & Caragea, 2024), and the two families we study, Ouro (Zhu et al., 2025) and Huginn (Geiping et al., 2025). A parallel literature studies loop training stability and inference infrastructure (Prairie et al., 2026; Wang et al., 2026a; Li et al., 2026; Wang et al., 2026b; Schwethelm et al., 2026; O'Neill & Reid, 2026), and non-monotone returns to extra loops (Yang et al., 2026). It asks how to *train* stable loops; we ask how to *spend* loops at test time.

Adaptive computation, halting, and early exit. Adaptive computation time (Graves, 2016) and PonderNet (Banino et al., 2021) learn the stopping policy jointly with the task, and recent work shows halting gates entangle exit selection with trajectory formation (Popescu et al., 2026) and that learned stochastic stopping can stabilize suboptimal compute (Kuo et al., 2026). Fixed-depth early exits instead use output-level signals—depth-adaptive attention (Elbayad et al., 2020), structured layer dropout (Fan et al., 2020), CALM's next-token confidence (Schuster et al., 2022)—and loop-native criteria adapt the idea to recurrence: Huginn's KL exit (Geiping et al., 2025), ZTT's learned loop controller (Li et al., 2025), and two-scale latent dynamics (Pappone et al., 2025). All of these retrain part of the model or read the output distribution, the same confidence family that fails to track looped benefit in our taxonomy (Section 5). Our rule is strictly post hoc: it reads attention reallocation from an ordinary forward pass, needs no halting head, calibration (Guo et al., 2017), or behavioral labels, and comes with causal ablations that learned policies lack. We compare against the shipped output-stability and KL exits at matched compute in Section 5.4 and Appendix J.

Test-time compute. Chain-of-thought spends test-time compute on generated tokens (Wei et al., 2022); looped models spend it on latent depth (Zhu et al., 2025; Geiping et al., 2025), with latency advantages. The compute is spent inside one forward pass, so unlike sampling-based scaling (best-of-n, self-consistency) there are no samples to compare and the exit decision must come from the computation itself. Prior work characterizes *whether* extra latent depth helps; we read out *how much the next iteration* helps, validated against controlled-depth synthetic tasks and grounded in a model-level invariant—the stop is a property of the model, not the input.

B Notation

Table 2: Notation for the loop axis and the stopping rule.

Term	Meaning
The loop axis	
loop / iteration	one execution of the shared block
UT step (Ouro)	one pass of the entire layer stack; $S \leq 4$
depth d	UT steps executed, one-indexed (Ouro)
recurrence R (Huginn)	repetitions of the 4-layer core
The stopping rule	
θ=10%	stop threshold: fraction of peak signal above which an iteration counts as active
margin iteration	the $+1$ iteration run after the last active one

C Experimental protocol details

Models. Ouro-2.6B and Ouro-1.4B are loaded from `ByteDance/Ouro-2.6B` and `ByteDance/Ouro-1.4B` with `trust_remote_code=True`, bfloat16. The 2.6B model has 48 attention layers executed once per UT iteration, up to 4 iterations, and the 1.4B model has 24. Early exit is controlled by the model's `exit_at_step` generation argument, whose exact semantics are given in Appendix D, and each depth is a separate generation run over the same prompts. Huginn is loaded

from `tomg-group-umd/huginn-0125` (Raven-3.5B: 2 prelude, 4 core repeated R times, and 2 coda layers); the recurrence count is set at generation time, and we evaluate $R \in \{1, 2, 4, 8, 14, 16, 32, 64\}$ (pilot n=100 for $R \leq 4$, n=300 for $R \geq 8$; R=14 is the exact rule stop, Section 4.3). The grid is coarse by design and shared with the exit criteria of Table 9. Within it, the rule's named stop (r=15 on pat_{ℓ_1}) rounds up to R=16, and since the two stops differ by 1.3pt in paired difference, the grid resolution rather than the rule sets the last digit of the reported stop; a finer grid would report the rule's exact cost. The profile that selects the stop is not grid-mediated, because the peak and the crossing are measured before generation. Huginn's absolute accuracy is low (0.21 at R=32) because Raven-3.5B is a small base model with weak arithmetic skill, which does not affect within-family comparisons between loop counts.

Prompts and decoding. The math evaluations (GSM8K, SVAMP) use a 4-shot prefix in GSM8K format (`Question: ... Answer: <reasoning>\\#### <number>`), built from training-split exemplars so that prompt format is held constant across the two benchmarks. ARC-Challenge uses the test split of `allenai/ai2_arc` (Clark et al., 2018) with 4 training-split exemplars in a letter-answer format (`Question: ... Answer: (A)`); the predicted letter is the first `([A-E])` group in the generation. Decoding is greedy (`do_sample=False`) with at most 512 new tokens. Answers are extracted as the number following `####` when present and otherwise as the last number in the generation, and scored by exact match against the gold value after normalization (commas, units, trailing zeros). Subsets are drawn with `numpy.random.RandomState(0).choice(test_split, 300, replace=False)`, and Ouro-1.4B uses the identical protocol at the same n=300.

