WHEN DOES RECURRENCE BECOME AN ALGORITHM? CONVERGENCE SELECTION IN WEIGHT-TIED LOOPED TRANSFORMERS*

Tong Zhang
Fudan University
tongzhang25@m.fudan.edu.cn

Junhao Hu
Peking University
junhaohu@stu.pku.edu.cn

Yun Peng
Fudan University
yunpeng4@sigsoft.org

Tao Xie†
Peking University
taoxie@pku.edu.cn

ABSTRACT

When does a weight-tied looped transformer—one block applied T times—implement an actual algorithm? We answer with four findings from controlled populations on group word problems. **(1) The budget law:** free training installs a *linear computation frontier*, a mechanism that solves v positions per loop, and prices its speed by the training contract: $v \approx n_{tr}/T_{tr}$, exactly unity under $T{=}n$ training—SGD consistently selects a frontier whose speed matches the minimum the contract demands. Speed is an experimenter-controllable knob, and granting more test-time loops than ever trained rescues late positions at fixed input length. **(2) Architecture prior, not expressivity, picks the algorithm:** standard-depth transformers learn parallel scans on this task family; weight tying flips the selection to the serial frontier, under global attention, even when positional addressing for a $\log$-depth scan is supplied. **(3) The walls are not where circuit complexity says:** NC^1-completeness costs nothing (A_5 generalizes fully), while group order does (S_5's 120×120 operator deadlocks joint learning)—and an operator-first curriculum dissolves the wall, revealing it as an optimization pathology, not an impossibility. **(4) Mechanisms are portable objects:** warm-starting across budget contracts transfers the algorithm in every seed attempted—re-pricing its speed to the new contract—and bypasses a seed lottery that defeats most from-scratch runs, while imposing seriality through the input schedule fails where free training succeeds—the mechanism can be moved, not mandated. These results are invisible to standard instruments: skip and cross-step similarity metrics provably saturate at the fixed points trained loops converge to, measuring the trajectory's tail while the algorithm lives in its head. We introduce a head instrument, the convergence-time scaling $\tau(n, i)$, validate it causally by activation patching (damage cones whose slope reproduces v), and show head instruments predict which seeds generalize where tail metrics and path-independence scores do not. The phenomena replicate on the public easy-to-hard benchmark, including $4\times$ exact-match length extrapolation.

1 INTRODUCTION

Depth-recurrent (looped) transformers reuse a single block T times, promising adaptive test-time compute: harder or longer inputs get more loops (Dehghani et al., 2019; Geiping et al., 2025; ByteDance Seed, 2025). The implicit mechanistic story is that the loop is an *algorithm*—that loop t performs the t-th step of an iterative computation, so that running more loops executes more steps. This story motivates training recipes, halting rules, and interpretability agendas alike. Is it true?

*Project page: https://huggingface.co/StoneZhang/recurrent-convergence-selection
†Corresponding author: taoxie@pku.edu.cn

The question splits into two: (i) what mechanism does a trained looped transformer actually implement, and (ii) can our instruments even tell? We show the second question gates the first. The natural tests—does skipping a loop hurt? do successive loops use the same circuit?—are *tail instruments*: they probe the trajectory near its end state. But trained loops converge, and a converged state is a fixed point of the loop map; near an idempotent point, skipping is free and successive local circuits agree by construction (Proposition 1). A model that finishes its real work in the first few loops and idles thereafter is indistinguishable, to every tail instrument, from a model that never had discrete steps at all. The algorithm, if there is one, lives in the head of the trajectory.

We make this concrete with a controlled population study: dozens of weight-tied looped transformers on tasks with known one-shot (constant-depth) solutions, spanning recipes (positional encoding, input injection, loop schedule) and seeds. The population exhibits an order-of-magnitude spread in length generalization—including a near-full spread across seeds of a single recipe—yet *every* tail instrument is blind to it: single-step skip retention is exactly total for every model with enough solved examples, cross-step Jacobian alignment is uniformly high, and no cross-step similarity metric correlates with generalization (Appendix C).

We then introduce a *head* instrument. The convergence-time scaling $\tau(n, i)$ is the first loop at which the decoded output at position i reaches its final value and stays there, measured as a function of input length n on solved inputs. Its scaling class is a mechanism signature: a constant τ is a shortcut (or pure refinement); $\tau \propto \log n$ is a parallel scan; $\tau \propto n$ is serial, step-indexed computation. Unlike tail metrics, τ has provable dynamic range: we validate it on a streaming positive control—token i is revealed only at loop i, so serial computation is forced by construction—where it reads exactly $\tau \propto n$ (slope 1.0), while shortcut controls read flat.

Armed with a calibrated instrument, we characterize what free training actually installs. Group word problems supply a task ladder with known theory: solvable groups ($\mathbb{Z}_{60}$, S_4) admit constant-depth shortcuts (Liu et al., 2023), the S_5 and A_5 word problems are NC^1-complete (Barrington, 1989), and $\Theta(\log n)$ looped depth suffices in principle via an associative scan (Merrill & Sabharwal, 2025). Against every option this theory menu offers—one-shot shortcut, $\log$-depth scan, fixed-point refinement—trained models choose a fourth: a linear frontier whose speed is a function of the training contract, whose walls track operator scale rather than circuit class, and which moves between contracts by warm-starting but cannot be installed by input scheduling.

