LOOPED TRANSFORMERS LEARN ONE OF TWO ALGORITHMS, AND TRAINING MOVES THE ODDS

Anonymous authors
Paper under double-blind review

ABSTRACT

A looped transformer applies the same block many times over. On a task with a known depth requirement, computing every running product of a sequence of group elements, such a model could in principle work in logarithmic depth. Trained on one, it instead reads the sequence one element per loop, and a growing literature asks what would make it do otherwise. We answer that by intervening rather than by timing: overwriting the model's internal state at one position and one loop, and asking which outputs change. Doing this across two independent sets of checkpoints, 237 and 90, shows that these models do not learn a spectrum of more or less efficient circuits. They learn one of **two** algorithms, and the distribution of their measured speed is bimodal, with a slow mode sitting exactly at one position per loop and a fast mode near three. The structure survives a test that assumes no parametric form, and survives refitting on one checkpoint per training contract. **Training therefore controls the odds of getting the better algorithm, not the algorithm's quality.** Making the efficient one *available* does not move those odds: un-tying the weights leaves the median slope at exactly 1.00, as does conditioning each loop on its index, as does supervising the efficient circuit's own intermediate quantities, which hands the model the target computation. Only making the slow algorithm infeasible moves them. Since the achievable speed varies from seed to seed while each model faces a hard threshold, the population curve is smooth: we registered a sharp critical budget, found none, and report that as a failure. Holding the depth budget proportional to length, reach does not fall from n=64 to n=128, so length is not the binding variable; capacity is, rising from 17 of 40 solved at 5M parameters to 11 of 12 at 50M ($p = 0.003$). It does not buy speed. What it buys is fidelity at matched depth and a shift in which algorithm is installed, and we measure both. This is one family of synthetic tasks, models up to 50M parameters, and a causal measurement that reaches $n \leq 32$. A frontier is only measurable against a known depth requirement, which natural text does not supply; on the two public looped LMs we can therefore only test behaviour, and there depth from 1 to 64 does not move a two-composition ceiling. Within that scope every claim was registered before the run that tests it, which is what makes the negative results usable: the capability result is believable because it was a prediction rather than a summary. Every registered prediction is scored in the appendix, including those that failed.

1 INTRODUCTION

A weight-tied transformer applied T times can in principle compute a prefix product in logarithmic depth. Trained on one, it does not. It learns a literal one-position-per-loop scan, and a large recent literature asks what would make it do otherwise (Geiping et al., 2025; Saunshi et al., 2025; Schwarzschild et al., 2021).

Our evidence is causal. We overwrite the state at one position and one loop and record which outputs change, across two independent sets of checkpoints, 237 and a later 90. Neither gives back a spectrum. **The distribution of frontier speed is bimodal.** On the 237, measured before we had the hypothesis, a two-component mixture beats a unimodal fit by 78 BIC, with modes at $v = 1.03$, the literal scan,

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.

and $v = 2.56$, a compressed front. The later 90, trained under a different contract, reproduce it, their slow mode again at the scan ($v = 1.16$) and their fast mode close to the earlier one without matching it ($v = 2.95$). There is no continuum between them.

That reframes what training controls. When the outcome is one of two algorithms, no budget, curriculum or capacity increase can make it better; all they can change is *which one you get*. So we asked what moves those odds, and almost nothing does. Un-tying the weights leaves the median slope at exactly 1.00, and so does conditioning each loop on its index, and so does supervising the parallel-prefix circuit's own intermediate quantities, which amounts to handing the model the target computation. Only denying the depth a scan needs moves the odds. Even then what appears is the compressed front, not the textbook tree.

Achievable speed varies continuously across seeds while each model faces a hard threshold, so the population curve is smooth. We registered a step function in the depth budget and there is none, since model selection prefers a logistic over a free-knee step by 8.7 to 10.8 BIC. We report that as a failed prediction, and the resolution, that a sharp per-model constraint integrates to a smooth population curve, is offered as an interpretation rather than a confirmed result, because its own held-out test failed (§11).

Two further control variables behave very differently from each other. **Length does not bind**: holding the covered-fraction demand constant, reach does not fall from n=64 to n=128. **Capacity does**: reach rises from 17 of 40 at 5M parameters to 11 of 12 at 50M ($p = 0.003$). It does not buy causal speed. It buys fidelity at matched depth, and it changes which algorithm is installed.

Contributions. **(1)** Depth-recurrent transformers on this task install one of two discrete algorithms: the split replicates across independent checkpoint sets, and demanding a speed between them produces neither (§5). **(2)** Capability interventions do not move the odds between them; only infeasibility does, with a measured dose-response (§3). **(3)** The depth budget moves the odds smoothly, and the critical demand we registered does not exist (§6). **(4)** Length does not bind and capacity does, not by making the front faster but by composing more reliably at the same depth and shifting which algorithm is installed (§7). **(5)** These recurrences converge in direction while their raw norm diverges, so the standard convergence diagnostic misclassifies every model here (§8). **(6)** On the two public looped LMs, inference depth from 1 to 64 does not move a two-composition ceiling (Appendix B).

2 SETUP

Task. Prefix products over a finite group: given $g_1, \ldots, g_n$, output every partial product $g_1 \cdots g_i$. Position i requires composing i elements, so the task has an exact, known depth requirement, which is what makes a frontier measurable. We use S_5 throughout, with A_5, the solvable non-abelian S_4 and the abelian $\mathbb{Z}_{60}$ as contrasts.

Model. A decoder-only transformer whose single core block is applied T times with tied weights, no prelude or coda layers, readout $\mathrm{head}(\ln_{\mathrm{f}}(x))$, presets at 5M, 15M and 50M. Having no coda matters for §9: the loop update never sees the total T, so the state after k steps of a T-step run is identical to the final state of a k-step run, verified numerically (max logit difference 0.000×10^{0}).

Contracts. A contract fixes T as a function of n. T=n leaves room for a scan. $T = \lceil c \log_2 n \rceil$ is the log budget. The *demand-constant* contract sets $T = \lceil n/d \rceil$, holding the covered fraction $n/T(n)$ at d for every length, which is the contract §6 sweeps. Jittered variants draw T uniformly over $[\mathrm{base}/2,\ 2\,\mathrm{base}]$, leaving the mean pinned.

Lengths follow a curriculum (Bengio et al., 2009), and the held-out split withholds transition pairs rather than whole sequences, so a model cannot pass by memorising bigrams (Lee, 2026).

An accounting identity. To emit position n correctly a model must have composed n elements, and in T loops a front advancing at v positions per loop covers vT. So any contract that solves length n in $T(n)$ loops requires

$$v(n) \;\geq\; \frac{n}{T(n)} \;=\; d, \tag{1}$$

intervention	median T^* slope	efficient seeds
none (control, $T{=}n$)	1.00	0 / 9
un-tying (loop index, FiLM)	1.00	0 / 8
circuit supervision	0.79	1 / 10
hard depth budget	**0.35**	92 / 94

Table 1: Making the efficient algorithm *available* does not install it. Only making the scan infeasible does. Slope 1.00 is a literal one-position-per-loop scan; *efficient* means a causal settle slope below 0.6, the floor of the empty band in §5. The budget row is out of 94 rather than 8 to 10 because that arm is the one we swept, and 2 of its 96 checkpoints fail the inclusion gate.

the *covered-fraction demand* d. It is architecture-independent and says only that a budget demanding more speed than a model achieves cannot work.