Generation batch size. Batched generation for these architectures requires left padding, and the custom attention classes of both Ouro and Huginn compute position IDs incorrectly under left padding in the transformers version we used, so batched and single-sample outputs diverge on identical prompts. All generations in this paper are therefore single-sample. This affects wall-clock only, since FLOPs accounting is per-token. A two-prompt equality check does not certify batching: we observed such a check pass while 16-sample batches collapsed to near-chance accuracy, and the corruption grows with batch size and padded length.

Signal measurement. Reallocation (Eq. 1) is captured with eager attention and forward hooks that copy each attention layer's row at the final position, per UT iteration or per recurrence. Profiles are measured in two passes that agree, one on the question prompt alone and one teacher-forced on the gold solution: for Ouro-2.6B on GSM8K, $\mathrm{pat}_{\cos}$ is $0.810/0.039/0.010$ (prompt) against $0.854/0.072/0.024$ (solution), and the rule's stop is iteration 3 under both. Ablation damage (freeze and remove) is the change in mean per-token log-probability on teacher-forced solutions relative to the unmodified model. Probe overhead: on a 256-token prefill, eager attention with hooks costs $+37.7$ms against a 2.02s saving from exiting one iteration early, an overhead of 1.9%.

Wall-clock. FLOPs is our primary accounting because it is implementation-independent: it measures the computation a stopping rule saves, whatever stack runs it. Wall-clock on this research stack is not yet representative, since generation is single-sample with eager attention and per-step hook captures, and realizes $\sim 6\%$ of the FLOPs saving on Ouro-2.6B. Closing that gap requires batched, depth-scheduled inference in a serving stack (Schwethelm et al., 2026), which we leave to future work.

Hardware. A single NVIDIA RTX 6000D GPU per run, bfloat16, PyTorch $2.x$ with `trust_remote_code` model classes.

Minimal rule implementation. With per-iteration attention rows $a_{\ell h}^{(t)} \in \mathbb{R}^T$ at the answer position (layer ℓ, head h, iteration t), the entire rule is:

```
p_t   = mean_{l,h} ( 1 - cos(a_t, a_{t-1}) )   # pat_cos, per iteration
peak  = max_t p_t
last  = max{ t : p_t >= 0.10 * peak }          # threshold theta
stop  = last + 1                                # margin loop
```

For smooth-decay profiles (Huginn) the same code with the ℓ_1 distance in place of the cosine complement defines pat_{ℓ_1}. The probe that collects $a^{(t)}$ is a forward hook that copies each attention layer's final-position row once per iteration; nothing else is model-specific.

Validation gates. Every claim passes a pre-registered gate. **G1 (peak alignment):** on synthetic tasks whose benefit peak varies by construction, the signal peak must match it and the active window must overlap it. **G2 (zero-loss stopping):** rule-stopped accuracy must be indistinguishable from full-loop accuracy under exact McNemar, and equivalence is asserted only where the pre-registered TOST passes (Section 2.2). **G3 (causality):** freezing attention patterns at iteration t must damage answers in proportion to the signal, while removing them must damage according to position, and the two must dissociate. **G4 (per-sample stress test):** any per-sample adaptive claim must survive a permutation null and split-half cross-validation (Section 5.3).

D Exit-argument semantics and axis correction

Semantics. Ouro's `exit_at_step` generation argument is a 0-based index into the model's per-iteration hidden states, not a 1-based loop count: setting it to k returns the state after iteration $k{+}1$.

Correction. An earlier sweep of ours labeled that argument one iteration shallow, so the runs labeled "step 3" and "step 4" both selected the final state of the 4-iteration model and produced byte-identical outputs. All results in this paper use the corrected axis (`exit_at_step` $\in \{0, 1, 2, 3\}$, i.e., depths 1–4). The correction changes two numbers from that sweep: depth-3 stopping flips $31/300$ answers for a net -1.7pt (McNemar $p{=}0.47$), and the 72.3% accuracy previously attributed to depth 1 belongs to depth 2. The reallocation profile is unaffected, since it is measured per loop-block iteration: it still collapses across iterations 3–4, and the rule still stops after iteration 3 for the same 25% saving. The teacher-forced ablations patched loop-block iterations directly and are likewise unaffected.

E Full result tables

Table 3: **Answer-level damage of single-iteration interventions**, Ouro-2.6B GSM8K: greedy accuracy at full depth with one iteration patched, $n{=}300$, paired. Freeze@t substitutes iteration $t{-}1$'s attention weights for iteration t's in every layer and head, while the values, output projection, and MLPs of iteration t still run fresh; Remove@t zeroes the attention output at iteration t so the residual passes through. Freeze@t is undefined at $t{=}1$, since there is no earlier iteration to copy. **Bold** marks each intervention's largest damage: freeze damage peaks where the reallocation signal peaks and decays with it, while remove damage follows position, with complete collapse at iteration 1 ($0/300$ correct) and near-total damage at the final iteration. A graded freeze@2 that freezes only the $k\%$ of heads with the largest per-head reallocation drift shows the damage is not localized there: the top 25% costs 5.3 of the 18.0 points ($p{=}0.026$ against clean), the bottom-25% control is indistinguishable from clean (83.0 vs. 81.3, $p{=}0.49$), and the top 50% is worse than freezing all heads (52.3 vs. 63.3), so a partially stale pattern breaks coordination across heads rather than silencing a few critical ones. The $k{=}100\%$ run reproduces the full-freeze@2 records exactly ($0/60$ mismatches), which validates the head-mask machinery.

