NOT ALL RECURRENCE IS CREATED EQUAL: DEPTH SENSITIVITY DIFFERS BETWEEN RECURRENCE DIMENSIONS OF A SINGLE CHECKPOINT

Anonymous authors
Paper under double-blind review

ABSTRACT

Recurrent language models reuse computation across inference steps, so recurrence depth appears to offer an adjustable test-time compute budget. That interpretation assumes a checkpoint remains reliable when its depth is changed after training. We test it through a controlled within-checkpoint comparison: HRM-Text-1B couples two recurrence dimensions in one set of weights, so varying each in turn holds architecture, scale, data and training schedule fixed, a comparison no cross-model study can make. Its high- and low-level dimensions turn out to respond very differently. Reducing L from three steps to two retains most trained-depth accuracy while removing 25% of recurrent block passes; increasing H from two to four reduces GSM8K accuracy from 84.0% to 10.0% and causes substantial output-format failure. Two configurations with identical total block passes differ sharply (one scores zero on every benchmark while the other does not), so total recurrent compute does not characterise the effect of altered depth. A hidden-state intervention implicates the high-level representation: additional H steps displace it from its trained endpoint while preserving its norm, and interpolating back toward that endpoint recovers most of the lost accuracy before the trained state is fully restored. We situate the result against four single-loop checkpoints from two further families, none of which shows comparable degradation over the depths we evaluate. We distil the findings into a short protocol for validating recurrence depth as an inference-time compute dial.

1 INTRODUCTION

Increasing inference-time computation has become an important means of improving language-model reasoning without increasing model size (Snell et al., 2025). Recurrent-depth models offer a direct mechanism for allocating such computation: a shared block is applied repeatedly to an evolving hidden state, so inference depth can change without introducing a distinct parameter set at every layer (Dehghani et al., 2019; Geiping et al., 2025; Zhu et al., 2025; Wang et al., 2026). Recurrence depth can therefore appear to provide an adjustable test-time "thinking budget".

That interpretation depends on an assumption parameter sharing does not guarantee: a checkpoint trained under one recurrence schedule must remain reliable when its depth changes after training. An additional iteration may refine an existing representation, but it may instead apply an unintended stage of a computation whose useful output belongs to a particular step. Conversely, reducing depth may remove either dispensable refinement or a necessary computational stage. Recurrence depth is a useful inference-time control only when the trained checkpoint remains functional under the proposed intervention.

Prior work shows that recurrent computation can support reasoning and depth extrapolation, but also that excessive iteration can produce overthinking, degraded predictions, or unstable internal dynamics (Bansal et al., 2022; Yang et al., 2026; Blayney et al., 2026). An analysis of the original HRM reports degraded puzzle performance when its hierarchical iteration counts move away from their trained values (ARC Prize Foundation, 2025). Two questions remain insufficiently resolved. First, can different recurrence dimensions *within the same checkpoint* respond differently to altered depth? Second, which recurrent state trajectories accompany depth-sensitive and depth-tolerant behaviour?

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.

Our central comparison is within a single checkpoint. HRM-Text-1B contains coupled high- and low-level recurrent modules trained with $H{=}2$ and $L{=}3$ (Wang et al., 2025; 2026), so varying H and L separately isolates the effect of *which* recurrence is altered from every confound that separates two different models: architecture, scale, training data, tokenizer, objective and schedule are all held fixed. We then situate that result against four single-loop checkpoints from two further families. Huginn-3.5B applies a shared single-loop block and samples recurrence depth during training, with a mean of 32 steps (Geiping et al., 2025). We additionally evaluate three Ouro checkpoints (Ouro-1.4B, Ouro-2.6B and Ouro-1.4B-Thinking), trained with a maximum recurrent depth of four and an adaptive-depth objective (Zhu et al., 2025). This collection supports a within-checkpoint comparison of two recurrence dimensions and a broader comparison of depth-response patterns across families. Because the families also differ in architecture, scale, training data, objectives and implementation, the cross-family comparisons are observational and do not isolate the cause of depth robustness.

Before varying depth we validate HRM-Text-1B's model-specific inference protocol, for which a 20-example diagnostic shows a 40-point gap between the bare and complete interfaces (Section 3), and standardise the evaluation harness for the single-loop checkpoints.

Our contributions are:

- We use a within-checkpoint design to isolate *which* recurrence dimension is altered from the confounds that separate different models, and show that two dimensions of the same checkpoint have qualitatively different tolerance to post-training depth change, so total recurrent compute does not determine the effect. We also validate HRM-Text-1B's model-specific inference protocol; a 20-example diagnostic shows a 40-point gap between the bare and complete interfaces.
- We connect those behavioural responses to recurrent-state trajectories through an H-state intervention showing graded recovery along the direction back to the trained endpoint, a phase-conditioned L-loop analysis, and one additional consistency case.
- We distil the results into a short validation protocol for treating recurrence depth as an inference-time compute dial (Section 5).

Together these results show that recurrence depth is a checkpoint- and dimension-specific operating condition rather than a reliable inference-time compute control guaranteed by parameter sharing.

2 Background and Related Work

Adaptive and recurrent computation. Adaptive-computation methods vary the processing assigned to an input. Adaptive Computation Time and PonderNet learn when to halt repeated computation (Graves, 2016; Banino et al., 2021), while Universal Transformers share parameters across recurrent depth (Dehghani et al., 2019). Deep Equilibrium Models represent weight-tied computation through an equilibrium state (Bai et al., 2019). A parallel line in NLP varies depth by exiting early rather than by iterating: depth-adaptive Transformers (Elbayad et al., 2019), DeeBERT (Xin et al., 2020) and confident adaptive language modelling (Schuster et al., 2022) all allocate a variable number of *distinct* layers per input. Our setting differs in that the same weights are reapplied, so the question is not when to stop but whether the loop remains well behaved at a count it was not trained for.

Closest to our setting is work that makes looped depth itself elastic. Jeddi et al. (2026) train looped Transformers with variable-length trajectories so that inference depth can be shortened or extended without collapse, and Bae et al. (2025) assign recursion depth dynamically per token. Both change *training* so that depth becomes adjustable; we instead diagnose what happens when depth is altered in checkpoints trained without such a provision, which is the situation a practitioner faces with an already-released model. Giannou et al. (2023) show that looped Transformers can implement step-indexed programs, and Xu & Sato (2025) find that conditioning on the timestep changes what a looped model can express; both bear on why an unconditioned shared block may not "know" which stage it is executing, which is one candidate reading of our H results.

Recent language models instantiate recurrent depth in different ways. Huginn-3.5B iterates a shared latent-reasoning block and samples recurrence depth during training (Geiping et al., 2025). The Ouro family repeatedly applies a Transformer stack and trains with an adaptive-depth objective up to a

maximum recurrent depth (Zhu et al., 2025). The Hierarchical Reasoning Model family instead couples high- and low-level recurrent modules operating at different timescales (Wang et al., 2025; 2026). CoTFormer allocates variable computation at the token level (Mohtashami et al., 2025). These architectures make recurrence depth technically adjustable but do not guarantee that changing it after training preserves model behaviour.