Instruments. Reach is the largest n solved at ≥ 0.90 held-out accuracy at the model's own best depth T^*. Causal speed comes from activation patching: $\mathrm{settle}(j)$ is the loop after which no intervention at any upstream position $i < j$ changes output j, and its slope against j is loops per position, whose reciprocal v is positions per loop. Only upstream sites count, since including $i{=}j$ makes $\mathrm{settle}(j) = T$ for every model, which an early control caught. A scan gives slope 1.

3 The sequential default, and what does not move it

Trained at $T{=}n$, every model we have learns the scan, with a median slope of 1.00 and causal v flat at every length on all 29 unbudgeted checkpoints. That default then survives both of the interventions one would expect to break it.

It tracks non-commutativity, not circuit complexity. The natural reading is that S_5 is NC^1-complete, so nothing better exists at this depth. That reading is wrong. Solvable non-abelian S_4 is not NC^1-complete and shows the same strict scan anyway, while abelian $\mathbb{Z}_{60}$ does not show it. So what the default tracks is non-commutativity of the group, and not the hardness of the language.

Capability does not move it; only infeasibility does. Un-tying the loop, by loop-index encoding or by FiLM conditioning, leaves the median slope at exactly 1.00 on 8 of 8 trained seeds. We then supervised the target circuit itself, attaching auxiliary heads that ask the state at loop k to encode the width-2^k block products a parallel-prefix circuit would compute. That is handing the model the answer. The median slope moves to 0.79, with 6 of 10 supervised seeds falling below the sequential line against 0 of 9 controls ($p = .0077$), though only 1 of 10 reaches efficient against 0 of 9 ($p = .53$). So the displacement is significant while the conversion is not, and only a hard budget reaches 0.35.

4 Where the constant-speed frontier law breaks

If the default is a scan, then the frontier advances one position per loop and its speed does not depend on length at all. Zhang et al. (2026) report exactly that, as a law. In our own runs it holds without a budget and fails under one.

Behaviourally. Fitting $T^*(n)$ per checkpoint and selecting between a constant, an affine and a sub-linear frontier by AICc, speed is constant on **23 of 23** unbudgeted runs and grows on **105 of 164** deterministic budgeted ones. Arms are assigned by the rule the causal counts below use, so both halves describe one population; jittered budgets are a separate condition, excluded rather than pooled.

The unbudgeted side is robust, and no unbudgeted run grows under any arm definition or run set we tried. The budgeted fraction is not, moving between roughly two thirds and nine tenths as "budgeted" is redefined, so we report it under a stated rule and lean on the contrast, not the proportion.

Causally. The same split appears in the intervention measurement, which does not depend on where anyone draws a solve threshold. Causal v is non-growing on **29 of 29** unbudgeted checkpoints and

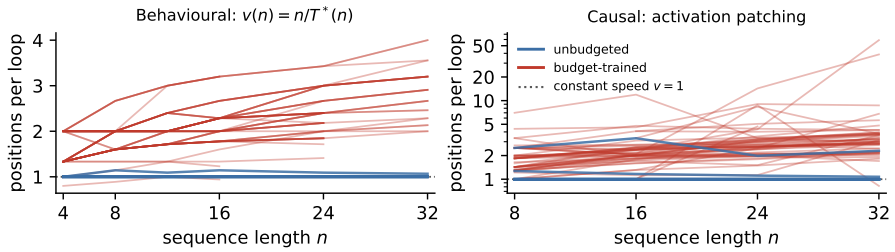


Figure 1: Frontier speed against length. Unbudgeted runs are flat, which is the reported law. Budgeted runs grow. The split is the same behaviourally and causally, and §5 shows the growing side is not a continuum.

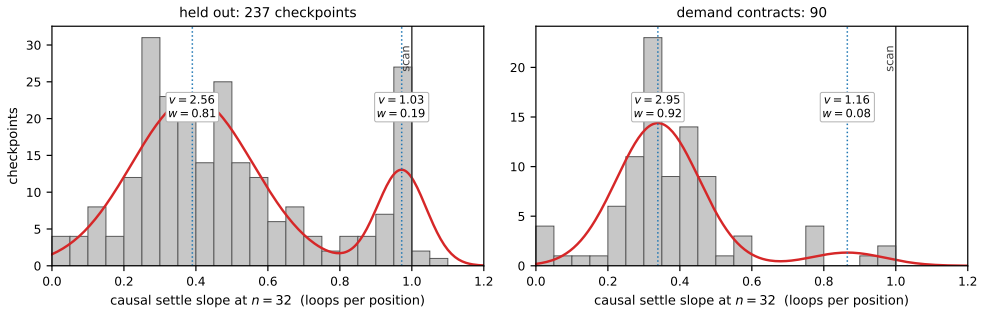


Figure 2: The distribution of causal settle slope at n=32, on the bounded slope rather than its reciprocal. **Left:** the 237 held-out checkpoints, measured before this hypothesis existed. **Right:** the 90 demand contracts. Both are bimodal and in both the slow mode sits at the literal scan (slope 1). What training moves most visibly is the mixing weight. Red curve is the fitted mixture.

rises monotonically on **87 of 96** budgeted S_5 ones. Median growth across the length range is $1.00\times$ without a budget and $\mathbf{1.91}\times$ with one.

Figure 1 shows both regimes on one axis. The constant-speed law describes one of them rather than the architecture, and Equation 1 says which, because a budget leaving room for a scan is exactly the case demanding no speed growth. What replaces the scan when that room is taken away is §5.

5 Two algorithms, not a continuum

Table 1 reads naturally as a continuum of efficiency. It is not one.

The distribution of causal speed is bimodal. Every checkpoint with a usable damage cone gives a settle slope at n=32, and Figure 2 plots both samples. On the 237 of the original campaign a two-component Gaussian mixture beats a single Gaussian by **77.7 BIC**, its components sitting at slope 0.972 ($v = 1.03$, weight 0.19) and 0.390 ($v = 2.56$, weight 0.81).

The result rests on replication, not on our registered test. This pattern turned up after a different registered prediction had failed, the setting where a reframe deserves least trust. So we lead with checkpoints it could not have influenced. The 237 were measured weeks earlier, before the hypothesis existed. Their slow mode sits at $v = 1.03$, the literal one-position-per-loop scan, and the eight baseline checkpoints in that set have median slope exactly 1.000, the instrument's own control passing.

A separate 90 demand-contract checkpoints, measured afterwards, reproduce both modes: 0.339 ($v = 2.95$, weight 0.92) and 0.865 ($v = 1.16$, weight 0.08), preferred by 21.6 BIC. In both sets the slow mode sits at the literal scan. The fast modes are 2.56 and 2.95, a difference of 0.051 in slope, only 0.6 of the seed-to-seed spread within a single contract, so we do not treat them as distinct. Our registered test of mode stability across demand failed (Appendix A). What replicates is the two-class

structure and the scan mode's location. The mixing weight moves as the contracts predict: 19% scan in the older set, which contains unbudgeted baselines, against 8%.

Our registered test of this failed and is reported as failed (Appendix A): we pinned $v = 1/\text{slope}$, which is unbounded, and a Gaussian mixture fitted to it put a zero-width component on one outlier. On the bounded slope the tests pass, but that reparameterisation is not evidence, having been chosen after the first choice misbehaved.

Three ways this could be an artifact. The mixture is fitted, the checkpoints are not a random sample, and the inclusion rule is a filter, so we tested all three.

Independence. The 237 cluster into 40 contract families that BIC treats as exchangeable. Refitting on **one checkpoint per family** leaves the structure intact, at **30.0** BIC for n=40.