Contributions. (1) The linear computation frontier and its *budget law*: free training installs a serial mechanism whose speed obeys $v \approx n_{tr}/T_{tr}$ (exactly 1.00 under $T{=}n$), with per-position loop-extrapolation rescue at fixed length (§6). (2) Algorithm selection is set by architecture prior: weight tying flips SGD's choice from the parallel scan of standard-depth transformers to the serial frontier, with untied and positional-encoding controls (§6). (3) The learning walls are optimization objects, not complexity-theoretic ones: NC^1-hardness is free (A_5), operator scale is not (S_5), and an operator-first curriculum dissolves the wall; imposed serial schedules fail where free training succeeds (§6). (4) Mechanism portability: budget annealing transfers the algorithm across loop contracts in $4/4$ seeds, bypassing the from-scratch seed lottery (§7). (5) The measurement theory making (1)–(4) visible: a saturation proposition showing standard instruments are blind at fixed points, the $\tau(n, i)$ head instrument with causal validation, and three 16-seed races in which head instruments predict seed-level generalization while tail metrics and path independence do not (§4–5, §8).

2 Related work

Looped and depth-recurrent transformers. Universal Transformers (Dehghani et al., 2019) introduced weight tying across depth; recent open models scale the idea (Huginn-3.5B, Geiping et al., 2025; Ouro, ByteDance Seed, 2025). Looped transformers provably length-generalize on n-RASP-L tasks (Fan et al., 2024) and can emulate iterative solvers (Yang et al., 2023; Gatmiry et al., 2024); Gatmiry et al. (2024) show the global minimizer for in-context regression implements preconditioned gradient descent—an inductive bias toward convergent refinement, consistent with our selection result.

Mechanistic accounts of loops. Blayney (2026) document cyclic fixed points and attention-pattern stabilization in Ouro and Huginn; their Prop. 4.2 (attention similarity follows state convergence) an-

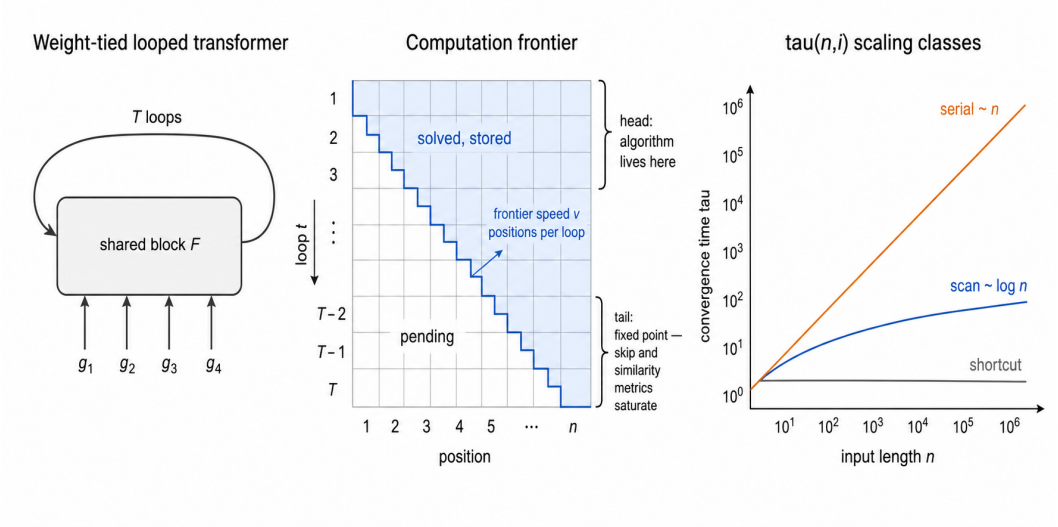

Figure 1: Overview. **Left:** a weight-tied looped transformer applies one shared block T times. **Middle:** trained models solve per-position tasks via a *computation frontier* that advances a fixed number of positions per loop; tail instruments only see the converged region, where they provably saturate, while the algorithm lives in the head of the trajectory. **Right:** the scaling class of the convergence time $\tau(n, i)$ (flat / $\log n$ / linear) identifies the mechanism.

ticipates our tail-saturation analysis, which turns their observation into an instrument-validity constraint. Lens-based studies of Huginn find limited latent-CoT structure (Lu et al., 2025). Kohli (2026) patch hidden states at (position, iteration) granularity in toy recurrent-depth models; step-resolved *data* attribution exists for looped models (Kaissis, 2026). None of these measure when loops carry step-indexed computation; our τ instrument targets exactly that gap. Feature-level circuit tracing (Ameisen et al., 2025) has been extended to diffusion transformers with timestep-conditioned transcoders (DifFRACT authors, 2026), but not to weight-tied loops; our results suggest the loop-mechanism question must be settled first, since attribution graphs inherit the same tail blindness.

Convergence dynamics. Path independence—convergence to input-determined fixed points—predicts upward generalization in deep equilibrium models (Anil et al., 2022); recall-style training explicitly promotes it (Bansal et al., 2022). We adopt the asymptotic-alignment score as a baseline and show the selection of convergence dynamics is a property of the training pressure, not of recurrence per se.