Depth extrapolation and recurrent dynamics. Recurrent computation can support extrapolation beyond training-time depth, but additional iterations may also cause overthinking or prediction degradation. Bansal et al. (2022) demonstrate this in recurrent algorithmic models, while more recent work examines excessive recurrence and stability in recurrent-depth Transformers (Kohli et al., 2026; Yang et al., 2026). An analysis of the original HRM reports degraded puzzle performance when its hierarchical iteration counts move away from their trained values (ARC Prize Foundation, 2025).

Stable behaviour need not imply that the complete recurrent state converges to one unchanged representation. Blayney et al. (2026) show that position-specific states can approach distinct endpoints while the aggregate computation follows a cyclic trajectory. Related studies examine the information represented across latent recurrent steps (Lu et al., 2025) and whether output-level supervision sufficiently constrains the states propagated through a recurrent loop (Sharma & Vu, 2026). This literature motivates examining complete trajectories rather than inferring depth robustness from parameter sharing or a single state-similarity statistic.

Our work differs in three respects. First, we compare two recurrence dimensions within the same checkpoint, reducing the architectural and training confounds inherent in cross-model comparisons. Second, we validate the checkpoint-specific inference protocol before attributing performance changes to recurrence depth. Third, we combine behavioural depth sweeps with a direct H-state intervention and an extended L-loop analysis, showing that the same lower-level recurrent weights exhibit different dynamics under different higher-level conditioning states.

Evaluation-procedure effects. Generated answers and probability-based option ranking can yield different assessments of language-model behaviour (Robinson et al., 2023; Lyu et al., 2024; Wang et al., 2024). Ranking selects among evaluator-supplied continuations, whereas generation additionally requires the model to produce a valid answer independently. We revisit this divergence after a recurrence-depth intervention in Appendix F, where we also report why our own instance of it is weaker evidence than it first appears.

3 Evaluation Protocols

HRM-Text-1B protocol validation. HRM-Text-1B is a PrefixLM that requires a task-conditioned prompt envelope and a `token_type_ids` mask identifying the bidirectional prefix (Wang et al., 2026). We use `synth,cot` for generative mathematics and `direct` for classification. The generic generation interface constructs neither requirement automatically.

At the trained setting $(H{=}2, L{=}3)$ the bare, envelope-only, mask-only and complete protocols obtain 40%, 60%, 70% and 80% accuracy respectively on the same 20 GSM8K examples (Appendix B). On a deterministic seed-42 $n{=}150$ sample the complete protocol obtains 86.0%, close to the published 84.5% result; the corresponding three-seed mean is 84.0%. All subsequent HRM-Text-1B experiments use the complete protocol.

Tasks and prompting. For HRM-Text-1B we evaluate GSM8K (Cobbe et al., 2021) and SVAMP (Patel et al., 2021) through answer generation. GSM8K uses zero-shot `synth,cot` prompting and boxed-answer extraction. The multiple-choice evaluation uses ARC-Challenge (Clark et al., 2018), MMLU (Hendrycks et al., 2021), HellaSwag (Zellers et al., 2019) and BoolQ (Clark et al., 2019). Generation uses 25 ARC validation demonstrations, five MMLU development examples per subject, ten HellaSwag training examples and five BoolQ training examples. Options are represented by letters, and MMLU accuracy is macro-averaged across subjects.

For Huginn-3.5B and the Ouro checkpoints a matched harness holds examples, prompts, decoding and metrics fixed across depths. Mathematical tasks use two-shot chain-of-thought prompting. This differs from the eight-shot GSM8K evaluation originally reported for Huginn-3.5B (Geiping et al.,

2025); its absolute scores are therefore not treated as reproductions of the published headline results. Cross-family comparisons concern depth-response patterns rather than absolute model quality.

Scoring. For generative mathematics an output is correct only when the extracted final value matches the reference; outputs without an extractable answer are invalid and receive zero credit. For multiple-choice generation we report strict accuracy after whitespace and case normalisation. A response must contain a valid option letter matching the gold answer; invalid responses receive zero credit and are reported separately.

The released HRM-Text-1B evaluator instead assigns an invalid answer $1/k$ credit on a question with k options, producing a 25% floor on four-option tasks and a 50% floor on BoolQ. We therefore use strict accuracy and invalid-output rate for the headline generative results.

We additionally apply zero-shot log-likelihood ranking to the same sampled multiple-choice items. This procedure selects the highest-scoring evaluator-supplied continuation rather than requiring the model to generate a valid answer independently (Robinson et al., 2023; Lyu et al., 2024; Wang et al., 2024).

Sampling and depth interventions. All generations use greedy decoding. For each principal HRM-Text-1B seed we evaluate 150 GSM8K examples, 100 SVAMP examples and 300 examples from each multiple-choice benchmark. Principal HRM-Text-1B results aggregate subsample seeds 42–44, which identify deterministic benchmark subsets rather than stochastic decoding runs; "seed" below always refers to these subsets. Every depth setting within a seed uses the same examples.

The Huginn-3.5B sweeps use matched GSM8K and ARC-Challenge subsets at seed 42, over depths from 4 to 128 steps; per-depth values are given in Appendix D. The Ouro sweeps use seed 42 with 50 GSM8K and 150 ARC-Challenge examples at every depth. Because a data seed here selects only which examples are subsampled, we estimate sampling uncertainty by bootstrapping over examples (5,000 resamples, percentile 95% intervals).

All interventions are applied after training. We change only the checkpoint's recurrence setting and, where required, the associated cache allocation; model weights, prompts, examples, decoding and scoring remain fixed. For Ouro, adaptive exit is disabled so that each example executes the requested number of steps. Environment details, exact API arguments and the cache-allocation requirement are given in Appendix I. An anonymised repository containing the evaluation harness, prompts, per-example predictions, trajectory records and analysis scripts accompanies this submission.[1]

State trajectories. We analyse hidden states during prompt prefilling and mean-pool them over sequence positions. For consecutive states z_{t-1} and z_t we report

$$c_t = \frac{z_t^\top z_{t-1}}{\|z_t\|_2 \|z_{t-1}\|_2}, \qquad r_t = \frac{\|z_t - z_{t-1}\|_2}{\|z_t\|_2},$$

where c_t is successive-state cosine similarity and r_t is the current-normalised relative update. Both are computed per prompt and averaged, so every reported value is a mean of per-prompt ratios. We also inspect state norms to distinguish bounded movement from activation explosion or vanishing. All cross-model comparisons use this normalisation.

For the HRM-Text-1B H analysis we record the high-level state after each H step for 12 GSM8K prompts and compare successive states with the trained step-2 endpoint. For the L analysis we record the lower-level state after each invocation on 50 GSM8K prompts, using seed 42, $H{=}2$ and $L \in \{3, 6, 16, 32\}$. The two H phases are analysed separately. We exclude the first within-phase L transition because the updated H state is re-injected immediately before it. Complete per-step trajectories appear in Appendix E.