A test that assumes no parametric form. Our own registered mixture misfired once, so a mixture-BIC argument is the very thing we have shown can fail. Silverman's critical-bandwidth test assumes no Gaussian form, and it **rejects unimodality** on the held-out set: $p < 0.001$ on all 237 ($h_{\text{crit}} = 0.150$), and $p = 0.021$ refitting one checkpoint per family (n=40). It does *not* reject on the 90 demand contracts ($p = 0.29$), which we state instead of omitting, because that replication rests on the mixture fit and on both modes landing where the held-out set puts them.

The inclusion gate, and a stricter one. Of 249 cones, 12 are excluded: 9 with fewer than three usable rungs, 3 with a non-positive slope at n=32. Twelve points, wherever they landed, cannot open a valley of that width. Our rule also admits a cone that rises then collapses at the tail (§11). Requiring instead that the profile rise across the fitted range, with no free parameter, excludes **no held-out checkpoint at all** (the same 237, 77.7 BIC, $p < 0.001$) and 2 of the 90 demand contracts, where the mixture *strengthens* to 29.2 BIC. Nothing here turns on which gate is used; the comparison is in the released `gate_sensitivity.py`.

We demanded a speed inside the valley and did not get one. Everything above defends an observation. We sampled contracts and found two clumps, which leaves open that the valley belongs to the grid, not the learner. Registered before any run at these demands existed (Appendix A), E34 trains 20 seeds at each of $d = 1.25$, 1.5 and 1.75, asking for a speed in $[1.25, 1.67]$, the band the fitted components leave empty. Of 59 usable checkpoints **4 land in the demanded band** against a registered ceiling of 15%, Silverman **rejects unimodality** on the pooled 59 ($p = 0.005$), and every arm's median slope sits more than 0.10 from the demanded $1/d$ (gaps 0.144, 0.171, 0.115). All three registered predictions hold, and every arm overshoots, running faster than its contract asks instead of settling in the band.

What the demand does to the modes was not registered, so this is post-hoc. Fitted on these arms they sit at 0.485 and 0.921, against 0.390 and 0.972 on the original 237: both move inward while the gap stays empty. The attractors are not independent of the contract and we do not claim they are.

One contract, one rung: the split is not an artifact of pooling. Every distribution above pools contracts, and a mixture of contracts can produce a mixture of speeds without any single model being one of two things. Inside one demand arm everything is identical but the seed. Silverman still **rejects unimodality within a single arm** on 2 of the 3, at $p = 0.047$ each on 20 seeds. On the clearest of them sixteen checkpoints lie between 0.35 and 0.55, then comes a gap, then 0.97 and 1.00. Nor is the class a property of the rung. Assign each checkpoint to its nearer component at n=24 and then again at n=32, and the two agree on **58 of 59**. Two caveats: at 20 seeds an arm this is underpowered, and the arm that does not reject has no slow-mode seed in its sample. And Hartigan's dip, a second distribution-free test, does *not* reject on the pooled sample ($p = 0.62$) where Silverman does ($p = 0.005$). Dip is conservative against a minority mode and this one holds 19% of the mass, but we registered the two agreeing as a check and they did not.

One attractor plus a dial? The band says no. A weaker reading fits much of the above, in which the scan is a discrete attractor at slope 1 and everything below is a continuum of compressed fronts whose speed the budget sets. Parts of our own evidence support it. Within one arm, where only the seed differs, the fast side has a pooled spread of 0.088 in slope, and its arm means move from 0.529 to 0.440 as demand rises, 2.5 of that spread. The fast side is not a fixed point; the contract moves it.

What the reading cannot explain is the band, because a dial lands where it is pointed. The case that separates the two readings is d=1.5, whose contract asks for slope 0.667, the centre of the empty band. A continuum would put its checkpoints there. Instead **18 of 19 land below** 0.6, one at 0.76, with a median of 0.496. We report the other two arms rather than lean on them. d=1.25 asks for 0.80, above the band, and its median of 0.656 does fall inside it, but that is what a bimodal sample does, with 10 of its 20 checkpoints below 0.6 and 8 above 0.8, so the median lands in the gap between the clumps rather than on any checkpoint near it. d=1.75 asks for 0.571, below the band, so it is not evidence about the band either way. Across all three, 4 of 59 land in the band, the figure P2 registered against a 15% ceiling. So the contract shifts the fast region without moving it into the gap, the scan sits at 1, and no contract we tried fills the space between. That is the sense in which there are two algorithms, given that at any fixed contract the outcome is one of two, and what the budget sets is the faster one's location, not its existence.

What pressure installs is not the textbook tree, with an exception. Fitting each checkpoint's causal settle curve linear-in-position against logarithmic at the longest rung, the log form wins on **56 of 96** budgeted checkpoints and on **0 of 25** unbudgeted ones (25 rather than the 29 of §4: that count spans S_5 and A_5, this fit is S_5 only and needs at least eight settle points): a compressed front rather than a scan, but not a parallel-prefix tree either, since a tree would put every budgeted checkpoint in the log column and it does not.

The budgeted count runs from 45 to 60 of 96 across rungs and across whether the log is taken at j or $j+1$, and we report the longest rung with $\log j$. **The contrast is not sensitive**, since the unbudgeted count is 0 under every variant. This is the shortcut-to-an-automaton of Liu et al. (2023), arrived at under depth pressure and measured causally. Nine of the 121 checkpoints this fit covers have a settle profile that does not rise with position at all; a genuinely parallel solution does occasionally appear (Appendix E).

6 What moves the odds: the depth budget

The registered step function does not exist. Holding d constant across length, we swept $d \in \{2, 2.5, 3, 3.5, 4, 5, 6, 8\}$ at ten seeds each and registered that reach at n=64 would be a *step* function of d with the knee in $[3, 5]$, naming a smooth decline as the failure condition. Reach falls $9, 7, 7, 6, 6, 3, 1, 0$ of ten. Model selection prefers a logistic in d over a free-knee step by 10.8 BIC scored at unconstrained depth and 8.7 scored within the trained budget. **There is no critical demand at the population level**, and the prediction failed.

An interpretation, offered as such. Equation 1 is a hard threshold per model, and achievable speed varies continuously across seeds, so the population curve is the distribution of that threshold rather than a degradation of any one model. Fitted at 5M the reading is strikingly accurate. A model solves within its budget exactly when its own causal speed exceeds d, true of 70 of 75 checkpoints. We then tested it on 32 held-out checkpoints at 15M and 50M with the sample size fixed in advance, and **it failed**, at 64% and $p = 0.137$ (§11). We therefore offer the reading and withdraw the rule.

Jitter, at matched demand, does nothing. The demand-constant contract has a zero-width twin, so jittered and deterministic arms are matched on compute by construction. No earlier comparison in this literature is. At $d \in \{3, 4, 5\}$ reach is 16 of 30 against 5 of 15 ($p = 0.171$), and the causal slope distributions are indistinguishable (medians 0.335 and 0.353, $p = 0.296$). Variance per se moves neither.

7 Capacity and length

Length does not bind. Holding d=3 and varying only trained length, reach at $n_{\max}$ is $7/10$, $3/5$ and $4/5$ at $n = 64$, 96 and 128. Pooling the two longer arms against the n=64 reference gives $7/10$ against $7/10$, one-sided Fisher $p = 0.686$. Reach does not fall across a fourfold length increase.

This is a null and we state its power alongside it. The test could not have detected a fall from 70% to 50%, and would have caught a fall to 20% about two thirds of the time. So it is evidence against

collapse, not of exact flatness. The log-budget control at n=128 was registered at $0/3$ and came in at $1/3$, so the head-to-head is $4/5$ against $1/3$, $p = 0.286$, directionally as predicted but not significant.