Complexity of transformers. Fixed-depth transformers sit in TC^0 (Merrill & Sabharwal, 2023); constant-depth shortcuts to solvable-group automata exist and are learned, while non-solvable groups (S_5, A_5) admit none unless $\mathrm{TC}^0 = \mathrm{NC}^1$ (Liu et al., 2023; Barrington, 1989). Log-depth looped transformers close the gap via associative scan (Merrill & Sabharwal, 2025), and fixed-depth SSMs fail S_5 state tracking (Merrill et al., 2024). We use this ladder as the hardness axis of the selection grid. Parity is expressible at constant depth (Chiang & Cholak, 2022) yet hard to learn length-generalizably (Hahn & Rofin, 2024)—expressibility and learnability diverge, which is why our claims are about what training *selects*, not what architectures *can do*.

Imposed step-indexing. Iteration-wise supervision aligns loops with CoT steps (Yu, 2026); CLRS-style hints align processor iterations with algorithm steps (Veličković et al., 2022), though hint-free models drift toward parallel strategies (Rodionov & Prokhorenkova, 2023). These are existence proofs that loops *can* be step-indexed when supervision imposes it; our streaming arm imposes it architecturally, and the free arms measure whether anything short of imposition suffices.

3 SETUP

Architecture. A weight-tied looped transformer applies a block F_θ (2 or 4 pre-norm transformer layers) T times to a state $h^t \in \mathbb{R}^{N\times d}$: $h^{t+1} = F_\theta(h^t, e)$, where e is the token embedding, optionally re-injected each loop via a learned adapter (input injection, Bansal et al., 2022). Readout is a linear head on $\mathrm{LN}(h^T)$. Scales: S (2L, d=128), M (2L, d=256), L (4L, d=512). No positional encoding (NoPE) in the main grid; positional-encoding effects are factored in Appendix D.

Tasks. All tasks are per-position sequence-to-sequence, which blocks answer-only shortcuts and gives every intermediate value a ground truth. *Pilot*: prefix parity, and LSB-first increment-with-carry (both constant-depth shortcuttable). *Selection grid*: prefix products over $\mathbb{Z}_{60}$ (abelian), S_4 (non-abelian, solvable), and S_5 (non-solvable; NC^1-complete word problem): input $g_1 \dots g_n$, target $p_i = g_1 \circ \cdots \circ g_i$ at every i. The $S_4/\mathbb{Z}_{60}$ controls separate non-commutativity from non-solvability.

Regimes. *Free*: all tokens visible from loop 0; loop schedules $T = \lceil \log_2 n \rceil + 3$ (jittered) or fixed T; length curriculum (required for S_5, Merrill & Sabharwal, 2025). *Streaming*: token i is revealed—embedded and attendable—only from loop $i-1$ on, so serial computation is forced by construction; $T = n + 2$. Streaming cells are positive controls and metric ceilings, not discoveries: they calibrate instruments and bound what emergence could look like.

4 TAIL INSTRUMENTS SATURATE AT FIXED POINTS

Proposition 1 (Tail blindness, informal)**.** *Let $h^\star$ be a fixed point of the loop map $G(h) = F_\theta(h, e)$ reached at loop $\tau^\star \le T - k$. Then (i) skipping any k loops after $\tau^\star$ leaves the output unchanged; (ii) the local Jacobians $J_t = \partial h^{t+1}/\partial h^t$ agree for all $t \ge \tau^\star$; (iii) any feature dictionary $z = E(h)$ and any attribution graph built from it agree across loops $t \ge \tau^\star$. None of these quantities constrain the computation performed at loops $t < \tau^\star$.*

The proof is immediate from idempotence ($G(h^\star) = h^\star$) and is given in Appendix B, together with the quantitative version for approximate fixed points ($\|G(h) - h\| \le \epsilon$). The consequence is an instrument-validity constraint that, to our knowledge, has not been stated: *any* loop-mechanism claim based on skip tolerance, cross-step similarity, or shared-dictionary attribution is uninformative unless the measurement is localized to the head of the trajectory, $t < \tau^\star$.

A population that tail instruments cannot order. We train 64 looped transformers (2 tasks $\times$ inject $\{0, 1\}$ $\times$ PE $\{\text{learned}, \text{NoPE}\}$ $\times$ T-schedule $\{\text{prop}, \text{fixed}\}$ $\times$ 4 seeds) to in-distribution exact-match accuracy 1.0; the population spans length generalization 0.00–1.00, including a 0.10–0.95 spread *across seeds of a single recipe*. Every tail instrument fails to order it: single-step skip retains every solved example in all 50 models with enough solved mid-OOD inputs (retention exactly 1.0000); cross-step Jacobian alignment spans 0.85–0.99 and hidden-cosine / attention similarity carry no relation to generalization (all tie-corrected $|\rho| \le 0.32$; per-recipe details in Appendix C). By Proposition 1 this is the expected reading for trajectories that converge with slack—and measured convergence times sit well inside the loop budget for all models. The pilot's information is not "these models have no algorithm"; it is "these instruments cannot say."