The same measurements are applied to the single-loop checkpoints. As a non-recurrent control we measure successive Transformer-layer states in Qwen2.5-1.5B (Qwen et al., 2024), testing whether high consecutive-state cosine alone is sufficient evidence of recurrent contraction.

[1] https://anonymous.4open.science/r/Not-All-Recurrence-EA60

Config	GSM8K	SVAMP	ARC-C	MMLU	HeSw	BoolQ
H1L1	0	0	0[a]	0[a]	0[a]	0[a]
H1L3	0	0	0[a]	0[a]	0[a]	0[a]
H2L1	18.2	36.0	36.6	32.0	31.1	68.4
H2L2	79.3	85.7	75.9	53.9	55.0	85.9
H2L3	**84.0**	**88.7**	**84.0**	**60.0**	**64.4**	**86.3**
H2L4[b]	76.0					
H2L6[b]	70.0					
H3L3[b]	52.0					
H4L3	10.0	12.0	26.8	45.6	12.7	40.4
H4L6	3.1	6.7	18.2	39.3	5.2	19.9

Table 1: HRM-Text-1B strict accuracy (%). H2L3 is the trained setting. Unmarked cells are means over seeds 42–44; across-seed standard deviation is at most 4.6 points. [a]All multiple-choice generations are invalid. [b]Seed-42 GSM8K probes with n=50; H2L3 obtains 72.0% on the same subset. Blank cells were not evaluated. HeSw denotes HellaSwag.

H-state intervention. To test whether displacement from the trained H endpoint contributes to the altered-depth failure, we run HRM-Text-1B at H4L3 and replace its final high-level state with

$$z' = \alpha z_{\text{step 4}} + (1 - \alpha) z_{\text{step 2}}$$

for interpolation weights $\alpha \in \{1.00, 0.75, 0.50, 0.25, 0.00\}$. The experiment uses 100 GSM8K examples and a trained-depth baseline on the same examples. Computation following the replacement is unchanged. Because this intervention runs under a local two-shot protocol with a 256-token budget, its absolute accuracies are lower than the official-protocol results reported elsewhere and are compared only within it.

Statistical comparisons. We treat as principal the six-benchmark HRM-Text-1B grid at H2L1–H2L3 and H4L3–H4L6 and the equal-compute contrast; the n=50 seed-42 probes, the 12-prompt H-state analysis, the extended L trajectories and the non-recurrent control are exploratory and are reported as such. This distinction reflects sample size and task coverage, not a pre-registered analysis plan. Comparisons between principal HRM-Text-1B settings are paired at the example level. Paired bootstrap tests compare H2L2 with trained H2L3, and paired sign tests compare correct-to-incorrect with incorrect-to-correct transitions between H2L3 and H4L3.

4 Results

4.1 H and L respond differently to altered depth

Table 1 summarises the evaluated HRM-Text-1B configurations. Unless marked as probes, results are means over seeds 42–44; per-seed values appear in Appendix C.

Reducing L from 3 to 2 lowers accuracy by 0.4–9.4 percentage points across the six benchmarks while reducing recurrent block passes from $2(3 + 1)$=8 to $2(2 + 1)$=6, a 25% reduction. Paired bootstrap tests do not reject equal per-example accuracy on four of the six benchmarks (p=0.15–0.42). This supports the narrower conclusion that H2L2 retains most trained-depth performance; it is not an equivalence result.

On the matched seed-42 GSM8K probe, H2L4 and H2L6 obtain 76.0% and 70.0% respectively, against 72.0% for H2L3. The tested above-training L settings therefore avoid catastrophic degradation, although their single-seed and single-task coverage is limited. Reducing L to 1 causes a substantially larger loss (GSM8K falls to 18.2%) but retains non-zero accuracy on every benchmark. L is therefore tolerant over a window rather than without limit. Only the reduction to L=2 is established across all six benchmarks; the above-training settings rest on the seed-42 GSM8K probe alone.

The H dimension shows no such window over the tested values. At H=1 strict accuracy is zero on all six benchmarks and all multiple-choice generations are invalid. On the matched GSM8K probe, increasing H from 2 to 3 reduces accuracy from 72.0% to 52.0%. In the principal evaluation, increasing H from 2 to 4 reduces GSM8K from 84.0% to 10.0% and SVAMP from 88.7% to 12.0%.

The degradation is strongly asymmetric at the example level. Among GSM8K examples answered correctly at H2L3, 94.6%, 84.9% and 85.4% become incorrect at H4L3 for seeds 42–44 respectively, whereas at most one example per seed changes in the opposite direction. Paired sign tests give $p < 10^{-26}$ for each seed.

The contrast is not explained by total recurrent computation. H1L3 and H2L1 both execute $H(L + 1) = 4$ block passes, yet H1L3 obtains zero strict accuracy on every benchmark while H2L1 remains non-zero on every benchmark. The contrast is not reducible to output formatting: H2L1 executes the same four block passes and produces valid, non-zero output on all six benchmarks, including 68.4% on BoolQ, so four block passes suffice for format when allocated as two H steps and do not when allocated as one. GSM8K, scored by boxed-answer extraction rather than option matching, shows the same pattern. Increasing L from 3 to 6 at $H{=}4$ also fails to restore performance. The allocation and ordering of recurrent computation therefore matter, not only the number of block applications.

4.2 Single-loop checkpoints avoid catastrophic above-reference degradation

Figure 1 compares depth-response curves within each checkpoint; complete numerical sweeps appear in Appendix D.

On seed 42, Huginn-3.5B obtains 12.0% on GSM8K at its mean training depth of 32 steps and 10.7% at both 64 and 128 steps, while ARC-Challenge changes from 41.3% at 32 steps to 44.7% at 64 steps (Figure 1c). Additional unrolling therefore produces no collapse comparable to the HRM-Text-1B H intervention over the evaluated range. We note that Huginn-3.5B's GSM8K accuracy under our two-shot harness is low in absolute terms, so ARC-Challenge carries more of this comparison than GSM8K does.

For Ouro-1.4B, GSM8K changes from 56.0% at four steps to 60.0% at eight, while ARC-Challenge changes from 90.7% to 94.0%. Ouro-2.6B is non-monotonic but remains near its four-step performance: GSM8K changes from 70.0% to 64.0% and ARC-Challenge from 92.7% to 91.3%. Invalid-output rates are at most 2%, and exactly zero in every Ouro cell across all 1,200 generations per model.

Bootstrap intervals over examples place every above-reference result inside the trained-depth interval and every one-step result below it (all 36 cells in Appendix G). The above-reference changes are therefore stable-to-improving rather than established gains, while the reduction below the reference depth is genuine. These observations establish that catastrophic above-reference degradation is not shared by all tested recurrent checkpoints; they do not identify which architectural or training difference is responsible.