Capacity binds. The same demand sweep at three sizes gives reach $17/40$ at 5M, $17/20$ at 15M and $11/12$ at 50M; one-sided Fisher against 5M gives $p = 0.002$ and $p = 0.003$. The whole demand curve shifts. At d=6 a 5M model reaches n=64 in 1 of 10 seeds, and a 50M model in 3 of 3.

But not through anything we can measure. The obvious mechanism would be that capacity buys speed, and our measurement does not show it, since all twelve 50M checkpoints have measured causal speed below their own demand while four solve regardless.

It is not width either, and what it moves is the mixing weight, on which two measurements agree. Four arms trained for a different purpose (§11) vary the tied block's depth at matched parameter count and add a narrow control, and across three arms near 5M spanning widths 640, 448 and 384 the share of checkpoints that learn the scan is 20%, 0% and 5%, no trend in width, while holding width at 384 and raising capacity 2.9× takes it from 68% **to** 5%. The registered comparison across model scale points the same way and is underpowered, with the scan mode at $7/75$ at 5M against $0/28$ at 15M and 50M combined, $p = 0.100$. Neither is decisive on its own. The large effect is post-hoc and no pair is a clean one-variable contrast, since the narrow control differs from the 384-wide arm in block depth too, and the registered one is directional but not significant. Together they say capacity changes *which* algorithm is installed rather than how fast the fast one runs.

That is not a violation of Equation 1, and cannot be. It is an accounting identity. A front covering vT positions in T loops either reaches n or it does not. Four solves below the line say one of the two quantities compared is wrong for these models, not that capacity outran the counting. The likely culprit is named in §11. Causal speed is measured at n=32 and applied at n=64, and since speed grows with length the measured value systematically understates what is actually available at the length being solved. Read that way, the twelve are a result about extrapolating short-ladder cones, not a result about capacity.

So capacity buys reach without buying speed, and the question is through what channel. Two candidates are separable on the cones we already have.

Not the shape of the front. The cone counts only upstream sites $i < j$, so a model gathering information non-locally instead of sweeping left to right would solve while looking slow to the instrument. It is not what happens. The settle profile is just as monotone at 50M as at 5M, with a non-monotone step fraction of 0.000 at both and the same normalised jump size. Larger models sweep the same way.

Fidelity, not speed. At matched length and matched depth, n=32 and T=16, larger models are simply more often correct, with 73 of 90 checkpoints clearing 0.90 at 5M against 32 of 32 at 15M and 50M combined, one-sided Fisher $p = 0.0036$. The denominators differ from the $0/28$ above because reach and fidelity need only a solve ladder, while a class needs a usable damage cone, and four of the 32 fail the gate of §5. Equation 1 counts positions covered and assumes each is composed correctly; a model that composes more reliably at the same depth needs less margin over the bound to finish the sequence. On that reading the effective demand for a larger model is below $n/T(n)$, and the identity is being applied with the wrong d rather than being outrun.

We label this exploratory. It was not registered, the fidelity measure is whole-sequence accuracy at a matched cell not a decomposed per-step error rate, and it is a consistency between two measurements, not an intervention. What it does settle is which candidate survives, and the non-local-front explanation does not.

8 These recurrences are rays, not fixed points

Traced past their optimal depth, 130 of 131 checkpoints converge in *direction*, with lag-1 cosine 1.00000, while the raw state norm grows without bound, by up to 316×. LayerNorm divides that scale out before the readout, so the output is unaffected.

This matters because looped transformers are increasingly analysed as fixed-point iterations (Blayney et al., 2026; Movahedi et al., 2026), and the standard convergence diagnostic reads the raw update

norm. On that measure every model in this study is non-convergent. We made that error ourselves before measuring on $\ln_f(x)$, the state the readout actually consumes. The same diagnostic is being applied to public looped language models. We checked whether the ray behaviour transfers to one and it does **not**, so we report the measurement lesson and not a general claim about the architecture.

9 What depth variance does and does not buy

Over-looping does not destabilise the computation. §8 already established that the recurrence settles, with **zero** limit cycles and zero divergences across the same 131 checkpoints. What varies is *which* attractor it settles into. Measured on the model's own answer rather than on any collapse threshold, deterministic budgets land on a wrong answer 16 of 44 times and jittered budgets 5 of 75 ($p = 7.0 \times 10^{-5}$), and the landing is early, at $1.1\times$ to $1.6\times$ peak depth. Figure 3 in Appendix D shows both halves.

This contingency has no independent confirmation. We registered a jitter-width sweep as its confirmation (Appendix D), and although both halves have run, neither fires. Reach separates no widths, and the attractor half gives 0 of 6 at width zero against 4 of 46 pooled above it, nominally the wrong way round and underpowered by roughly a factor of three on the arm carrying the contrast. The contingency therefore rests on its original 119 checkpoints.

And the mechanism we previously asserted is withdrawn. An earlier account of ours explained this by saying that jitter forces the model to solve the same length under many budgets, which is a claim about variance per se. §6 tests exactly that, on arms matched by construction, and variance per se changes neither the algorithm nor reach. We withdraw the explanation and let the attractor numbers stand on their own, which is weaker and is what the evidence supports.

10 Related work

Depth-recurrent and looped transformers are studied as an efficiency mechanism (Bae et al., 2025; Dehghani et al., 2019; Geiping et al., 2025; Zhu et al., 2025) and as a route to algorithmic generalisation (Bansal et al., 2022; Fan et al., 2025; Giannou et al., 2023; Schwarzschild et al., 2021). How models represent and update state has been studied directly (Li et al., 2025), looped depth argued to substitute for reasoning tokens (Saunshi et al., 2025), and adaptive depth goes back to Graves (2016). Expressivity results place log-depth recurrence above fixed depth (Merrill et al., 2024; Merrill and Sabharwal, 2025; Xu and Sato, 2025), and NC^1-completeness of word problems over non-solvable groups (Barrington, 1989; Ladner and Fischer, 1980) is the standard reason to expect these tasks to be hard. §3 shows the default tracks non-commutativity rather than that hardness, since solvable S_4 scans too.

Zhang et al. (2026) are the closest concurrent work and study the same family of tasks. They report the constant-speed frontier law and ask when a weight-tied recurrence becomes an algorithm, in terms of convergence selection. §8 and §9 measure that from the other side: the recurrence essentially always converges, in direction, and what training selects is *which* fixed point. The two efforts were independent and agree on the unbudgeted regime.

What we add is what happens outside it. Constant speed is a property of budgets that leave room for a scan, which by Equation 1 is the case demanding no speed growth; outside it the outcome is one of two discrete algorithms rather than a faster version of one. They also train with a ± 1-jittered log budget without isolating it, which §9 discusses. H. Zhang (2026) give a memory-budget separation between chain-of-thought and compressed looped transformers, which bounds what a compressed recurrence can represent rather than which algorithm it installs.

11 Limitations

The per-model bound failed its confirmation. The reading of Equation 1 in §6 scores 70 of 75 at 5M and 18 of 28 out of sample, $p = 0.137$. It beats a majority-class baseline by 11 points, so it is not worthless, but it is not a law and does not lead. All eight out-of-sample errors have speed just below

demand, which is consistent with a bias registered in advance. Speed is measured at n=32, applied at n=64, and grows with n. Testing that needs cones at n=64, roughly 16× the ladder used here.

Causal speed is measured only to $n \leq 32$. The scaling claims of §7 are at $n = 96$ and $n = 128$, where we have none, since cost is T^2L^2B per length.

Reported reach overstates what was learned. At their own trained length, roughly 47% of demand-contract solvers get there only by being run deeper than they were trained. The evaluation picks T^* freely, so "solves $n_{\max}$" is not "learned to solve it under its budget". Where that matters, §6 scores within the budget.