	Loop-block iteration t			
	1	2	3	4
Freeze@t (stale pattern; FLOPs preserved)				
Accuracy (%)	–	63.3	75.3	78.7
Δ vs. full (pt)	–	$\mathbf{-18.0}$	-6.0	-2.7
Remove@t (attention deleted; residual passes through)				
Accuracy (%)	0.0	31.0	46.3	6.0
Δ vs. full (pt)	$\mathbf{-81.3}$	-50.3	-35.0	-75.3

F Statistical policy: difference intervals, TOST, calibration, and stop stability

Difference intervals for every paired comparison. McNemar non-significance does not by itself establish equivalence; the paired difference $\hat{\Delta} = (b{-}c)/n$ with Wald SE $= \sqrt{b{+}c}\,/n$ has a 95%

Table 4: **Huginn-3.5B on GSM8K (n=300), full recurrence grid.** The shaded rows are the rule's stops: un-margined (r=14, evaluated directly) and margined (r=15, evaluated on the grid at R=16). p is the exact McNemar against R=32 and $\hat{\Delta}$ the paired difference against full depth, in points. We use pat_{ℓ_1} as the primary signal for smooth-decay profiles because $\mathrm{pat}_{\cos}$ approaches its numerical floor near convergence. All Huginn results use the fixed GSM8K 4-shot prefix. A check under the model card's chat template (system prompt, the same exemplars as alternating turns, n=300) preserves the rule: the profile peaks at the same recurrence, the stop moves by at most one step (r=13 vs. 14), and the same stops are indistinguishable from full (R=14: -0.3pt, p=1.0; R=16: -3.0pt, p=0.25). Absolute accuracies are $\sim$12pt higher under the template, so cross-format absolutes are not comparable, and stop-against-full comparisons are paired within each format.

Recurrence	Accuracy (%)	95% CI (%)	Compute (% of full)	p vs. R=32	95% CI of $\hat{\Delta}$ (pt)
R=1†	1.0	–	7	$< 10^{-6}$	–
R=2†	1.0	–	10	$< 10^{-6}$	–
R=4†	8.0	–	16	$< 10^{-4}$	–
R=8	9.0	[6.3, 12.8]	28	1.2×10^{-6}	[−17.4, −7.3]
R=14	20.7	[16.5, 25.6]	46	0.87	[−4.7, +3.4]
R=16	19.3	[15.3, 24.2]	52	0.42	[−6.0, +2.0]
R=32 (full)	21.3	[17.1, 26.3]	100	–	–
R=64‡	21.3	–	196	1.0	–

† Pilot runs with n=100; McNemar against the n=300 R=32 reference uses the 100 common problems.
‡ R=64 matches R=32 in accuracy: $6/6$ discordant problems in each direction, p=1.0.

Table 5: **Causal ablations on Ouro (GSM8K), two damage metrics.** *(a)* Teacher-forced change in mean per-token log-probability against the unmodified model. *(b)* Ouro-2.6B answer-level accuracy at full depth with one iteration patched (n=300, paired), an order of magnitude more sensitive than (a), shown together with the reallocation profile that freeze damage tracks; answer-level damage was not measured on Ouro-1.4B. Freeze@t substitutes iteration $t-1$'s attention weights in every layer and head, while iteration t's values, output projection, and MLPs still run fresh; Remove@t zeroes the attention output at iteration t, so the residual stream passes that iteration's attention through unchanged.

	Ouro-2.6B				Ouro-1.4B			
	t=1	t=2	t=3	t=4	t=1	t=2	t=3	t=4
(a) Teacher-forced damage: Δ log-probability per token								
Freeze@t	–	−0.021	−0.009	−0.003	–	−0.015	−0.022	−0.013
Remove@t	−6.13	−0.11	−0.13	−1.11	−5.92	−0.28	−0.25	−0.89
(b) Answer-level accuracy (%) at full depth, one iteration patched (Ouro-2.6B)								
Reallocation profile pat_{ℓ_1}	–	1.45	0.25	0.13			–	
Freeze@t	–	63.3	75.3	78.7			–	
Remove@t	0.0	31.0	46.3	6.0			–	

interval whose width says what "indistinguishable" can mean at each n. Table 7 reports it for the rule's stops and for the one-loop-shorter stops that the margin guards against.

Pre-registered equivalence margin. δ=3pt was fixed as the smallest equivalence margin worth claiming before any equivalence test was run: one loop shorter than the rule loses 4.3–14pt, so a stopping rule must bound its loss well below that to be useful. On the full GSM8K test set (n=1319) the rule's stop passes at δ=3pt (p_{TOST}=0.030) but not at δ=2pt (p=0.25), and ARC-Challenge passes even at δ=2pt (p_{TOST}=0.007). SVAMP, the 1.4B model (p_{TOST}=0.050 at n=300), and Huginn do not reach the margin, so their claims are stated as confidence bounds rather than as equivalence. MATH is the one setting where the rule stop is significantly below full (p=0.015), which we treat as a measured scope bound on the zero-loss claim.