4.3 State trajectories distinguish endpoint displacement from recurrent contraction

H-state displacement. For 12 GSM8K prompts the mean cosine similarity between the first and second trained H states is 0.19, indicating that the two steps do not behave as repeated small refinements. This analysis uses the smallest sample in the paper and we report no dispersion for it; the claims drawn from it should be read accordingly. At $H{=}4$, successive-state cosines for the third and fourth H states are 0.92 and 0.96, but their similarity to the trained step-2 endpoint decreases as $1.00 \rightarrow 0.92 \rightarrow 0.82$. State norms remain stable, so the failure is not explained by activation explosion or vanishing. The trajectory is consistent with an endpoint-sensitive computation whose useful state is associated with the trained H count.

The displacement is causally implicated. Replacing the final H4 state with $z' = \alpha z_{\text{step}\,4} + (1 - \alpha) z_{\text{step}\,2}$ produces graded recovery on 100 GSM8K examples:

α	1.00	0.75	0.50	0.25	0.00
Accuracy (%)	3.0	31.0	63.0	66.0	**71.0**

The trained H2L3 configuration obtains 71.0% on the same 100 examples. Both the baseline and the intervention run under the local two-shot protocol of Section 3, so their absolute values sit below the official-protocol numbers in Table 1; all comparisons here are within-protocol. The $\alpha{=}0$ endpoint is

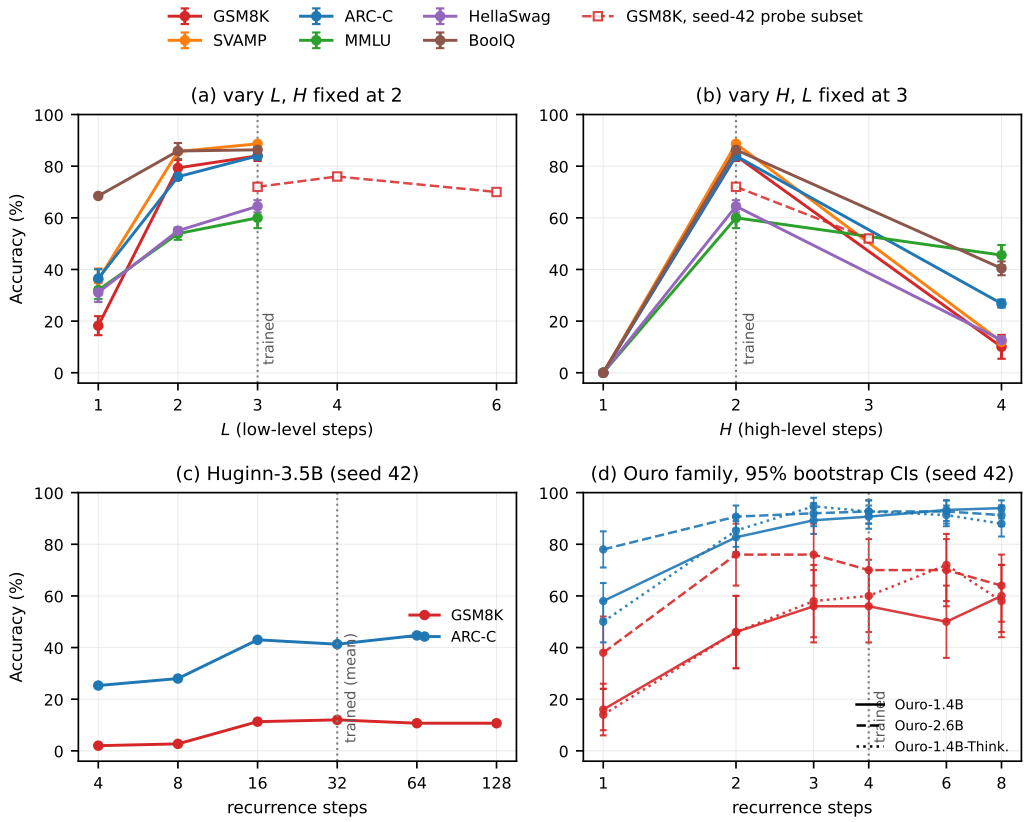


Figure 1: Depth response, separated by recurrence dimension. **(a)** Varying L with H held at its trained value of 2 gives a broad tolerant region: accuracy is close to trained between $L{=}2$ and $L{=}3$ and falls only at $L{=}1$. **(b)** Varying H with L held at 3 gives a sharp peak at the trained $H{=}2$ and collapse on both sides. Filled markers are means over seeds 42–44 with across-seed standard deviation; open squares are the seed-42 $n{=}50$ GSM8K probe, plotted against *its own* trained-depth value of 72.0% rather than the principal 84.0%, because it uses a different subset. **(c)** Huginn-3.5B and **(d)** the Ouro family under the matched harness at seed 42, with bootstrap 95% intervals in (d). Panels compare within-checkpoint depth responses, not absolute model quality.

close to definitional (substituting the trained final H state should reproduce trained-depth behaviour if the readout depends on that state), so we do not treat it as the finding. Two things are informative. First, most of the recovery arrives well before $\alpha{=}0$: of the 68-point gap between $\alpha{=}1$ and $\alpha{=}0$, about 88% has closed by $\alpha{=}0.50$, so the damage is graded in displacement magnitude rather than switching mode at a threshold. Second, that replacing this single state suffices indicates that the high-level representation mediates much of the observed failure, though it does not by itself establish that the L-loop states and cache that also differ at $H{=}4$ are irrelevant. It does not determine whether additional H steps destroy task-relevant information or preserve it in a representation that subsequent computation cannot use, and it examines only the one direction connecting the observed step-4 and step-2 states.

Phase-conditioned L dynamics. The two H phases induce different trajectories in the same L module. During the first phase, successive-state cosine increases from 0.493 at step 2 to 0.9990 at step 32 while the relative update decreases from 1.284 to 0.0357, with state norms near 19.9–20.1. This trajectory contracts toward a fixed-point-like endpoint over the measured horizon.

During the second phase the mean-pooled trajectory instead exhibits a persistent three-step pattern. Near the end of the measured horizon relative updates repeat approximately as $0.464, 0.440, 0.350$, with corresponding cosine similarities $0.913, 0.898, 0.938$, and state norms bounded between ap-

proximately 18.3 and 20.9. This is evidence of a persistent three-step pattern in the mean-pooled trajectory, not proof that every token follows an exact period-3 orbit.

At the trained value $L{=}3$ neither extended pattern is yet visible: the final relative updates are 0.633 ± 0.073 and 0.437 ± 0.025 for the first and second H phases. A trajectory measured only at the trained depth would not distinguish the two regimes.

Cross-model trajectory comparison. Under the same normalisation the first HRM-Text-1B L phase (0.0357, 0.9990 at step 32) sits between Huginn-3.5B (0.0084, 0.9999) and Ouro-1.4B (0.1466, 0.9881), whose late updates are non-monotone; the second HRM-Text-1B L phase retains larger one-step updates while staying bounded, so persistent updates do not by themselves establish divergence. Cyclic recurrent structure has been reported in related looped models (Blayney et al., 2026). As a non-recurrent control, Qwen2.5-1.5B reaches cosine 0.999 in some intermediate layers yet ends at 0.736 and 0.262 with relative updates of 0.943–1.624, so high cosine at an isolated transition is not evidence of recurrent contraction. Full values are in Appendix E.