5 THE CONVERGENCE-TIME INSTRUMENT

Definition 1 (Convergence-time scaling)**.** *For a solved input of length n, let $\hat{y}_i^t$ be the label decoded from h^t at position i. Define $\tau(n, i) = \min\{k : \hat{y}_i^t = \hat{y}_i^T \ \forall t \ge k\}$, and $\tau_{\max}(n) = \mathrm{median}_{\text{inputs}} \max_i \tau(n, i)$. The* scaling class *of a model is the growth of $\tau_{\max}(n)$: flat (shortcut/refinement), $\Theta(\log n)$ (parallel scan), or $\Theta(n)$ (serial).*

τ is forward-only (no gradients, no dictionaries, no patching), is defined on solved inputs only (decoupling mechanism from competence), and measures the head of the trajectory by construction: it reports when the *decodable output* stabilized, which upper-bounds when the underlying computation finished but is not identical to it—the causal cones of §6 are what license reading τ mechanistically, and we restrict mechanism claims to cone-validated models. Per-position curves $\tau(n, \cdot)$ further localize *where* the frontier of computation moves across loops, and on group tasks every intermediate p_i has ground truth, so head-of-trajectory decoding can be scored exactly.

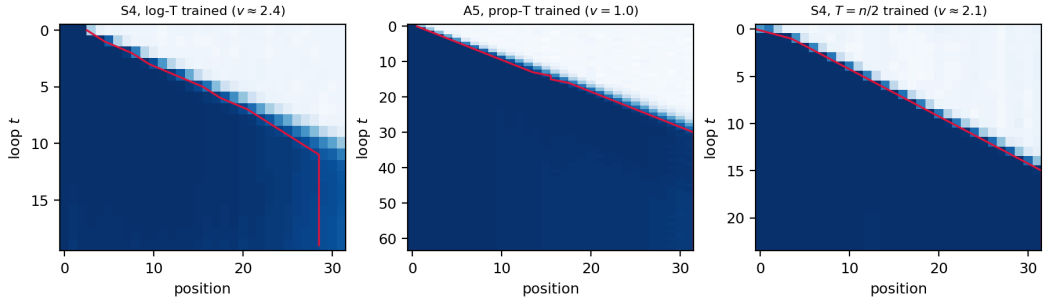

Figure 2: Per-(loop, position) accuracy heatmaps at n=32 (blue: solved; red line: contiguous solved prefix). Free training discovers a *linear computation frontier*: each loop extends the solved prefix by a constant number of positions. The slope is a per-model constant across lengths—a mechanism fingerprint—and is set by the training budget: tight log-T training yields $v \approx 2.4$ (left), T=n training yields exactly v=1.0 (middle), T=$n/2$ yields $v \approx 2.1$ (right).

Calibration. The instrument must fire on a known serial mechanism and stay flat on a known shortcut. On streaming parity (seriality forced by construction) it reads slope $d\tau_{\max}/dn = 1.00$. On fixed-T parity models (bounded budget) it reads 0.39–0.41 locally within their narrow solved range, bounded by T. Learned-PE models, which hit the positional wall, are correctly reported as out of instrument range rather than misclassified.[1]

A first head-of-trajectory signal. On the pilot's seed-only recipe (parity, proportional T), τ slopes order with generalization where every tail metric reads flat—the signal quantified at population scale in §8.

6 The Linear Computation Frontier

Free training discovers frontier algorithms. On S_4 and A_5 prefix products, every model that learns the task implements a moving computation frontier (Fig. 2): the contiguous solved prefix advances v positions per loop, with v essentially constant across input lengths within a model (Table 1). This is loop-indexed computation in the head of the trajectory—not the constant-depth shortcut the solvable groups admit, not a $\log n$ scan, and not fixed-point refinement. Because the frontier is linear, budget-starved models fail *late positions* rather than whole inputs, and granting more loops than ever seen in training rescues them per-position (Table 2): accuracy at late positions of in-distribution lengths goes from near-zero at the training budget to near-perfect at roughly twice the budget. Unlike prior loop-extrapolation results on longer *inputs*, this is rescue at *fixed* length, position-resolved, and quantitatively predicted by $T \geq \tau(n, i)$. Table 2 also exposes the other side of the coin: pushing T far beyond frontier completion *degrades* tight-budget models (overthinking), while loose-contract models remain stable—the safe operating range of test-time compute is itself a contract property.

The budget law. Frontier speed is set by the training contract (Fig. 4, Table 1): across contracts spanning an $8\times$ range of demand, the measured v tracks n_{tr}/T_{tr} with a power-law exponent of 0.98 ± 0.04 (95% CI; R^2=0.99 over 23 models at 8 demand levels; Fig. 6, left)—i.e. speed equals demand, exactly unity under the T=n contract—so SGD consistently arrives at a frontier whose speed matches the *minimum the contract demands*, and v is an experimenter-controllable knob (Fig. 3, left). We state this as a robust empirical regularity rather than a selection theorem: demonstrating that faster-but-equally-trainable solutions exist at the same contract (and are avoided) would require a constructive search over solutions we have not performed; what the data establish is that the demanded minimum is met, never exceeded, across an $8\times$ range of consistent demands. Two refinements. First, under the *mixed* demands of the log-T schedule, models compromise near the middle of the demand range and under-deliver at the long end; a *consistent* high demand (T=$n/4$ at every length) is met even at the smallest scale (Table 1)—the apparent speed ceiling is a product of demand inconsistency, not capacity. Second, scale buys length robustness more than speed:

[1] Complete calibration protocol, including the frontier-resolved skip curve that replaces single-step skip (skip k consecutive loops at every offset, stratified by distance to measured τ), in Appendix C.