Table 6: **Output-stability baseline with corrected compute accounting**: stop at the first depth $t \in \{3, 4\}$ whose extracted answer matches depth $t-1$ (stability events evaluated on depths 2–4). Compute is the fraction of full-loop FLOPs executed, $1 - \text{skipped}/4$. Reported savings assume the stability event is observable without paying for the confirmatory generation.

Setting	Mean depth	Compute (%)	Acc. (%)	Δ vs. full (pt)
GSM8K 2.6B	3.23	81	81.0	-0.3
SVAMP 2.6B	3.14	78	85.3	-0.4
GSM8K 1.4B	3.36	84	75.0	$+1.0$

Table 7: **Paired differences vs. the full setting**: point estimate $\hat{\Delta}$, 95% CI, exact McNemar p, and the TOST p-value at the pre-registered equivalence margin δ=3pt: two one-sided z-tests, $H_0^- : \hat{\Delta} \le -\delta$ and $H_0^+ : \hat{\Delta} \ge +\delta$, each at α=0.05, using the Wald SE $= \sqrt{b+c}\,/n$; p_{TOST} is the larger of the two one-sided p-values, and equivalence is supported when $p_{\mathrm{TOST}}<0.05$ (**bold**). *Top:* the rule's stops. *Bottom:* one loop shorter than the rule, which the margin guards against.

Setting	Stop	n	$\hat{\Delta}$ (pt)	95% CI	p / p_{TOST}
The rule's stop					
GSM8K 2.6B	d=3	300	-1.7	$[-5.3, +2.0]$	0.47 / 0.24
GSM8K 2.6B (full set)	d=3	1319	-1.4	$[-3.1, +0.2]$	0.099 / **0.030**
SVAMP 2.6B	d=3	300	-1.3	$[-4.0, +1.3]$	0.45 / 0.11
ARC-C 2.6B	d=3	300	0.0	$[-1.6, +1.6]$	1.0 / $< \mathbf{10^{-3}}$
GSM8K 1.4B	d=3	300	0.0	$[-3.6, +3.6]$	1.0 / 0.050
Huginn	R=16	300	-2.0	$[-6.0, +2.0]$	0.42 / 0.31
Huginn	R=14	300	-0.7	$[-4.7, +3.4]$	0.87 / 0.13
GSM8K 2.6B, T=0.7	d=3	300	-2.7	$[-7.2, +1.9]$	0.31 / 0.44
MATH 2.6B	d=3	500	-4.2	$[-7.5, -0.9]$	0.015 / 0.76
One loop shorter than the rule					
GSM8K 2.6B	d=2	300	-9.0	$[-13.9, -4.2]$	4×10^{-4} / 0.99
SVAMP 2.6B	d=2	300	-5.7	$[-9.2, -2.2]$	0.002 / 0.93
ARC-C 2.6B	d=2	300	-4.3	$[-7.3, -1.3]$	0.007 / 0.81
GSM8K 1.4B	d=2	300	-12.3	$[-17.6, -7.1]$	$< 10^{-5}$ / 1.0
Huginn	R=8	300	-12.3	$[-17.4, -7.3]$	10^{-6} / 1.0

Instantiation choice. The two instantiations of Eq. 1 decay at different relative rates, so the rule uses one per family, fixed by the decay regime: $\mathrm{pat}_{\cos}$ for spike-then-collapse (the synthetic runs and every Ouro setting) and pat_{ℓ_1} for smooth decay (Huginn). The choice is consequential. Under pat_{ℓ_1} all four Ouro settings keep iteration 3 at 17–31% of peak, above θ, so the rule would stop at full depth and save nothing, whereas under $\mathrm{pat}_{\cos}$ they stop at depth 3. On Huginn, $\mathrm{pat}_{\cos}$ crosses θ at r=10, before the plateau's edge (the R=8 grid point loses 12.3pt, p=10^{-6}), while pat_{ℓ_1} stops at r=15; the evaluated grid rounds both to R=16, so the reported stop coincides at this resolution, but the two instantiations are not interchangeable. On the synthetic runs, where θ and the margin were fixed, $\mathrm{pat}_{\cos}$ is the primary instantiation and pat_{ℓ_1} the conservative variant.

Selection criterion. The choice is discrete but not a free parameter fit to accuracy. The two regimes are separated by the shape of $\mathrm{pat}_{\cos}$ itself: at the peak iteration t^* and the one after, $\rho = \mathrm{pat}_{\cos}(t^*)/\mathrm{pat}_{\cos}(t^*{+}1)$ is 20.9, 20.5, 13.7, and 11.0 on the four Ouro settings and at least 11 on the synthetic spike runs, against 1.30 on Huginn. A threshold of ρ=10 selects $\mathrm{pat}_{\cos}$ for spike-then-collapse and pat_{ℓ_1} for smooth decay on every setting we measure, using only quantities available in the same label-free forward pass that produces the signal. Any threshold in $[2, 10]$ gives the same partition on our settings. The criterion assumes a single dominant peak; with two comparable local maxima, t^* would be ambiguous and ρ would not separate the regimes. No setting we measure produces such a profile.

Labeled-calibration baseline. The baseline of Section 5.4 sweeps depth on the labeled problems, which are the same problems the rule is evaluated on and therefore the favorable direction for the baseline. *Argmax accuracy* picks the numerically best depth, and *earliest indistinguishable* picks

the shallowest depth with McNemar $p>0.05$ against full, the compute-efficient objective (Table 8). The compute-efficient calibration coincides with the rule's margined stop on all four Ouro settings

Table 8: **Labeled-calibration baselines vs. the rule.** Stops are UT depths d for Ouro and recurrences R for Huginn. The rule's stop is label-free.