4.4 An additional consistency case

Ouro-1.4B-Thinking, whose trajectory contracts through eight steps ($r_8{=}0.0594$, $c_8{=}0.9984$), stays within 4.7 points of its four-step reference on both tasks and produces no invalid generation at any depth. It is consistent with the trajectory-based reading but does not establish a prediction rule; details are in Appendix H.

5 Validating a Recurrence-Depth Compute Dial

Our measurements suggest four checks before recurrence depth is treated as an inference-time compute control. Each is derived from a contrast we observed; none is validated here as a general predictor.

- **Sweep each dimension separately, below and above its trained value.** The H/L contrast is invisible to a sweep over total depth, and the two dimensions have different tolerance windows.
- **Do not infer robustness from total recurrent compute.** H1L3 and H2L1 execute the same four block passes; one scores zero everywhere and the other does not.
- **Score generatively and report invalid-output rate.** Format and accuracy breakdown co-occurred for the depth-sensitive dimension, while depth-tolerant Ouro settings produced no invalid output, so this is a cheap canary needing no extra inference.
- **Measure the trajectory well past the trained depth.** At $L{=}3$ the two H-conditioned phases are indistinguishable in kind; the contracting and periodic regimes separate only by $L{=}32$.

6 Discussion

Recurrence depth is a learned operating condition. Parameter sharing makes recurrence depth adjustable in implementation but does not guarantee that a checkpoint remains reliable when that depth changes. The H/L contrast in HRM-Text-1B provides the clearest evidence: L is comparatively tolerant over the tested range, whereas changing H either below or above its trained value produces severe degradation. The equal-compute comparison further shows that recurrent block count alone is insufficient; the allocation and ordering of those applications matter.

The tested Huginn-3.5B and Ouro checkpoints show no comparable catastrophic degradation over the evaluated above-reference depths, although their responses are sometimes non-monotonic. Because the families differ in architecture, scale, data, objectives, state interactions and implementation, this comparison does not identify which design choice causes the difference. It establishes only that post-training depth sensitivity varies substantially across trained checkpoints.

Endpoint-sensitive and phase-conditioned dynamics. The HRM-Text-1B H results are consistent with *program-like* recurrence: successive H steps appear to occupy distinct computational roles,

and the useful representation is associated with the trained endpoint. Performance fails when H is shortened or extended, additional H steps move the state away from the trained endpoint, and interpolating back toward that endpoint recovers most of the lost accuracy in a graded way. Several explanations remain compatible with the evidence: additional H steps may remove task-relevant information, preserve it in a representation that subsequent computation cannot use, or alter only a subset of token-level features required by the readout.

The first H-conditioned L trajectory is instead *refinement-like*, contracting toward a fixed-point-like endpoint, while the second exhibits a persistent three-step pattern within bounded norm, so persistent one-step movement does not by itself imply divergence, as related work on cyclic recurrent structure also finds (Yang et al., 2026; Blayney et al., 2026). These two labels are empirical descriptions of measured trajectories, not a taxonomy. That the same L weights behave differently under different H states suggests the object of analysis is a *conditioned* trajectory rather than a module in isolation.

7 Conclusion

Recurrence depth is not a uniform test-time compute control, even within a single checkpoint. In HRM-Text-1B the low-level L dimension is comparatively tolerant over the tested range while the high-level H dimension degrades severely in both directions, and configurations with the same number of recurrent block applications yield markedly different outcomes depending on how those applications are organised. The state analyses offer a candidate explanation: additional H steps displace the representation from its trained endpoint, and interpolating back recovers accuracy in a graded way, indicating that the high-level state mediates much of the observed failure. The L loop behaves differently again, contracting under one H phase and following a bounded periodic pattern under the other, so tolerance to added recurrence does not require every phase to converge. Parameter sharing makes depth adjustable in implementation; whether the adjustment preserves usable behaviour must be established empirically, per checkpoint and per dimension. Limitations of the study are discussed in Appendix A.

AI Use Statement

We used generative AI assistants for drafting and editing the manuscript text and for LaTeX formatting support. All research design, experiments, analysis and claims are the authors' own; the authors reviewed and verified all AI-assisted content and take full responsibility for this work.

Ethics Statement

This work analyses publicly released model checkpoints on public benchmarks and introduces no new model, dataset or deployment. The practical risk it identifies runs in the opposite direction from misuse: treating recurrence depth as a compute dial without per-dimension validation can reduce a deployed system's accuracy and output validity with no error raised and no signal that anything has changed. In our experiments the failure was silent under one common evaluation procedure and obvious under another. The validation protocol in Section 5 is intended to mitigate exactly that.

Reproducibility Statement

All experiments use publicly released checkpoints and public benchmarks. An anonymised repository (linked in Section 3) contains the evaluation harness, prompts, per-example predictions, trajectory records and analysis scripts. The HRM-Text-1B inference protocol is specified in Section 3 and ablated in Appendix B; sample sizes, subsample seeds, decoding and scoring rules are given in Section 3. Library versions, hardware and the exact configuration arguments used to change recurrence depth, including the cache-allocation adjustment, are listed in Appendix I. Per-seed results, complete depth sweeps, bootstrap intervals and full trajectory values appear in Appendices C, D, G and E.

REFERENCES

ARC Prize Foundation. HRM analysis. https://arcprize.org/blog/hrm-analysis, 2025. Accessed: 2026-08-03; archived at https://web.archive.org/web/20260622171002/https://arcprize.org/blog/hrm-analysis.

S. Bae, Y. Kim, R. Bayat, S. Kim, J. Ha, T. Schuster, A. Fisch, H. Harutyunyan, Z. Ji, A. Courville, and S.-Y. Yun. Mixture-of-Recursions: Learning dynamic recursive depths for adaptive token-level computation. arXiv preprint arXiv:2507.10524, 2025.

S. Bai, J. Z. Kolter, and V. Koltun. Deep equilibrium models. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

A. Banino, J. Balaguer, and C. Blundell. PonderNet: Learning to ponder. In *Proceedings of the 8th ICML Workshop on Automated Machine Learning*, 2021.

A. Bansal, A. Schwarzschild, E. Borgnia, Z. Emam, F. Huang, M. Goldblum, and T. Goldstein. End-to-end algorithm synthesis with recurrent networks: Logical extrapolation without overthinking. In *Advances in Neural Information Processing Systems*, volume 35, pp. 20232–20242, 2022.

H. Blayney, Á. Arroyo, J. Obando-Ceron, P. S. Castro, A. Courville, M. M. Bronstein, and X. Dong. A mechanistic analysis of looped reasoning language models. arXiv preprint arXiv:2604.11791, 2026.

C. Clark, K. Lee, M.-W. Chang, T. Kwiatkowski, M. Collins, and K. Toutanova. BoolQ: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2924–2936. Association for Computational Linguistics, 2019.

P. Clark, I. Cowhey, O. Etzioni, T. Khot, A. Sabharwal, C. Schoenick, and O. Tafjord. Think you have solved question answering? Try ARC, the AI2 reasoning challenge. arXiv preprint arXiv:1803.05457, 2018.