Thin arms. The n=128 comparisons rest on three to five control seeds, the 50M cells on three; no per-cell claim is made.

Why the task here is synthetic, and what that costs. A frontier speed is only measurable against a *known* depth requirement. Position i of a prefix product needs exactly i compositions, so $\mathrm{settle}(i)$ has a ground truth to have a slope against. Natural text has no such quantity, which is why the instrument runs on a synthetic task and the pretrained models can only be probed behaviourally. The cost is real. Nothing here establishes that the two-algorithm structure exists in a model trained on language, and the one property we could check across the boundary, the ray behaviour of §8, did *not* transfer. Testing the class structure on a pretrained looped LM needs a task with a known depth requirement that such a model can do, a construction we have not solved.

We do not know what sets the fast mode, and we tested one guess. The fast mode is characterised negatively throughout, as a compressed front and not a tree. We registered a mechanism, that it is a chunked scan whose chunk size is one tied block's serial composition capacity, and trained 80 models over four architectures to test it. **It is refuted.** Fast v is not monotone in block depth (4.08, 4.19, 3.48), the predicted ratio is 0.85 against a registered band of $[2, 4]$, and a width control lands with the wrong arm. Where the fast mode sits remains unexplained.

The speed statistic misbehaves near zero. We report $v = 1/\text{slope}$, which is unbounded. One checkpoint has a settle profile that rises normally and then collapses over its last six positions; the fit flattens, slope reaches 0.036 and v reaches 27.6. It passes our inclusion gate and shifts its arm's mean v by 48%. A principled gate exists, requiring the profile to rise across the fitted range, but it would have to be applied to all 344 cones and would move §5. We report the registered statistic, a robust median beside it, and name the affected checkpoints.

What to do differently, if you are training one of these. Three things follow without accepting our interpretation. Un-tying the loop or supervising the circuit will not buy the efficient algorithm; only removing the depth to run a scan reliably does, a decision about the budget rather than the model. Capacity is the other lever and works differently. It does not make the front faster, but makes the scan less likely and each composition more often correct at the same depth. And evaluate inside the training budget, not at the best depth on a ladder. The 47% above is the share of reported successes a fixed-loop deployment would not get.

12 Conclusion

Depth-recurrent transformers on this task do not learn a spectrum of circuits. They learn a scan or a compressed front. Training moves the odds between them instead of improving either, and asked for a speed in between they supply one of the two. Making the efficient algorithm available does not move those odds; making the scan infeasible does. Length does not bind and capacity does, not by making the front faster but by composing more reliably at the same depth and by shifting which algorithm is installed. The registered critical budget is false at the population level, and the per-model reading that would explain it away failed its own held-out test.

AI USE STATEMENT

We used a generative AI assistant (Anthropic Claude) to monitor experiment runs, to produce initial drafts of code and in the analysis of the training-contract comparison of §9. All other work is the authors': the research programme, the pre-registered hypotheses and their success criteria (Appendix A), and every judgement about what the evidence does and does not support.

All AI-assisted work was reviewed by the authors. Every reported number is regenerated from the released artifacts by the script that produced it, which is how the errors we report as corrected were caught, including an inconsistent filter that had the two halves of §5 computed on different checkpoint sets. The authors take responsibility for the content of this work, including text, claims and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work trains small models on synthetic algebraic tasks and evaluates two publicly released pretrained models. It involves no human subjects, no personal data, and no dataset collection. We see no direct route from these results to harm; the practical prescription concerns training-time loop budgets.

REPRODUCIBILITY STATEMENT

Every claim in this paper is computed from released per-run JSON artifacts by released analysis scripts. §2 and Appendix E give the task construction, held-out protocol, model presets, training contracts, ladder grids and sample sizes. The causal instrument is validated against synthetic ground truth with a known answer, and the exactness of the loop-ladder trace used in §9 is verified numerically rather than assumed. Registered predictions, including the fourteen that failed, are scored in Appendix A. Code, per-run artifacts and the analysis scripts are available at `https://anonymous.4open.science/r/iclr2027-artifacts-D31A`.

REFERENCES

S. Bae et al. Mixture-of-recursions: Learning dynamic recursive depths for adaptive token-level computation. *arXiv:2507.10524*, 2025.

A. Bansal, A. Schwarzschild, E. Borgnia, Z. Emam, F. Huang, M. Goldblum, and T. Goldstein. End-to-end algorithm synthesis with recurrent networks: Extrapolation without overthinking. In *NeurIPS*, 2022.

D. A. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in NC^1. *Journal of Computer and System Sciences*, 38(1), 1989.

Y. Bengio, J. Louradour, R. Collobert, and J. Weston. Curriculum learning. In *ICML*, 2009.

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:2604.11791*, 2026.

M. Dehghani, S. Gouws, O. Vinyals, J. Uszkoreit, and L. Kaiser. Universal transformers. In *ICLR*, 2019. *arXiv:1807.03819*.

Y. Fan, Y. Du, K. Ramchandran, and K. Lee. Looped transformers for length generalization. In *ICLR*, 2025. *arXiv:2409.15647*.

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. *arXiv:2502.05171*, 2025.

A. Giannou, S. Rajput, J. Sohn, K. Lee, J. D. Lee, and D. Papailiopoulos. Looped transformers as programmable computers. In *ICML*, 2023.

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

R. E. Ladner and M. J. Fischer. Parallel prefix computation. *Journal of the ACM*, 27(4), 1980.

J. Lee. A held-out transition-pair falsifier for long-horizon non-abelian state tracking. *arXiv:2606.07254*, 2026.

B. Z. Li, Z. C. Guo, and J. Andreas. (How) do language models track state? *arXiv:2503.02854*, 2025.

B. Liu, J. T. Ash, S. Goel, A. Krishnamurthy, and C. Zhang. Transformers learn shortcuts to automata. In *ICLR*, 2023.

W. Merrill, J. Petty, and A. Sabharwal. The illusion of state in state-space models. In *ICML*, 2024.

W. Merrill and A. Sabharwal. A little depth goes a long way: The expressive power of log-depth transformers. In *NeurIPS*, 2025. *arXiv:2503.03961*.

S. Movahedi, V. Milovanović, S. L. Feigin, A. Theus, T. Hofmann, V. Boeva, T. K. Rusch, and A. Orvieto. Fixed-point reasoners: Stable and adaptive deep looped transformers. *arXiv:2606.18206*, 2026.

A. Power, Y. Burda, H. Edwards, I. Babuschkin, and V. Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets. *arXiv:2201.02177*, 2022.

N. Saunshi, N. Dikkala, Z. Li, S. Kumar, and S. J. Reddi. Reasoning with latent thoughts: On the power of looped transformers. In *ICLR*, 2025. *arXiv:2502.17416*.

A. Schwarzschild, E. Borgnia, A. Gupta, F. Huang, U. Vishkin, M. Goldblum, and T. Goldstein. Can you learn an algorithm? Generalizing from easy to hard problems with recurrent networks. In *NeurIPS*, 2021.

K. Xu and I. Sato. On expressive power of looped transformers: Theoretical analysis and enhancement via timestep encoding. In *ICML*, 2025. *arXiv:2410.01405*.

H. Zhang. Chain-of-thought and compressed looped transformers: A memory-budget separation. *arXiv:2605.30757*, 2026.

T. Zhang, J. Hu, Y. Peng, and T. Xie. When does recurrence become an algorithm? Convergence selection in weight-tied looped transformers. *arXiv:2607.20594*, 2026.