Table 1: Frontier speed and uniformity by training contract. v: median frontier speed over $n \in \{16, 32, 64\}$ [seed range]; uniformity: median fraction of n=64 solved by the frontier at 2.5× the schedule budget. A_5 pools all seeds including non-cracking ones (bimodal; §8).

Group	Contract (demand n/T)		runs	v	unif.@64
S_4 S	T=$2n$	(0.5)	4	0.51 [.51, .52]	.55
S_4 S	T=n	(1)	2	1.00 [1.00, 1.00]	.98
S_4 S	T=$n/2$	(2)	4	2.00 [1.93, 2.13]	.19
S_4 S	T=$n/4$	(4)	2	4.00 [3.60, 4.00]	.00
S_4 S	log-T	(mixed)	16	2.00 [1.86, 2.42]	.33
S_4 M	log-T	(mixed)	4	1.88 [1.86, 1.88]	.41
S_4 L	log-T	(mixed)	4	2.73 [2.09, 3.33]	.09
A_5 S	T=n, horizon	(1)	16	0.96 [0.08, 1.03]	—
S_5 M	op.-first, T=n	(1)	2	1.00	.56
$\mathbb{Z}_{60}$ S	log-T	(mixed)	8	2.15 [1.82, 5.17]	—
ETH parity	T=n	(1)	4	≈ 1.5	—

Table 2: Rescue surface and overthinking: mean per-token accuracy vs. test-time loops T (group means over first three seeds). Tight-budget (log-T) groups are rescued by T beyond their training budget and then degrade; the loose-contract operator-first S_5 improves monotonically to frontier completion; $\mathbb{Z}_{60}$ and annealed models collapse sharply past their operating range.

		test loops T						
Group	n	4	8	12	16	24	32	48
S_4 log S	32	.38	.69	.95	**.97**	.91	.85	.78
S_4 log S	64	.21	.37	.52	.63	**.66**	.60	.55
S_4 log L	32	.38	.79	**.96**	.86	.74	.69	.64
S_5 op-first M	32	.13	.26	.38	.51	.78	**.99**	—
S_5 op-first M	64	—	.13	.19	.26	.38	.51	.64
anneal A_5 p→l	32	.35	.61	**.76**	.76	.52	.26	.13
$\mathbb{Z}_{60}$ log S	32	.53	**.75**	.74	.74	.45	.12	.05
ETH prop	32	.52	.63	.69	.73	.75	**.76**	.75

larger blocks run their frontiers slightly faster but hold them together at longer lengths (Table 2), a speed–robustness trade visible in the uniformity column of Table 1.

Cross-task form of the law. Replicating the demand sweep on three further tasks yields the law's complete form: v tracks demand within a task-dependent operating band, $v \approx \mathrm{clip}(n_{tr}/T_{tr},\ v_{\mathrm{free}},\ v_{\max})$. On A_5 the law is exact across an 8× demand range ($0.5 \to 0.48$, $1 \to 1.00$, $2 \to 1.90$) with mild under-delivery at the top ($4 \to \approx 2.5$: the expensive 60×60 operator caps $v_{\max}$). Parity floors: measured v never drops below ≈ 1–1.5 however loose the contract, because attention composes cheap XOR operators over more than one position at no extra cost (v_{free})—slowing down further would not be simpler, so simplicity bias has nothing to push against; this is also why loose-contract parity models carry idle tail loops (the slack that saturates tail instruments, §4). $\mathbb{Z}_{60}$ obeys the law at loose contracts but *escapes* under pressure ($2 \to 4.2$, $4 \to 8.0$): the abelian task has a parallel shortcut, and a tight contract pushes the model off the frontier class entirely rather than to a faster frontier. The three regimes—law-bound (A_5, S_4), floored (parity), escaped ($\mathbb{Z}_{60}$)—share one reading: training pressure sets the mechanism's speed exactly when the task forbids anything cheaper.

The frontier is causal. Resampling the state at (position i, loop t) and rolling out the remaining loops yields damage that is strictly downstream-in-position (zero upstream damage in all 72 grid cells for both models tested), structured by the frontier: positions already passed by the frontier are immune to upstream corruption, pending positions inherit it, and the boundary between the two moves at each model's measured v (Fig. 4, right). τ is therefore a readout of a causal computational object, not a decoding artifact. The cones also reveal two storage phenotypes (Table 3): one model class *repairs* corrupted stored values within a loop (negligible self-damage behind the frontier), while the rest are store-once (near-total, permanent self-damage)—a repair axis orthogonal to speed, and notably rare: of the frontier models probed, only the star A_5 solution self-heals.