K. Cobbe, V. Kosaraju, M. Bavarian, M. Chen, H. Jun, L. Kaiser, M. Plappert, J. Tworek, J. Hilton, R. Nakano, C. Hesse, and J. Schulman. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168, 2021.

M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and Ł. Kaiser. Universal transformers. In *International Conference on Learning Representations*, 2019.

M. Elbayad, J. Gu, E. Grave, and M. Auli. Depth-adaptive transformer. arXiv preprint arXiv:1910.10073, 2019.

J. Geiping, S. McLeish, N. Jain, J. Kirchenbauer, S. Singh, B. R. Bartoldson, B. Kailkhura, A. Bhatele, and T. Goldstein. Scaling up test-time compute with latent reasoning: A recurrent depth approach. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

A. Giannou, S. Rajput, J.-Y. Sohn, K. Lee, J. D. Lee, and D. Papailiopoulos. Looped transformers as programmable computers. In *Proceedings of the 40th International Conference on Machine Learning*, 2023.

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

D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

A. Jeddi, M. Ciccone, and B. Taati. LoopFormer: Elastic-depth looped transformers for latent reasoning via shortcut modulation. arXiv preprint arXiv:2602.11451, 2026.

H. Kohli, S. Parthasarathy, H. Sun, and Y. Yao. Loop, think, & generalize: Implicit reasoning in recurrent-depth transformers. arXiv preprint arXiv:2604.07822, 2026.

W. Lu, Y. Yang, K. Lee, Y. Li, and E. Liu. Latent chain-of-thought? Decoding the depth-recurrent transformer. arXiv preprint arXiv:2507.02199, 2025.

C. Lyu, M. Wu, and A. Aji. Beyond probabilities: Unveiling the misalignment in evaluating large language models. In *Proceedings of the 1st Workshop on Towards Knowledgeable Language Models (KnowLLM 2024)*, pp. 109–131. Association for Computational Linguistics, 2024.

A. Mohtashami, M. Pagliardini, and M. Jaggi. CoTFormer: A chain-of-thought driven architecture with budget-adaptive computation cost at inference. In *International Conference on Learning Representations*, 2025.

A. Patel, S. Bhattamishra, and N. Goyal. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2080–2094. Association for Computational Linguistics, 2021.

Qwen, A. Yang, B. Yang, B. Zhang, B. Hui, B. Zheng, B. Yu, C. Li, D. Liu, F. Huang, H. Wei, et al. Qwen2.5 technical report. arXiv preprint arXiv:2412.15115, 2024.

J. Robinson, C. M. Rytting, and D. Wingate. Leveraging large language models for multiple choice question answering. In *International Conference on Learning Representations*, 2023.

T. Schuster, A. Fisch, J. Gupta, M. Dehghani, D. Bahri, V. Q. Tran, Y. Tay, and D. Metzler. Confident adaptive language modeling. arXiv preprint arXiv:2207.07061, 2022.

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

C. Snell, J. Lee, K. Xu, and A. Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. In *International Conference on Learning Representations*, 2025.

G. Wang, J. Li, Y. Sun, X. Chen, C. Liu, Y. Wu, M. Lu, S. Song, and Y. Abbasi Yadkori. Hierarchical reasoning model. arXiv preprint arXiv:2506.21734, 2025.

G. Wang, C. Liu, C. Wang, C. Zhou, Y. Sun, Y. Wu, S. Zhen, L. Scimeca, and Y. Abbasi Yadkori. HRM-Text: Efficient pretraining beyond scaling. arXiv preprint arXiv:2605.20613, 2026.

X. Wang, B. Ma, C. Hu, L. Weber-Genzel, P. Röttger, F. Kreuter, D. Hovy, and B. Plank. "My answer is C": First-token probabilities do not match text answers in instruction-tuned language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 7407–7416. Association for Computational Linguistics, 2024.

J. Xin, R. Tang, J. Lee, Y. Yu, and J. Lin. DeeBERT: Dynamic early exiting for accelerating BERT inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 2020.

K. Xu and I. Sato. On expressive power of looped transformers: Theoretical analysis and enhancement via timestep encoding. In *Proceedings of the 42nd International Conference on Machine Learning*, 2025.

X.-W. Yang, Z. Han, X.-H. Zhang, W.-D. Wei, J.-J. Shao, L.-Z. Guo, and Y.-F. Li. Stabilizing recurrent dynamics for test-time scalable latent reasoning in looped language models. arXiv preprint arXiv:2605.26733, 2026.

R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800. Association for Computational Linguistics, 2019.

R.-J. Zhu, Z. Wang, K. Hua, T. Zhang, Z. Li, H. Que, B. Wei, Z. Wen, F. Yin, H. Xing, et al. Scaling latent reasoning via looped language models. arXiv preprint arXiv:2510.25741, 2025.

A LIMITATIONS

One coupled architecture. The study evaluates five checkpoints from three model families, but only HRM-Text-1B contains coupled H and L recurrence, and it is also the only checkpoint showing catastrophic depth sensitivity. Our central result, that recurrence dimensions within one checkpoint can respond very differently, therefore rests on a single architecture. We chose the within-checkpoint design precisely because no cross-model comparison can control architecture, scale, data and training schedule at once; the cost of that choice is that the central result is demonstrated on the one public checkpoint that has two recurrence dimensions. The results do not establish that hierarchical recurrence is generally brittle or that single-loop recurrence is generally robust, and no larger coupled recurrent language model is publicly available for us to test.

Cross-family comparisons are observational. The three families differ in architecture, scale, training data, objectives and implementation. We report depth-response patterns under a matched harness, but we cannot attribute the difference in depth sensitivity to any one design choice.

Uneven experimental coverage. The principal HRM-Text-1B behavioural results aggregate three deterministic benchmark subsets, whereas the above-training L probes and the Ouro and Huginn-3.5B sweeps use one seed. The H-state analysis uses 12 GSM8K prompts and reports no uncertainty; the interpolation uses 100 examples; the extended L analysis uses 50 prompts from one seed. The Ouro sweeps are accompanied by bootstrap intervals over examples, but the H-state results are not. These state-level results provide focused evidence for the reported checkpoints rather than population-level estimates.

Absolute scores are not reproductions. Our matched cross-model harness uses two-shot chain-of-thought prompting for mathematics, unlike the eight-shot GSM8K evaluation originally reported for Huginn-3.5B. Huginn-3.5B's absolute GSM8K accuracy under our harness is consequently low, which limits how much that benchmark can contribute to an absence-of-degradation claim; ARC-Challenge carries more of that comparison.

Altered-depth execution is not separately validated. Running HRM-Text-1B away from its trained depth requires adjusting the recurrent cache allocation (Appendix I); without that adjustment the model fails at run time. The depth mechanism itself is the native `transformers` implementation, which reads the recurrence counts at every forward pass, and the cache-allocation formula was checked against that implementation (scripts in the released repository). We do not, however, report a bit-for-bit comparison of trained-depth inference with and without the adjusted allocation, so we cannot fully exclude that cache-indexing semantics contribute to the measured degradation.