Setting	Argmax accuracy	Earliest-indist.	Rule (margined)
GSM8K 2.6B	4 (full)	3	3
SVAMP 2.6B	4 (full)	3	3
ARC-C 2.6B	3	3	3
GSM8K 1.4B	3	3	3
Huginn	32 (full; 64 ties)	14	16 (signal stop 14)

and with the rule's un-margined signal stop on Huginn, while the naive argmax objective spends full depth on GSM8K/SVAMP. The rule reaches the compute-efficient stop without labels.

Stop stability under resampling. The rule's Ouro stop flips to full depth only if iteration-3 reallocation crosses 10% of peak. Bootstrap resampling of the per-sample profiles (10^4 resamples per setting) never flips any stop: iteration-3 reallocation sits at 4.8% of peak with bootstrap 95% interval $[4.7, 4.8]$ (GSM8K), 4.9% $[4.8, 5.0]$ (SVAMP), 7.3% $[7.0, 7.6]$ (ARC), and 9.1% $[8.9, 9.3]$ (1.4B). A deterministic $\pm 0.5\%$-of-peak perturbation of the iteration-3 signal also flips nothing. The residual exposure is to θ at the boundary rather than to sampling noise: the 1.4B model sits 0.9pt of peak below the crossing, the same setting where a looser θ forfeits the numerically best point (Section 5.4). A soft margin weighted by the residual signal tail would inherit the per-sample noise that Section 5.3 shows to be unexploitable, so we keep the hard one-loop margin.

θ-sensitivity windows (Ouro). Iteration-3 reallocation is $4.8\%/4.9\%$ of peak on GSM8K/SVAMP, so any $\theta \in [5\%, 20\%]$ yields the same stop there. On ARC-Challenge (7.3% of peak) the stop is unchanged for $\theta \in (7.3\%, 20\%]$, and a looser θ would run to full depth, which is still indistinguishable from the rule's stop but saves nothing. On 1.4B (9.0% of peak) a looser θ would likewise run to full depth and forfeit the numerically best operating point.

Difficulty-stratified probe: MATH by level. To test whether the signal survives harder, more heterogeneous inputs, we run the phase-A probe on Ouro-2.6B over 100 problems per difficulty level of the MATH test set (Levels 1–5, n=500), using the same GSM8K 4-shot prefix. The profile is essentially invariant across levels: iteration-2 reallocation is 0.810–0.825 and slightly increases with difficulty, the iteration-3 residual is 5.0–6.4% of peak, far below θ, with per-level coefficient of variation 1–2%, and the per-sample rule stops at depth 3 for $496/500$ problems, with the four exceptions spread across levels. Difficulty changes how much work the model does inside a loop, not where the reallocation event sits.

Generating on the same 500 problems bounds the zero-loss claim. Run to the margined stop (depth 3) against full depth, pooled accuracy is 51.2% vs. 55.4% ($\hat{\Delta} = -4.2$pt, 95% interval $[-7.5, -0.9]$, exact McNemar p=0.015, discordant $(24, 45)$), and every level is at or below full ($-5.0/0.0/-5.0/-6.0/-5.0$pt, each individually indistinguishable at n=100 per level). The profile is identical in shape to GSM8K's, with iteration-4 reallocation at 1.9% of peak and fully collapsed, so the signal still locates the event and the $496/500$ stops are still where it says. What MATH breaks is the co-location of accuracy value with the reallocation window: a small (4pt) but significant tail of value sits one loop past it. On GSM8K-like distributions the final loop is dead compute and the stop is free, while on harder distributions the margined stop trades that tail for one fewer loop pass.

G Synthetic task, training, and full run inventory

Task. The chase task presents a sequence of (subject, relation, object) facts over 48 entities and 4 relations. A chain of K relational hops starts at a query entity, D distractor facts are interleaved, and the answer is the entity at the chain end. K lower-bounds the number of loops a correct solution needs, since each loop can traverse roughly one hop worth of indirection, and D controls whether

shallow pattern-matching shortcuts exist. A retrieval variant places the answer under a single lookup among $D{=}4$ distractors. An anti-shortcut v3 variant adds interference chains designed to break an O(1) structural shortcut we audited in the flat mode.

Architecture and training. A 4-layer looped core, $d_{\text{model}}{=}128$, 4 heads, $d_{ff}{=}512$ (0.22M parameters; up to 2.4M for the largest variants), trained from scratch with Adam (lr 10^{-3}), batch 256, 1500–3000 epochs depending on the variant, loops during training set per task (e.g. 14 for fixed-$K{=}12$), evaluated exact-match on 4096 fresh samples per loop count.

Full run inventory. Eight runs converge to at least 99.9% at their needed depth: fixed-$K{=}4$, mix-8 seeds 1/2, the mix-8 seed-0 ablation carrier, the no-distractor control (mix-8, $D{=}0$), and three retrieval seeds. In all eight, the rule never under-stops, accuracy at the rule's stop is 0.999–1.0, and the compute used is 37.5–58.3%. Per-run (signal stop / true stop / compute): fixk4 (4/4/40%), mix8_D0 (5/5/42%), mix8_abl (6/6/50%), mix8_s1 (5/5/42%), mix8_s2 (7/5/58%), retr_s0 (3/2/37%), retr_s1 (4/2/50%), retr_s2 (3/2/37%).