R.-J. Zhu et al. Scaling latent reasoning via looped language models. *arXiv:2510.25741*, 2025.

A PRE-REGISTRATION SCORECARD

Hypotheses, interventions and success criteria were registered before the runs they govern. We report every registered prediction, including the fourteen that failed, and we mark the headline numbers that are post-hoc reframings.

The failures are of two kinds and we separate them, because they carry different information. *Predictions about the models* that turned out false: **H-B1** (un-tying), **criterion (b) of RQ-B4** (accuracy at $4\times$ the training length), **circuit supervision**, **jittered models never flip**, **the jitter-width confirmation**, **a critical demand**, **the long-context control**, **the per-model bound at scale**, and **all three block-depth predictions**. Each of those tells a reader something about depth-recurrent transformers.

Predictions where our own statistic was the problem, which tell a reader only about our instruments: **loop-tolerance retention**, **two of the three algorithm-class predictions** (a Gaussian mixture on the unbounded $1/\text{slope}$, which put a zero-width component on one outlier), and **the dip-test agreement**. All four share one cause, which is a distributional test applied to a heavy-tailed reparameterisation of a bounded quantity. That is why we report the bounded slope throughout, and why we flag the pattern instead of leaving a reader to find it. Twice now a distributional choice has misbehaved and the analysis has moved. Each move is defensible on its own, but the pattern is a real weakness of the measurement and not of the models.

The first two sit inside the original slate below, and the rest have headings of their own.

Earlier versions of this appendix reported five, then eleven. Each was correct when written and became wrong as later experiments landed. The count is stated here as fourteen and every addition is itemised, so a reader can check the arithmetic instead of taking it on trust.

The original slate. **RQ-B1 (replication): confirmed and sharpened**, with $T^* = n$ exactly on both non-solvable groups. **RQ-B2 (diagnosis): answered**, a literal one-token-per-loop scan whose efficient alternative turns out to be a compressed hierarchical front and not the textbook parallel-prefix tree. **H-B1: falsified** on 8 of 8 trained seeds across two un-tying knobs. We keep it at full prominence because it is the paper's main negative and because it is what makes Table 1 informative. **H-B2(a): supported**; H-B2(b), that infeasible budgets fail informatively, was also observed at c=1. **RQ-B3 (dynamics): answered**, with the registered grokking-style alternative (Power et al., 2022) wrong in direction, since no early associative phase exists to be abandoned. **RQ-B4 (intervention): partially answered**; we tested the loop-budget branch and it works subject to the reach caveats of §11, and we make no claim about the two branches not run.

The registered intervention success criterion required (a) a flipped probe signature, (b) accuracy $\geq 95\%$ at $4\times$ the training length under log budgets, and (c) 3/3 seeds. (a) holds and (c) holds for the algorithm class at $c \leq 2$. **(b) is not met**: the induced efficiency tracks lengths inside the trained range and does not extrapolate $4\times$ beyond it. We report this rather than weakening the criterion after the fact.

Registered and failed: circuit supervision. We predicted that supervising the block-product ladder directly would flip the algorithm to slope < 0.6. It did not: median slope 0.79, with 1 of 10 seeds reaching efficient against 0 of 9 controls ($p = .53$). The displacement is significant ($p = .0077$) and the conversion is not. The failure is what completes the dose-response of Table 1.

Registered and failed: loop-tolerance retention. We predicted median retention $R = \mathrm{acc}(4 \times \mathrm{peak})/\mathrm{acc}(\mathrm{peak})$ above 0.8 for both jittered families and below 0.2 for fixed depth and the log budget. The jittered half held (0.965 and 0.996). The deterministic half is refuted: restricted to checkpoints that can do the task at all, fixed depth retains 0.816 and the log budget 0.981. Every contract tolerates $4\times$ over-looping on accuracy, and the metric itself was the problem, which is why §9 measures the answer instead. Ungated, those two numbers read 0.468 and 0.667, which looks like a near miss in the predicted direction; it is not, because at n=32 the fixed-depth arms have median peak accuracy 0.026 against a chance rate of 0.0083, so the ratio divides one near-chance number by another.

Registered and failed: jittered models never flip. We predicted that at least 95% of jittered checkpoints would never change their answer under over-looping. The measured figure is 93%, 70 of 75, so the prediction fails and the claim "jittered models never land on the wrong attractor" must not be made. The effect is 7% against 36%, not 0%.

Registered and failed: the jitter-width confirmation. We registered an independent jitter-width sweep as the confirmation of the attractor contingency of §9. Both halves have run and neither fires: reach separates no widths, and the attractor half gives 0 of 6 at width 0 against 4 of 46 pooled above it, nominally the wrong way round and underpowered by roughly a factor of three on the arm carrying the contrast. The contingency therefore rests on its original 119 checkpoints without independent confirmation, which we state in the body as well as here.

Registered and failed: a critical demand. Holding the covered-fraction demand $d = n/T(n)$ constant across length, we registered that reach at n=64 would be a *step* function of d with the knee in $[3, 5]$, and named a smooth decline as the failure condition. On 95 checkpoints spanning $d \in \{2, \ldots, 8\}$ the decline is smooth: model selection prefers a logistic in d over a free-knee step by 8.7 to 10.8 BIC depending on whether reach is scored at the trained budget or at unconstrained inference depth. There is no critical demand at the population level. Every model has a threshold equal to its own achievable speed; what varies is the speed, so the population curve is the distribution of that threshold rather than a degradation of any one model.

Registered and failed: the long-context control. Testing the demand-constant contract at n=128, we registered that the log-budget control would reach 0 of 3. It reaches 1 of 3. The headline contrast

is therefore 4 of 5 against 1 of 3, one-sided Fisher $p = 0.286$: directionally as predicted and *not* significant. Three control seeds cannot carry that comparison and, unlike the primary test, it cannot be rescued by pooling.

Registered and failed: two of three algorithm-class predictions. Pointing the causal instrument at the demand contracts, we registered that causal speed would be bimodal, that the component means would be stable across demand while the mixing weights moved, and that the slow component would sit at $v \in [0.85, 1.35]$. On the registered statistic the second and third fail: means shifting with demand beats stable means by 73.3 BIC, and the slow component lands at 3.92. The registered statistic was $1/\text{slope}$, which is unbounded and heavy-tailed, and the mixture it selects places a zero-width component on a single outlier; the failure is therefore partly ours and is reported as a failure regardless. On the bounded slope the same tests pass, and the two-component structure replicates on 237 independently measured cones, where the slow component again sits at the literal scan ($v = 1.03$). The fast components are near but not identical across the two sets, 2.56 against 2.95, a difference of 0.6 of the within-contract seed spread, so §5 does not treat them as distinct. We report the structure and the location of the scan mode, not the registered test.

Registered and failed: the per-model bound at scale. A rule found post-hoc in the 5M data, that a model solves within its trained budget exactly when its own causal speed exceeds the demanded speed, scores 70 of 75 in sample. We registered its confirmation on 32 held-out checkpoints at 15M and 50M, with the sample size and the stopping rule fixed in advance. It scores 64%, one-sided Fisher $p = 0.137$, and is withdrawn. The informative part of the failure is that all twelve 50M checkpoints are predicted to fail and four solve: capacity buys reach without buying causal speed, so speed is not the channel through which scale helps, which is why a speed-based bound does not transfer.

Registered and confirmed. That the recurrence settles rather than oscillating or diverging: 130 of 131, with zero cycles. That the answer flips before accuracy-based collapse fires: 5 of 6 seeds. That jittered models show strictly less causal growth than their deterministic twins: $1.23\times$ against $1.99\times$, and it replicates at 15M and 50M. The registered null that jitter buys competence without inducing any compression did *not* fire, which is why §9 can make a mechanism claim at all.