Two observations bound how much they could contribute. First, H2L1 runs through the same generalised path at a non-trained setting and produces valid, non-zero output on all six benchmarks, so that path is not globally broken away from the trained configuration. Second, and more directly, the interpolation of Section 4.3 runs entirely under the $H{=}4$ configuration, and replacing only the final H state with the step-2 state recovers exactly the trained-depth accuracy on the same examples. Had the $H{=}4$ cache allocation corrupted execution, that corruption would persist through the readout and the substitution could not restore trained-level accuracy; the step-2 state captured under $H{=}4$ and everything downstream of it therefore behave as they do natively. Any execution artifact would have to be confined to the third and fourth H steps specifically. That is a narrower possibility than an unsound generalised path, but it is not excluded, and an equivalence check of the native and generalised paths at H2L3 remains the direct test.

The intervention tests one direction. The interpolation examines only the direction connecting the observed step-4 and step-2 states. It does not characterise the surrounding representation space, and we did not run a norm-matched random-direction control, so we cannot exclude that perturbations of comparable magnitude along other directions would also change accuracy. We therefore describe the displacement as causally implicated rather than as the sole cause.

Trajectory analyses use mean-pooled prefill states. Mean pooling can obscure token- or layer-specific behaviour, and the observed three-step pattern does not imply that every token follows the

same periodic trajectory. We do not test whether the pattern persists during autoregressive decoding, under other tasks or seeds, or after state perturbations. Extending L to 32 reveals behaviour invisible at the trained depth of three, but those extrapolated depths are not claimed to be useful operating points, and bounded state norms do not guarantee preserved task information.

The evaluation-procedure comparison is not conclusive. As set out in Appendix F, likelihood ranking sits close to chance at trained depth on three of four multiple-choice benchmarks, and our generative and ranking procedures differ in demonstration setting. We therefore do not claim to have shown that option ranking conceals depth-induced failure in general; prompt-matched ranking would be required to attribute the divergence to the scoring rule.

One consistency case, not a validated predictor. Ouro-1.4B-Thinking appears in the behavioural sweep as well as in the trajectory analysis, so we describe it as an additional consistency case rather than a held-out test. Broader conclusions would require additional architectures, scales, tasks, seeds, token-resolved analyses, prompt-matched generation and ranking, and interventions that distinguish information loss from downstream representation mismatch.

B Protocol Ablation

Table 2 isolates the contribution of each component of the HRM-Text-1B inference protocol at the trained setting (H=2, L=3).

Setup	Envelope+tags	Prefix mask	GSM8K
bare prompt	No	No	40
envelope only	Yes	No	60
mask only	No	Yes	70
complete	Yes	Yes	**80**

Table 2: Protocol ablation at trained depth (GSM8K strict accuracy %, n=20, seed 42, greedy decoding). On the larger seed-42 n=150 sample the complete protocol obtains 86.0%, against a published 84.5%.

C Per-Seed HRM-Text-1B Results

Table 3 gives every cell of Table 1 separately for seeds 42, 43 and 44. The across-seed standard deviation never exceeds 4.6 points; its maximum, 4.62, occurs for GSM8K at H4L3.

Seed	Benchmark	H1L1	H1L3	H2L1	H2L2	**H2L3**	H4L3	H4L6
42	GSM8K	0	0	20.7	82.0	**86.0**	4.7	1.3
42	SVAMP	0	0	41.0	85.0	**89.0**	14.0	9.0
42	ARC-C	0	0	40.3	77.0	**83.3**	27.7	17.7
42	MMLU	0	0	33.7	56.0	**60.0**	44.0	35.3
42	HellaSwag	0	0	35.3	56.0	**66.7**	12.7	6.7
42	BoolQ	0	0	68.7	82.3	**85.0**	43.3	22.0
43	GSM8K	0	0	20.0	80.0	**84.0**	12.7	4.7
43	SVAMP	0	0	33.0	86.0	**88.0**	12.0	5.0
43	ARC-C	0	0	36.0	74.7	**83.3**	27.7	16.7
43	MMLU	0	0	28.0	51.3	**56.0**	42.7	43.7
43	HellaSwag	0	0	29.7	55.0	**64.7**	12.0	4.7
43	BoolQ	0	0	68.0	88.0	**86.7**	40.0	20.0
44	GSM8K	0	0	14.0	76.0	**82.0**	12.7	3.3
44	SVAMP	0	0	34.0	86.0	**89.0**	10.0	6.0
44	ARC-C	0	0	33.3	76.0	**85.3**	25.0	20.3
44	MMLU	0	0	34.3	54.3	**64.0**	50.0	39.0
44	HellaSwag	0	0	28.3	54.0	**62.0**	13.3	4.3
44	BoolQ	0	0	68.7	87.3	**87.3**	38.0	17.7

Table 3: Per-seed HRM-Text-1B strict accuracy (%) under the complete protocol. H2L3 is the trained setting. All multiple-choice generations are invalid at $H{=}1$.

D Complete Single-Loop Depth Sweeps

Huginn-3.5B	recurrence steps					
	4	8	16	**32**	64	128
GSM8K	2.0	2.7	11.3	**12.0**	10.7	10.7
ARC-C	25.3	28.0	43.0	**41.3**	44.7	
Ouro-1.4B	recurrence steps					
	1	2	3	**4**	6	8
GSM8K	16.0	46.0	56.0	**56.0**	50.0	60.0
ARC-C	58.0	82.7	89.3	**90.7**	93.3	94.0
Ouro-2.6B	recurrence steps					
	1	2	3	**4**	6	8
GSM8K	38.0	76.0	76.0	**70.0**	70.0	64.0
ARC-C	78.0	90.7	92.0	**92.7**	92.7	91.3
Ouro-1.4B-Thinking	recurrence steps					
	1	2	3	**4**	6	8
GSM8K	14.0	46.0	58.0	**60.0**	72.0	58.0
ARC-C	50.0	85.3	94.7	**92.7**	91.3	88.0

Table 4: Single-loop depth sweeps (strict accuracy %, seed 42). Reference depth in bold; the blank cell was not evaluated.

E Complete L-Loop Trajectories

Both quantities are computed per prompt and then averaged over 50 GSM8K prompts, so every entry is a mean of per-prompt ratios.

L-step	2	3	4	6	8	12	16	24	32
c_t	.493	.834	.953	.980	.993	.998	.9986	.9989	.9990
r_t	1.284	.633	.303	.198	.113	.057	.046	.037	.036

Table 5: First H phase: monotone contraction toward a fixed-point-like endpoint, flattening from about step 24. State norms stay within 19.9–20.1.

L-step	2	3	4	5	6	7	29	30	31	32
r_t	.461	.433	.346	.464	.440	.350	.464	.440	.350	.464
c_t	.913	.901	.939	.913	.898	.937	.913	.898	.938	.913
norm	18.5	20.0	20.8	18.4	20.0	20.8	18.3	20.0	20.8	18.4

Table 6: Second H phase: a persistent three-step pattern repeating across the measured horizon, with norms bounded in 18.3–20.9. Steps 8–28 continue the same pattern and are omitted for space.