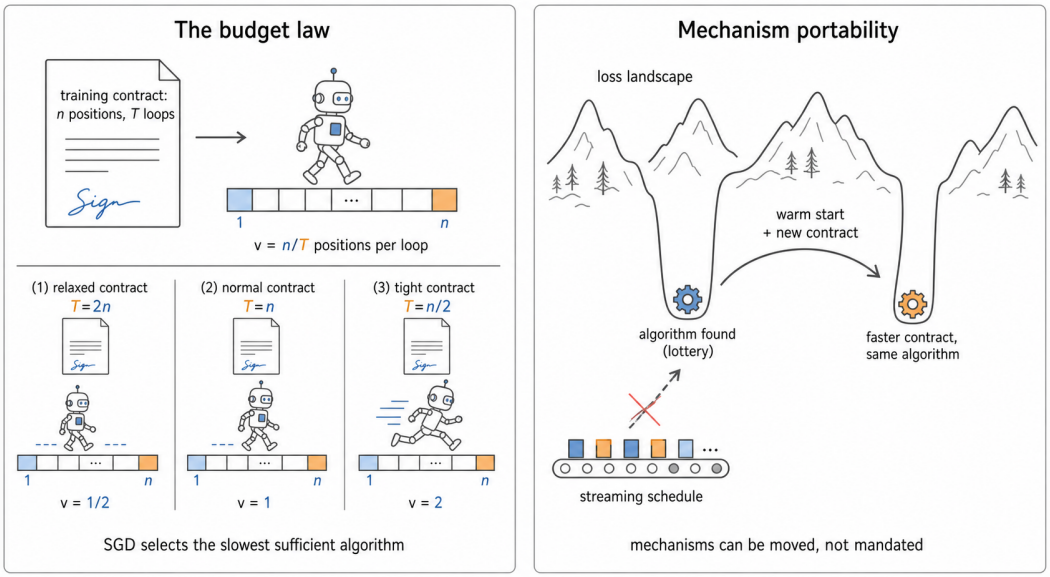

Figure 3: Two organizing findings. **Left:** the budget law—the training contract prices the frontier speed; the demanded minimum is met, never exceeded. **Right:** mechanism portability—the algorithm is found by a basin lottery, transfers across contracts by warm-starting, and cannot be installed through the input schedule.

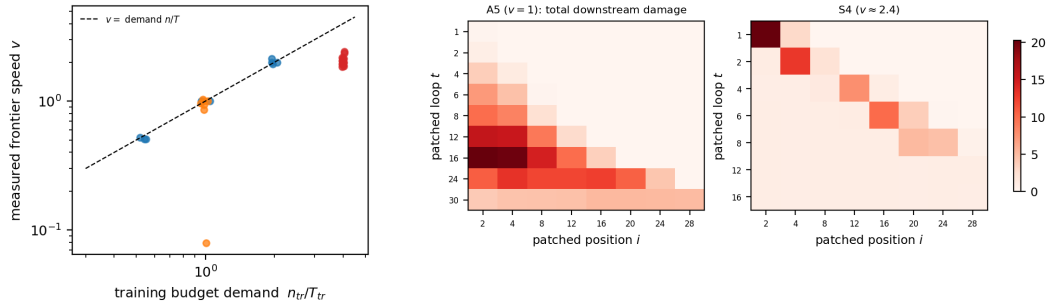

Figure 4: **Left (budget law):** measured frontier speed tracks the training demand n_{tr}/T_{tr} (dashed: $v =$ demand) for consistent demands; the log-T cell (red, mixed demands averaging 4) compromises at $v \approx 2$. **Right (causal cones):** total downstream damage from patching state at (position i, loop t). Damage is strictly downstream (zero upstream in every cell), tracks the moving frontier, and the cone slope reproduces each model's v independently.

What limits the frontier: operator scale, not circuit class. S_5 (NC^1-complete) is never learned by free training under the standard length curriculum—but the wall has nothing to do with non-solvability (Fig. 5). It is already present at n=2: the time to master the single composition $g_1 \circ g_2$ grows steeply with group order and, for S_5 at the smallest scale, exceeds the entire training budget (Table 4); one scale up, the same operator is learned comfortably. The matched-order contrast is decisive: A_5 is exactly as NC^1-hard as S_5, yet prop-T training chains it to full length generalization at twice the training length (Table 7, top). And the wall is a *curriculum* pathology, not an impossibility: an operator-first curriculum—an extended n=2 stage that masters the composition table before any chaining—cracks S_5 in *every* seed attempted ($10/10$ across two machines, exact match ≥ 0.92 at $2\times$ training length). Joint learning of the operator and the chain deadlocks; sequenced learning does not, reliably. Non-solvability blocks nothing at these scales; the order of the group sets the optimization cost of its operator, and curricula settle it.

Imposed seriality does not help. Streaming input release (token i revealed at loop i), the by-construction serial schedule, trains perfectly on parity—with length generalization to four times the

Table 3: Repair phenotype: mean self-damage when the stored value at a position already passed by the frontier is resampled (grid cells with $t > i$). Low = self-healing; high = store-once.