Twelve runs do not converge to the task solution and are excluded from the zero-loss ratio. We disclose them here: mix-8 seed 0 without ablations stalls at 0.574 while the same configuration converges in its other three runs, so training is bimodal and single seeds are not trusted in this paper; mix-12 (both seeds, 0.59/0.58), mix-16 (four runs, 0.60–0.62), and fixed-$K{=}12$ (0.46) fall into a heuristic basin; and the four anti-shortcut v3 runs (≈ 0.505) show the exact fingerprint of the weakened shortcut ($K{=}1$ near zero, $K{\geq}3$ at the shortcut ceiling, no depth dependence), so weakening the shortcut moved the training equilibrium to the weakened shortcut rather than to chasing. On basin runs the signal profile is flat with no peak, and the rule stops 1–2 loops early, correctly diagnosing that no productive reallocation is happening.

Shortcut audit. The flat-mode chase task admits an O(1) structural shortcut ("the answer is the value whose key is also a value") that a dataset-level heuristic exploits at 0.936 accuracy. The looped models we report do not use it: $K{=}1$ accuracy is 1.0, the per-K loop profile grows in K, and freeze@2 is catastrophic, whereas a shortcut user would not need loop-2 reallocation. The task is not shortcut-free, which is why the synthetic claims rest on the retrieval and fixed-K variants and on window alignment rather than on absolute task difficulty.

Synthetic ablations. All synthetic ablation curves are evaluated on the original saved mix-8 checkpoint with a deterministic eval set, under a protocol whose control (freeze@2) reproduces the stored curve bit-exactly. Freeze@2 leaves accuracy at 15.4% at loop 10, where the un-ablated model has been at 100% since loop 6, and extra loops do not recover it (16.9% at $T{=}12$); freezing at loops 3–6, outside the active window, costs exactly zero. Remove@2 dips ($2.0/6.6/22.4/59.5/91.0/99.0\%$ at loops 2–7, against $6.4/29.1/71.0/97.6/100/100\%$ un-ablated) and fully recovers to 100% by loop 8, and remove@3 and remove@5 return to 100% by loop 7. Removing attention mid-ramp delays convergence by one to three loops, while freezing the pattern at the peak breaks it permanently.

H Per-sample analysis details

Outcome groups. This section details the analysis summarized in Section 5.3 and Figure 5. Corrected-axis outcome groups (solved by depth 3 and still correct at full depth / fixed by the final iteration / broken by it / never solved) are $226/18/13/43$ for 2.6B GSM8K, $247/10/6/37$ for 2.6B SVAMP, and $217/15/15/53$ for 1.4B GSM8K. The per-sample rule ($\text{pat}_{\cos}$, $\theta{=}10\%$, $+1$ margin) stops at depth 3 for $300/300$ GSM8K and $300/300$ SVAMP samples on Ouro-2.6B (accuracy $79.7/84.3$ at 75% compute). On ARC-Challenge, $254/300$ stop at depth 3 and $46/300$ at depth 4, and the simulated per-sample accuracy (90.7%) sits below the fixed depth-3 stop (91.0%), so per-sample deviations from the model-level stop do not help at the same compute. The 1.4B pilot appeared to be an exception: at $n{=}100$, $84/100$ stopped at depth 3 and $16/100$ at depth 4, simulating to 78.0 at 79% of compute, $+4$pt over the pilot's full model. The $n{=}300$ evaluation closes it, with the final iteration indistinguishable from neutral at both depths (77.3% at both, $p{=}1.0$), so the apparent gain was the population effect rather than input-specific adaptation (Section 4.2).

Homogeneity. Per-sample $\mathrm{pat}_{\cos}$ at the peak iteration has CV 0.010 (2.6B GSM8K), 0.013 (1.4B), and 0.009 (SVAMP). At the decision iteration (3) the spread widens to CV 0.10–0.12, with all samples within $[0.030, 0.056]$ on 2.6B, but the derived stop is exactly constant on both 2.6B benchmarks and moves one iteration on 16% of the 1.4B pilot's samples. Next-token entropy at iteration 2 has CV 0.034. The quantity the rule reads therefore varies by one part in a hundred across inputs at the peak.

Threshold-sweep procedure and null. The sweeps behind Figure 5(b,c) use two per-sample skip rules with corrected FLOPs accounting (skipping the final iteration saves $1/4$ of compute, and stopping at depth 2 saves $1/2$): a conservative rule that skips the final iteration when iteration-2 reallocation is below τ, and an aggressive rule that stops at depth 2 when it is below τ. The conservative sweep's apparent zero-loss best on 2.6B GSM8K skips the final iteration for the most stable $\sim 62\%$ of samples, at 84.5% of full compute and accuracy 81.3. The permutation null (200 shuffles of the threshold assignment across samples) produces equally good apparent configurations with median 88.5% compute (p=0.345), and across all 18 cells (three signals, two variants, three settings) the minimum null p is 0.195, so no cell is significant. The aggressive variant's apparent best uses at least 93% of compute on GSM8K and 99% on 1.4B, so the boundary features cannot identify which samples are done early. Split-half validation (100 repetitions; tune τ on one half and evaluate on the other) realizes -1.24pt (GSM8K) and -1.3pt (SVAMP), and only 45% of repetitions hold within 0.5pt of the fixed depth-3 stop.