Configuration	phase 1 r_{final}	phase 2 r_{final}
L=3 (trained)	0.633 ± 0.073	0.437 ± 0.025
L=6	0.198 ± 0.064	0.440 ± 0.020
L=16	0.046 ± 0.033	0.350 ± 0.023
L=32	0.036 ± 0.032	0.464 ± 0.036

Table 7: Final-step relative update by loop length, mean $\pm$ standard deviation over 50 prompts.

Trajectory	step	r_t	c_t
Huginn-3.5B	32	0.0084	0.9999
Ouro-1.4B-Thinking	8	0.0594	0.9984
HRM-Text-1B L loop, phase 1	32	0.0357	0.9990
Ouro-1.4B	8	0.1466	0.9881
HRM-Text-1B L loop, phase 2	32	0.4639	0.9128
Qwen2.5-1.5B, last two layers	28	0.943–1.624	0.736 / 0.262

Table 8: Final-step values under one normalisation, so that every row is directly comparable. Ouro-1.4B's late updates are non-monotone: 0.182, 0.187, 0.092, 0.147 over its last four steps, so its step-7 value of 0.092 is not its endpoint.

F Evaluation Procedure and Altered-Depth Failure

Table 9 compares model-specific generation with zero-shot log-likelihood ranking on the same sampled multiple-choice items. Changing from H2L3 to H4L3 reduces generative strict accuracy by 14.4–57.2 points and produces substantial invalid-output rates, whereas no ranking score decreases by more than 0.6 points and BoolQ increases by 7.2 points.

	Generative			LL ranking	
Benchmark	H2L3	H4L3	Invalid	H2L3	H4L3
ARC-C	84.0	26.8	59.6	38.8	39.3
MMLU	60.0	45.6	14.0	34.3	33.8
HellaSwag	64.4	12.7	71.0	37.8	37.2
BoolQ	86.3	40.4	47.0	73.8	81.0

Table 9: Generative strict accuracy, generative invalid-output rate and zero-shot log-likelihood ranking accuracy (%), averaged over seeds 42–44. Chance is 25% on ARC-C, MMLU and HellaSwag, and 50% on BoolQ.

We report this as an observation rather than a contribution, because two features of our setup limit what it can establish. First, ranking accuracy at the *trained* depth is 38.8, 34.3 and 37.8 on ARC-C, MMLU and HellaSwag against a 25% chance level, so ranking recovers little of the model's trained-depth ability before any intervention is applied. Its stability across the intervention therefore does not by itself show that ranking conceals a failure it would otherwise have detected. Second, our generative and ranking procedures differ in demonstration setting (generation is few-shot, ranking zero-shot), so the comparison does not isolate the scoring rule.

BoolQ is the exception, and the one case we consider informative: ranking there is clearly above its 50% chance level at trained depth (73.8%), and under the depth intervention the ranking score *rises* by 7.2 points while generative accuracy falls by 45.9 points and 47.0% of generations are invalid. On that benchmark the two procedures move in opposite directions.

The defensible reading is narrow. When deployment requires the model to produce an answer independently, generated accuracy and invalid-output rate register a change that option ranking does not, and on the one benchmark where ranking is clearly above chance the two disagree in sign. Whether prompt-matched ranking would show the same divergence is untested here and would be needed to attribute the effect to the scoring rule.

G Bootstrap Intervals for the Ouro Sweeps

Steps	Ouro-1.4B	Ouro-2.6B	Ouro-1.4B-Think.
GSM8K (n=50)			
1	16 [8, 26]	38 [24, 52]	14 [6, 24]
2	46 [32, 60]	76 [64, 88]	46 [32, 60]
3	56 [42, 70]	76 [64, 88]	58 [44, 72]
4	**56 [42, 70]**	**70 [56, 82]**	**60 [46, 74]**
6	50 [36, 64]	70 [56, 82]	72 [58, 84]
8	60 [46, 72]	64 [50, 76]	58 [44, 72]
ARC-Challenge (n=150)			
1	58 [51, 65]	78 [71, 85]	50 [42, 58]
2	82.7 [77, 89]	90.7 [86, 95]	85.3 [79, 91]
3	89.3 [84, 94]	92.0 [87, 96]	94.7 [91, 98]
4	**90.7 [86, 95]**	**92.7 [88, 97]**	**92.7 [88, 97]**
6	93.3 [89, 97]	92.7 [88, 97]	91.3 [87, 95]
8	94.0 [90, 97]	91.3 [87, 95]	88.0 [83, 93]

Table 10: Ouro accuracy (%) with bootstrap percentile 95% intervals over examples (5,000 resamples, seed 42). Depth 4 is the trained maximum recurrent depth. Every interval at six and eight steps overlaps the depth-4 interval; every one-step interval lies below it.

H Additional Consistency Case

Ouro-1.4B-Thinking provides an additional consistency case: its eight-step trajectory is analysed separately from its behavioural depth sweep, in which it also appears, so it is not a held-out test. Its trajectory contracts through eight steps, with r_8=0.0594 and c_8=0.9984. Its eight-step performance remains within 4.7 points of the four-step reference on both tasks: GSM8K changes from 60.0% to 58.0% and ARC-Challenge from 92.7% to 88.0%. No generation is invalid at any evaluated depth.

Above the reference depth, performance is stable to improving rather than degrading, although not on ARC-Challenge. At six steps GSM8K reaches 72.0% while ARC-Challenge remains near its four-step result at 91.3%; with 50 GSM8K examples these differences lie within sampling noise. This checkpoint is consistent with the trajectory-based interpretation but does not establish a general prediction rule.

I Environment and Depth Manipulation

All experiments use greedy decoding, bfloat16 and released public checkpoints. Official-protocol evaluations run on an A100 GPU; the state-trajectory and H-state intervention experiments run locally on an RTX 4060.

Version constraints are checkpoint-specific: HRM-Text-1B requires `transformers` 5.9 or later, which provides the native `hrm_text` implementation; Huginn-3.5B requires `transformers` 4.49 and `torch` 2.5 or later; Ouro requires `transformers` 4.56, whose custom cache code is incompatible with both older and newer versions.

For HRM-Text-1B we set `H_cycles` and `L_cycles` in the model configuration, which the forward pass reads at every call. We also set `num_hidden_layers` $= s\,H(L+1)$ with $s = 16$ per-stack layers, so the recurrent KV cache allocates one slot per attention invocation; without this adjustment, off-trained depths fail at run time rather than producing degraded output. For Huginn-3.5B we pass `num_steps` to `generate`. For Ouro we set `total_ut_steps` with adaptive early exit disabled.

The n=30 pilot of the H-state interpolation, superseded by the n=100 run reported in Section 4.3, obtained 10.0, 40.0, 60.0, 63.3 and 73.3% at $\alpha = 1.00, 0.75, 0.50, 0.25, 0.00$.