	A_5 $T{=}n$	S_4 log	S_4 $T{=}n/2$	S_5 op-first
self-damage behind frontier	.02	.95	.96	.99

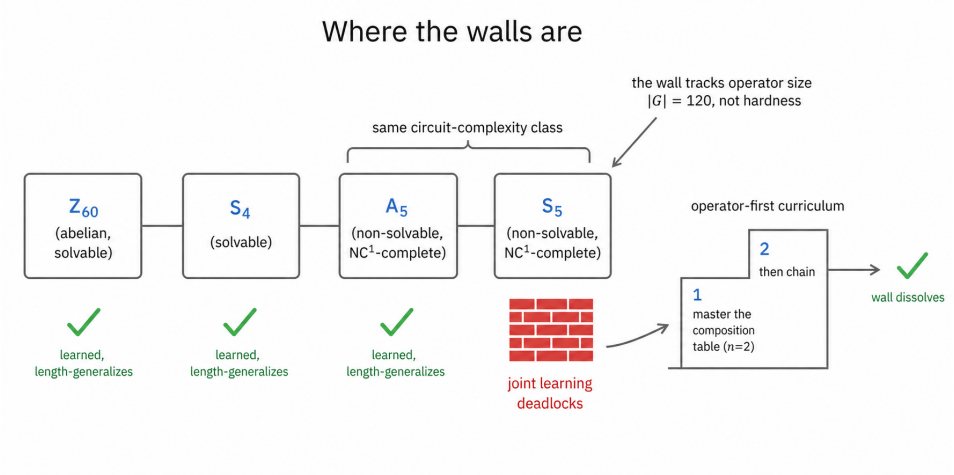

Figure 5: The learning walls track operator size, not circuit-complexity class: the non-solvable A_5 trains like the solvable groups, its complexity-class twin S_5 deadlocks, and an operator-first curriculum dissolves the deadlock.

training length—but *fails* on every group task including A_5, which free scheduling solves (Table 5). Adding per-loop supervision on the revealed prefix does not rescue it. The frontier that free training invents is easier to learn than externally imposed seriality: mechanism selection cannot be forced through the input schedule, which motivates the intervention that works (§7).

Controls: the tying axis isolated. We train a tying-degree ladder at fixed effective depth 12: the D layers are drawn from G weight groups cycled ($G{=}1$: our fully tied loop; $G{=}12$: a standard untied transformer; intermediate G: partial tying), in two regimes—constant width (parameters grow $10\times$ with G) and parameter-matched (width shrunk to hold parameters fixed) (Table 6). The regimes disagree, and the disagreement is the point: with free parameters, untying appears mildly beneficial; with parameters matched, the fully untied model is *worst* at length extrapolation, and on A_5 every configuration with $G \geq 3$ fails to learn the task at all ($9/9$ seeds) while tied configurations learn it. Weight reuse is not a resource handicap but a load-bearing inductive bias—necessary for learnability on the harder group, and for extrapolation on the easier one—consistent with standard transformers learning parallel-scan mechanisms on this family (Liu et al., 2023). Learned absolute positions (giving the model the addressing a log-depth scan would need) leave the mechanism unchanged: the frontier persists, with the same rescue signature.

7 Annealing the budget transfers the mechanism

If the frontier is real and its speed is priced by the budget, mechanisms should transfer across budgets by warm-starting—and transfer should bypass the seed lottery that afflicts training from scratch (across 22 seeds on two machines, roughly a third crack chained A_5; the rest never pass the second position—a bimodal outcome stable across hardware and run order). Both predictions hold, in every seed attempted (Table 7). Warm-starting from a $T{=}n$ solution and re-training under the tight log-T contract meets the new contract; the reverse anneal fully solves lengths that no from-scratch run of either schedule achieved. Most tellingly, the transferred mechanism is *re-priced*: annealing a $v{=}1$ solution into the tight contract doubles its measured frontier speed, and the reverse anneal slows a fast frontier back toward unity (Table 7)—the budget law acts on warm-started mechanisms just as it does on fresh ones, but without re-running the lottery. Once any budget regime finds the algorithm,

Table 4: Optimization cost of the group operator: training steps to per-token accuracy $\geq .98$ on pair composition (n=2). Cost grows steeply with group order $|G|$; one scale up (M), the S_5 operator that never converges at S is learned in 6.5k steps.

	S_4 ($\|G\|$=24)	$\mathbb{Z}_{60}$	A_5 ($\|G\|$=60)	S_5 ($\|G\|$=120)
scale S	∼0.5k	2.5k	∼4k	>40k (never)
scale M	0.5k	2.0k	1.8k	6.5k

Table 5: Mechanism cannot be mandated through the input schedule: outcome by task × schedule (✓ = learns and length-generalizes; × = fails to learn beyond trivial positions).

	free	streaming	streaming + per-loop sup.
parity	✓	✓	—
A_5	✓ (lottery)	×	×
S_5	× (op. wall)	×	×
S_5, operator-first	✓	—	—

the algorithm—not the budget—is the hard-won object, and it retrains cheaply into any contract (Fig. 3, right). The annealed models inherit the tight contract's narrow operating range (Table 2), a caveat for deployment.

A halting rule falls out of the law. Because v is estimable in distribution and the frontier completes at $T \approx n/v$, the budget law is directly actionable: stopping at $T^\star = \lceil n/\hat{v} \rceil$ (with $\hat{v}$ read off training lengths) recovers near-oracle accuracy at test lengths and avoids overthinking (Fig. 6, right)—e.g. on a tight-budget S_4 model at n=48 the training schedule's fixed T scores far below the rule, which nearly matches the best achievable over a full T sweep. Test-time compute for looped models has a principled setting, not a hyperparameter to tune.