Confirmed only in letter, and not used. We registered that causal growth would rank by demand rise, placing `const_jitter` above `log_jitter`. Medians are $1.16\times$ against $1.13\times$, in the predicted direction, but the within-family spread is 1.03 to 1.30. Three arms a side cannot separate distributions that wide, so we report the direction and build nothing on the ordering.

The dip test disagreed with Silverman, and that was a check we registered. Re-analysing the valley checkpoints, we registered that a second distribution-free test would agree with Silverman on the pooled sample. It does not: Hartigan's dip gives $p = 0.62$ where Silverman gives $p = 0.005$. Dip is conservative against a minority mode and this one carries 19% of the mass, which is a plausible reason but not one we registered. The two predictions that did hold in that re-analysis are the within-arm rejection and the rung agreement, both reported in §5.

Block depth does not set the fast mode: three predictions, all failed. The fast mode is characterised only negatively in this paper, so we registered a mechanism for it, that the compressed front is a chunked scan whose chunk size is the serial composition capacity of one tied block, and trained 80 models over four architectures to test it. All three predictions failed. Fast v is not monotone in block depth (4.08, 4.19, 3.48); the ratio $v(L{=}3)/v(L{=}1)$ is 0.85 against a registered band of $[2, 4]$; and a width control lands with the arm it shares width with rather than the one it shares block depth with. A secondary prediction added by amendment before the deciding arms ran, that the ordering holds on the subset where the budget still binds, did hold (2.12, 2.47, 2.61). The failure condition was named in advance and it fired. What that experiment did show is unregistered and reported as such in §7: capacity, not width, changes which algorithm is installed.

Registered before the data, and confirmed on every clause. E34 asked whether the empty band of §5 is a property of the learner or of the contracts we happened to run. It was registered before any run at these demands existed, against a tree of outcomes fixed in advance, including a named failure condition: had the valley arms produced a unimodal distribution centred on the demanded

speed, the two-algorithm reading would have been withdrawn as a claim about learning and kept only as a description of our contract grid. Sixty runs later, 20 seeds at each of $d = 1.25$, 1.5 and 1.75: Silverman rejects unimodality on the pooled 59 usable checkpoints ($p = 0.005$); 4 of 59 land in the demanded band against a ceiling of 15%; and every arm's median sits more than 0.10 from $1/d$. All three hold. One checkpoint was excluded by the same inclusion gate as the original 249, for having fewer than three usable rungs.

What we did not register is what the demand does to the modes themselves, and it does something: fitted on these arms they sit at slope 0.485 and 0.921 against 0.390 and 0.972 on the original 237. Both move inward while the gap between them stays empty. We report that as post-hoc.

Post-hoc, and labelled as such. Two numbers are reframings adopted after the registered prediction on the same data failed: the collapse-rate contingency (9/47 against 0/77, reported here rather than in the body) and the attractor contingency of §9 (16/44 against 5/75, $p = 7.0 \times 10^{-5}$). Both are stronger than what they replaced and neither was registered. We registered an independent jitter-width sweep as their confirmation and it has now run in full. **It did not confirm them.** The reach half separates no widths; the attractor half gives 0 of 6 at w=0 against 4 of 46 pooled over $w > 0$, which is nominally reversed and underpowered by a factor of about three on the arm that matters (Appendix D). We report this rather than describing the sweep as pending, and we have not replaced it with a different confirmation after the fact.

B THE TWO PUBLIC LOOPED LMS

Everything above is measured on models we trained. The two publicly released looped language models, Huginn-0125 (Geiping et al., 2025) and Ouro (Zhu et al., 2025), let us ask whether any of it is visible outside that setting, on natural text.

They track two composition steps, and depth does not buy a third. Given the same compositional structure in plain English, both answer above chance at one and two compositions and not beyond. We swept inference depth, T from 1 to 64, at three state sizes and two shot counts: 12 sweeps, 96 length-cells.

No sweep reaches the solve threshold at any depth, on any cell. Depth does buy accuracy, but only where the model already tracks: the mean gain from T=1 to the best depth is $+0.225$ at $n \leq 2$ against $+0.052$ at $n \geq 3$. Correcting for the maximum over ten depths, only n=1 and n=2 clear chance on every sweep; n=3 on 3 of 12, $n = 4$, 8 and 12 on none. The few longer cells that clear are *incoherent*: clearing n=16 while failing 4, 8 and 12 is not tracking sixteen compositions.

How weak a probe this is, stated in absolute terms. In absolute accuracy the best any sweep reaches is 0.609 at n=1 and 0.482 at n=2 against chance between 0.20 and 0.33: decisively above chance where they track, never close to solving. What this establishes is narrow and we claim no more: inference depth is not the missing ingredient, the ceiling being invariant to it, to shot count and to state size.

Both models face a demand that grows linearly in length. It would be wrong to describe these as models trained with depth to spare. Huginn-0125 samples its recurrence count during training, which is depth *variance* around a fixed mean, and Ouro trains at a fixed depth. For both the budget does not scale with input length, $T(n) = T_0$, so by Equation 1 the demand they face is n/T_0, growing linearly in n: the harshest of the three contracts in this paper, not the most generous.

A tension worth naming: Huginn is a jittered-depth model and ceilings anyway. That is consistent with §9, where variance alone moves neither the algorithm nor reach, and with a budget that never scales with length. A consistency, not a confirmation: we did not train these models.

C THE COVERED-FRACTION TABLE

Measured against demanded frontier speed under `log_n`, c=2, aggregated over seeds at each length. T^* is the inference depth reaching 0.9 accuracy, so n/T^* is the covered fraction the model actually achieves, and $n/T(n)$ is what the training budget demanded of it.

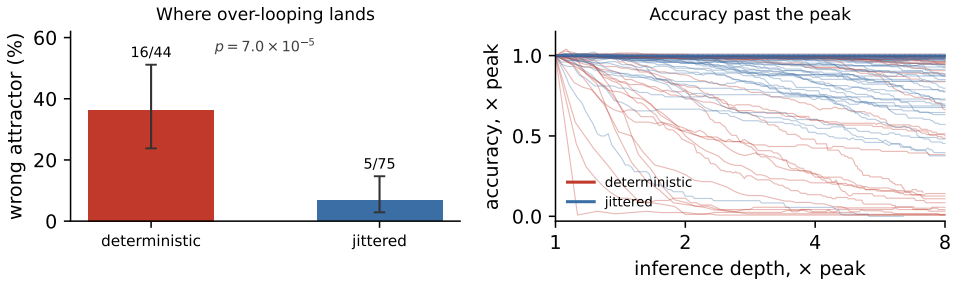


Figure 3: Left: over-looping lands a deterministic-budget model on a wrong answer 16 of 44 times and a jittered one 5 of 75, with Wilson intervals. Right: accuracy traced past the optimal depth on 131 checkpoints. The recurrence settles in every case; what differs is where.

n	4	8	12	16	24	32
demanded, $n/T(n)$	1.00	1.33	1.50	2.00	2.40	3.20
measured $v(n)$, median	1.33	2.00	2.00	2.29	2.67	3.20
ratio	1.33	1.50	1.33	1.14	1.11	**1.00**
measured $v(n)$, mean	1.61	1.88	2.03	2.22	2.62	3.05
ratio	1.61	1.41	1.35	1.11	1.09	**0.95**
seeds	79	79	79	79	71	46

D THE JITTER-WIDTH SWEEP

The mean-matched controls. Every jittered arm in §9 is paired against a fixed-depth arm of equal or greater mean, so the comparison handicaps jitter rather than flattering it. T_0=19 against T=24 is the pre-specified mean-matched pair.