Group contrasts and answer changes. Final-iteration fixes against solved-by-3 on $\mathrm{pat}_{\cos,3}$: means 0.0395 vs. 0.0387 on 2.6B GSM8K (permutation p=0.39, AUC 0.58; SVAMP p=0.43, AUC 0.60), so the decision-boundary contrast the rule would need is absent. Answer changes between iterations 2 and 3 do associate with iteration-2 reallocation (AUC 0.614, p=0.0045, n=70 on 2.6B GSM8K; 0.665, p=0.0006, n=41 on SVAMP; 0.745, $p=10^{-4}$, n=36 on 1.4B), but on 1.4B this flags a population effect rather than per-sample readiness: the pilot's final iteration looked net-negative ($76\% \to 74\%$ at n=100) yet is indistinguishable from neutral at n=300 (77.3% at both depths), and the null picks the same skip-all configuration (p=1.0). Final-answer changes show no usable association with $\mathrm{pat}_{\cos,3}$ (AUC 0.54/0.62/0.61 for GSM8K/SVAMP/1.4B). One nominally significant SVAMP contrast (final-breaks vs. solved, $\mathrm{pat}_{\cos,2}$, p=0.015, n=6) does not survive correction across the 45 contrasts tested and is not adopted.

I Signal failure details

Synthetic (fixed-K=4). $|\Delta\text{entropy}|$ peaks at loop 3, exactly the benefit peak, but sits at 27% of peak at loop 10, long after accuracy saturated at loop 4, so it cannot stop anything. Signed Δentropy peaks at loop 7, and $\|\Delta h\|$ is flat (13.2–16.6 across loops 2–10, with its maximum at loop 1).

Ouro-2.6B (GSM8K, prompt pass). Signed Δentropy is $-0.349/+0.069/+0.044$ at iterations 2–4, so its argmax is iteration 3 rather than the iteration-2 benefit peak. $|\Delta\text{entropy}|$ decays too slowly to stop: iteration 3 is still 20% of peak (26% on the solution pass), so a 10% rule would not stop before iteration 4, giving zero savings on a model whose final iteration is dead.

Ouro-1.4B and SVAMP. Signed Δentropy's argmax is iteration 4 (1.4B) and iteration 3 (SVAMP), both wrong. $|\Delta\text{entropy}|$ peaks at the wrong iteration (1.4B: iteration 3 at 0.214 against the benefit peak at iteration 2) or decays too slowly (SVAMP: iteration 3 at 16% of peak).

Huginn. The next-token entropy level rises monotonically across recurrences (4.11 at r=1 to 5.06 by r=11, still rising at r=32) while accuracy plateaus, so entropy has no peak and no decay structure, and neither its level nor its delta can locate or exit the active window.

J Huginn exit-criteria comparison

Huginn ships two label-free exit mechanisms (Geiping et al., 2025): a per-token KL-divergence exit between consecutive recurrent states, which stops a token once the KL falls below a threshold, and an argmax-stability criterion, which stops a token once its argmax prediction is stable across a window of recurrences. We swept the KL threshold over $\{5\times10^{-4}, 10^{-3}, 5\times10^{-3}, 10^{-2}, 5\times10^{-2}\}$, ran the

argmax-stability window at its recommended value, and added two further criteria (entropy difference and latent difference), each with n=300 and paired against R=32. A second-order latent-dynamics criterion that improves on the KL exit has since been proposed (Pappone et al., 2025); it operates on curvature statistics of the recurrent latent state, which require instrumentation beyond the released per-token exit API, and we leave that comparison to future work. On our grid, every compute-matched competitor is significantly worse than full, and the rule dominates the frontier (Table 9).

Dynamic KV cache. Per-token adaptive exit requires Huginn's dynamic KV cache with a lookup fallback (`cache_implementation="dynamic"` paired with `cache_lookup_strategy="latest-m4"`). Under a static cache, early-exited tokens leave stale KV slots that later tokens read, and generation collapses into repetition loops with accuracy near zero at otherwise sane compute fractions. All numbers below use the dynamic cache.

A non-monotonic cell. The tightest KL threshold is worse than the next-tightest despite spending more compute (τ=5×10^{-4}: 18.3% at 78%, against τ=10^{-3}: 22.0% at 65%). Under dynamic caching, later tokens read KV entries written at different recurrence depths (`latest-m4` lookup), so a tighter threshold shifts every token's exit time and changes the mix of KV states its continuation sees; we attribute the non-monotonicity to this interaction rather than to an accuracy–compute trade-off.

Table 9: **Huginn-3.5B GSM8K (n=300): shipped exit criteria vs. the fixed-R grid and the reallocation rule** (dynamic cache with `latest-m4` lookup throughout; see the dynamic-cache note above). Shaded rows: the rule. Every compute-matched competitor is significantly worse than full, and the rule dominates the frontier.