jittered arm	mean	fixed arm	mean	jittered	fixed	p
$T_0 = 8$	10	$T = 12$	12	12/20	4/20	.011
$T_0 = 12$	15	$T = 16$	16	11/20	5/20	.053
$T_0 = 19$	23.5	$T = 24$	24	14/20	7/20	.028

Width. §9 prescribes jittering a tight budget, and Zhang et al. (2026) use a narrow ± 1 jitter without isolating it. The obvious question is whether width matters. We trained six widths w of a c=2 log budget, ten seeds each, where w scales the multiplicative spread so that w=0 is deterministic and w=1 is the factor-of-four contract used in the main text.

width w	0	0.1	0.25	0.5	0.75	1.0
solves $n_{\max}$=32	6/10	8/10	7/10	8/10	9/10	8/10

The direction is right and nothing separates. w=0 against w=1.0 gives $p = .31$, and the narrow ± 1-scale arm is indistinguishable from the widest ($p = .71$), already recovering most of the effect. Ten seeds per arm is below the twenty §9 requires of itself after watching this same rate deflate from 80% to 62% as seeds were added, so we report a direction and not a measurement, and we claim no separation by jitter width.

The attractor half. This is the half that bears on the two post-hoc contingencies of §9, and it is the confirmation we registered for them. We traced all 60 checkpoints past their optimal depth, behind the same 0.5 peak gate used throughout.

width w	0	0.1	0.25	0.5	0.75	1.0
wrong attractor	0/6	2/9	0/8	1/9	1/10	0/10

Pooled, 0 of 6 at w=0 against 4 of 46 at $w > 0$, one-sided Fisher $p = 1.00$: no effect, nominally reversed. **The registered confirmation did not fire.**

Two things are worth separating. The jittered rate *replicates*, 9% here against 7% in §9. What fails is the control arm: only 6 of its 10 seeds clear the peak gate, because a deterministic log budget frequently cannot do the task at all, which is this paper's own reach finding turned against its own experiment. At the deterministic-log flip rate we measure elsewhere, $6/21 = 29\%$, six seeds are expected to yield 1.7 flips and $P(\text{zero}) = 0.13$. Zero of six is an ordinary draw under the effect we claim, so this neither confirms nor refutes it. We report a failed confirmation rather than no confirmation.

E PROTOCOL DETAILS

The nine flat profiles. Nine checkpoints have a settle profile that does not rise with position at all, and the flatness survives across rungs rather than appearing at one length. Seven of the nine come from the tightest budgets, where a positional scan is least survivable. Nine of 121 is no refutation of the compressed-front reading, but a genuinely parallel solution does occasionally appear and we do not claim otherwise.

Reconciling the checkpoint counts. Different claims rest on different subsets, because different instruments require different things. In one place:

- **389** checkpoints carry a damage cone in the release; not all are in scope for any one claim.
- **344** are in scope for §5: **249** held-out and **95** demand-contract.
- **237** of the 249 pass the inclusion gate (three usable rungs, a positive slope at n=32), and **90** of the 95 do. These are the two samples the mixture is fitted on.
- **121** enter the log-versus-linear fit: **96** budgeted and **25** unbudgeted S_5 checkpoints with at least eight settle points. The **29** of §4 is a different set, spanning S_5 and A_5 with no eight-point requirement.
- **59** of E34's 60 valley checkpoints pass the same gate.
- At 15M and 50M combined, **32** checkpoints have a solve ladder and **28** a usable cone, which is why reach and fidelity are quoted out of 32 and the class fractions out of 28.

Every count here is regenerated by the released scripts.

Training. AdamW, learning rate 10^{-3} with cosine decay, weight decay 0.1, batch size 256, 40k steps, bf16 mixed precision, gradient clipping at 1.0. Presets are 5M (d=640, 10 heads), 15M (d=1088, 16 heads) and 50M (d=2048, 16 heads), each a single tied block with no prelude or coda layers. Each batch draws one length from $[4, n_{\text{cap}}]$ with the cap ramped linearly from 4 to $n_{\max}$ over the first 60% of training; from-scratch training at a fixed length fails on S_5 outright, so the curriculum is necessary rather than chosen.

The pretrained-model probe (§B). Both models are evaluated through their released inference interfaces at bfloat16, with the recurrence count set explicitly per run. The task is presented as natural-language text: a set of labelled containers, a sequence of stated swaps, and a question asking where one object ends up, which is a prefix product over a symmetric group written in words. Sweeps cross three state sizes (3, 4 and 5 containers, giving chance rates of 0.33, 0.25 and 0.20), two shot counts (8 and 16 in-context examples), and ten inference depths spanning $T = 1$ to $T = 64$. Each of the 96 length-cells is scored on 128 items, which is not recorded in the artifacts and is recovered from the granularity of the stored accuracies, and a cell counts as clearing chance when it exceeds the chance rate for its state size by at least 0.10. Raw per-sweep outputs are in `runs/e1/` and the aggregation is `scripts/analyze_bridge_depth.py`.

Evaluation ladders. Loop ladders are dense over $T \in [1, 2n]$ and geometric beyond, 512 fresh sequences per (n, T) cell, lengths $n \in \{4, \ldots, 64\}$. Accuracy is last-position only. $T^*(n)$ is the smallest T reaching 0.9, and a run solves n when held-out-pair accuracy at T^* is at least 0.9.

Causal instrument. Activation patching at one (loop, position) cell, with only upstream sites $i < j$ counted; including $i \geq j$ makes every model score identically and was a real bug caught by the control. Validated against synthetic ground truth: a scan returns slope 1.000 and $v = 1.00$, a two-positions-per-loop reference returns 0.484 and 2.07, and resolving everything at once returns approximately 0.

Exactness of the loop-ladder trace. §9 reads the model's output at every loop count from a single forward pass. This is exact rather than an approximation because the presets have no coda layers, so the readout is $\mathrm{head}(\mathrm{LN}(x))$, and the loop update depends on the step index but never on the total T. We verified it numerically rather than assuming it: across nine loop counts the maximum absolute logit difference against separately run forward passes is 0.000×10^{0}, with no argmax disagreements.

Which checkpoints are in which count. The artifact contains 389 damage-cone files. The causal claims of this paper cover 249 of them: log-budget and baseline checkpoints trained to $n \leq 64$. The other 140 are a different condition, added later, and are held out of every count in §4 and §9 rather than pooled into them. They are the demand-constant contracts and the long-context arms, and their causal speed grows on 88 of 134 against 86 of 114 on the set reported here; pooling the two would produce a rate for a question neither section asks. Which 249 is determined by contract, not by date, and the rule is one function in `verify_numbers.py`.

Of those 249, 130 were trained without jitter and 119 with it, and the two are never pooled: §4 reports the former and §9 the latter. Within the 130, the counts quoted in §4 are 29 unbudgeted (19 S_5 and A_5 baselines plus 10 un-tying variants) and 96 budgeted S_5. The remaining five are three budgeted A_5 checkpoints, one abelian $\mathbb{Z}_{60}$ control whose wavefront has no frontier to have a speed, and one whose ladder was too short to fit.

Sample sizes and statistics. At least 3 seeds for every headline claim, 10 to 20 per arm for reliability comparisons, Wilson 95% intervals for rates and one-sided Fisher exact tests for contingencies. Retention and attractor statistics exclude checkpoints whose peak accuracy at n=32 is below 0.5, since a ratio computed on a model that cannot do the task at any depth contributes only noise; both gated and ungated figures are reported in Appendix A.