Criterion / setting	Accuracy (%)	Compute (% of full)	p vs. R=32
Per-token adaptive exit			
Argmax-stability (window 5)	9.0	32	1.2×10^{-6}
KL exit, τ=5×10^{-2}	5.0	28	3×10^{-10}
KL exit, τ=10^{-2}	15.7	40	0.021
KL exit, τ=5×10^{-3}	16.0	46	0.033
KL exit, τ=10^{-3}	22.0	65	0.88
KL exit, τ=5×10^{-4}	18.3	78	0.16
Entropy diff, τ=10^{-3}	19.0	83	0.31
Latent diff, τ=3×10^{-2}	20.3	84	0.66
Fixed recurrence R			
Fixed R=8	9.0	28	1.2×10^{-6}
Fixed R=14 (rule, un-margined)	20.7	46	0.87
Fixed R=16 (rule, margined)	19.3	52	0.42
Full R=32	21.3	100	–

K Position-resolved reallocation probe

To see what the iteration-2 spike moves, we bucket the attention row at the answer position by prompt region (sink at position 0, exemplar prefix, question, and self, with the question's numeric and last-word tokens broken out) and measure the per-position change in attention mass between iterations (n=16 GSM8K prompts, averaged over all layers and heads). The probe is run on Ouro-2.6B GSM8K only, and the pooling direction was not checked on ARC-Challenge or the 1.4B model, so we claim transfer for the magnitude profile rather than for the geometry.

The spike is a consolidation event. Attention mass drains out of the exemplar region and pools on the sink, an iteration-scale echo of the attention-sink and massive-activation phenomena documented for single-pass models by Xiao et al. (2024) and Sun et al. (2024): the sink's share of total attention rises from 4.4% at iteration 1 to 59.8% at iteration 2 and stays there (59.9% at iteration 3), and the drain is concentrated at a handful of prefix positions, with the single largest position-level loss at −0.51. The question region barely moves, so the productive re-read does not re-inspect the problem statement but reorganizes where attention mass sits. At iteration 3 every region's per-position change is at most 7×10^{-4}, two orders of magnitude below the spike, which is the stillness the stopping rule

Table 10: **Huginn-3.5B on GSM8K** (n=300): greedy accuracy, compute as a fraction of full-loop FLOPs, and exact McNemar p against full (R=32; R=64 is indistinguishable from it, $6/6$ discordant, p=1.0, at 1.96× compute). The shaded columns are the rule's evaluated stops: un-margined r=14 (run directly at R=14) and margined r=15 (rounded on the grid to R=16). Full grid with intervals: Table 4.

	Recurrence R				
	R=8	R=14	R=16	R=32 (full)	R=64
Accuracy (%)	9.0	20.7	19.3	21.3	21.3
Compute (% of full)	28	46	52	100	196
McNemar p vs. full	10^{-6}	0.87	0.42	–	1.0

Table 11: **Where the iteration-2 spike moves attention** (Ouro-2.6B): change in attention mass per position by prompt region, mean over 16 GSM8K prompts, in units of 10^{-3} (bucket values are per-position means, so rows are not directly comparable in total mass; total change sums to ≈0 across positions). **Bold** marks the dominant movement, mass consolidating on the sink; at iteration 3 every region moves two orders of magnitude less.

Prompt region	Attention-mass change per position ($\times 10^{-3}$)	
	iter. 1 → 2	iter. 2 → 3
Sink (position 0)	$\mathbf{+554.0}$	$+0.70$
Self (answer position)	$+25.0$	-0.60
Question	-0.2	$+0.30$
numerals	$+0.9$	$+0.40$
final word	-0.2	$+0.20$
Exemplar prefix	-1.2	-0.03

reads. The direction is family-dependent: on SMELT's looped MoE models the second pass moves mass away from sinks toward content tokens (Wang et al., 2026b), and what transfers across families is the magnitude profile, one large reorganization inside the active window and near-total stillness after it, rather than the geometry.

Is the signal just sink dynamics? Because iteration 2's largest single movement pools mass on the sink, the stopping signal could in principle be a sink artifact. Recomputing it on the same seed-0 GSM8K 300 subset with the sink excluded from both iterations' rows leaves the profile intact: dropping position 0 or positions 0–3 gives a peak of 0.77 versus 0.81 at iteration 2, and the iteration-3 change remains an order of magnitude smaller, at 12.4–12.7% of peak versus 4.8% with the sink included. The sink component sharpens the iteration-3 collapse but is not its cause. The failure mode is one-sided: with sinks excluded the iteration-3 residual exceeds the 10% threshold, so the rule would keep the full depth, meaning sink exclusion can forfeit the compute savings but cannot create a premature stop. The reorganization is also broad rather than sparse: at iteration 2 the median head's pattern is already nearly orthogonal to its iteration-1 pattern ($1 - \cos = 0.87$), and the top 10% of heads by drift carry only 12.3% of the total, with the top 25% carrying 29.9%, barely above the uniform share. A head-masked variant of the freeze patch that can freeze any subset of heads reproduces the full-freeze@2 records sample-for-sample (n=60, zero per-sample mismatches), so the graded top-k intervention is directly runnable from the released per-head drift matrices; its accuracy sweep is reported in Table 3 and confirms that the damage is not localized in the highest-drift heads.