8 Head instruments predict generalization; tail instruments do not

We race metrics measured *strictly in distribution* ($n \leq n_{tr}$=32: frontier speed at n=16 and 32, frontier completion at n=32 under the schedule budget, τ slope over ID lengths, and the tail baselines) against length generalization measured *strictly out of distribution* (mean per-token accuracy at $n \in \{64, 128\}$, schedule-matched T, recomputed from checkpoints), across three 16-seed single-recipe populations—no recipe confounds, no predictor sees any length beyond training, tie-aware Spearman, partial correlations controlling adjacent-loop cosine, and median-split AUROC.

In every cell the best predictor is a head instrument (Table 8), under a protocol in which no predictor sees any length beyond training. The consistent picture: frontier *completion at the training frontier*—did the mechanism finish its work within the contract, measured entirely in distribution—is the dominant predictor everywhere it is defined (AUROC up to 1.00), and ID frontier *speed* adds signal exactly where the budget law says it should: in the tight-budget cell, where speed is under pressure and varies across seeds, but not in loose-contract cells, where the contract pins it. Tail metrics are not uniformly blind—in the sharply bimodal A_5 cell, adjacent cosine separates "still computing" from "collapsed"—but they carry no signal *within* a mechanism class, where head instruments retain it. The path-independence score never wins a cell. Failed seeds having no measurable frontier is itself the strongest ID signal in the bimodal cells: the presence of a frontier at training lengths is close to a sufficient statistic for OOD fate. Where the negative Jacobian correlations are significant, they point the same way: seeds whose loop maps align early (converge early) generalize worse.

9 Open-benchmark validation

On the community easy-to-hard prefix-sums benchmark (Schwarzschild et al., 2021) (official 32-bit training split, official 512-bit test set), the same phenomena replicate without modification (Table 9). Under the T=n contract, successful seeds learn a linear frontier with speed constant across lengths

Table 6: Tying-degree control at fixed effective depth 12 (G weight groups cycled; mean per-token accuracy at n=64, 3 seeds per cell). With parameters matched, untying hurts extrapolation (S_4) and destroys learnability (A_5, $G \geq 3$ at chance).

		G=1 (tied)	G=2	G=3	G=4	G=6	G=12 (untied)
S_4	const. width	.21	.27	.30	.30	.32	.26
	param-matched	.21	**.28**	.27	.25	.25	**.15**
A_5	const. width	.10	.23	.17	.25	.23	.15
	param-matched	.10	**.15**	**.05**	.05	.05	.05

Table 7: Budget annealing: warm-start across loop contracts. "Cracked" = learns chained composition with length generalization. The frontier speed of the transferred mechanism is re-priced by the new contract.

Arm	contract	cracked	v before	v after
A_5 from scratch	T=n	8/22 seeds	—	≈ 1.0
A_5 from scratch	log-T	bimodal, weaker	—	≈ 2
A_5 anneal	T=$n \to$ log	4/4	1.0	2.0
S_4 anneal	T=$n \to$ log	4/4	1.0	2.4
S_4 anneal	log $\to$ T=n	4/4	2.4	1.2
S_5 operator-first	T=n	10/10	—	1.0

and extrapolate to four times the training length at near-perfect *exact-match* accuracy; the seed lottery replicates; and the budget law predicts the log-T arm's total failure at 512 bits exactly (demand more than an order of magnitude above the learned speed). The frontier breaks between 128 and 256 bits (Table 9), well short of the 512-bit extrapolation that convolutional Deep Thinking networks achieve on this dataset (Bansal et al., 2022)—an honest asymmetry with a mechanistic reading: convolutional locality *forces* a linear frontier and shields it from long-range interference, whereas the frontier our models *learn* inside global attention inherits attention's length fragility. The mechanism class transfers to the open benchmark; its range is architecture-limited. We emphasize this section is mechanism validation, not a state-of-the-art generalization claim—convolutional architectures remain stronger on this benchmark, for the architectural reason above.

10 Discussion

What the loop buys. The loop's contribution is neither "more refinement" nor "more algorithm steps" generically: free training installs a linear frontier whose speed is priced by the training contract, whose robustness varies by seed along an orthogonal repair axis, and whose completion—not convergence—is what test-time loops purchase. This reframes halting and loop-extrapolation design: the useful stopping signal is frontier completion ($T \geq n/v$), predictable per input length, rather than state convergence. It equally reframes interpretability for depth-recurrent LMs: instruments that probe converged states inherit Proposition 1's blindness, and should be replaced or augmented by head-of-trajectory measurements.

Localization, transport, and editing. Appendix A tests our claims against the localization literature's standards: frontier-state transplants succeed or fail exactly as the repair phenotype predicts, module-resolved cones place the frontier's routing in the first attention layer, and the localization-vs-editing dissociation of Hase et al. (2023) becomes an architectural theorem under weight tying, whose smearing signature we measure. All our localization claims accordingly live, and are validated, in activation space.

Selection, not expressivity. Standard-depth transformers learn parallel scans on this task family and fail to find sequential mechanisms (Liu et al., 2023); our untied control degrades similarly. Weight tying flips the selection to a serial frontier even when positional addressing for a log-depth scan is supplied. Which algorithm SGD finds is set by the architecture prior and the budget contract—not by what the architecture can express.

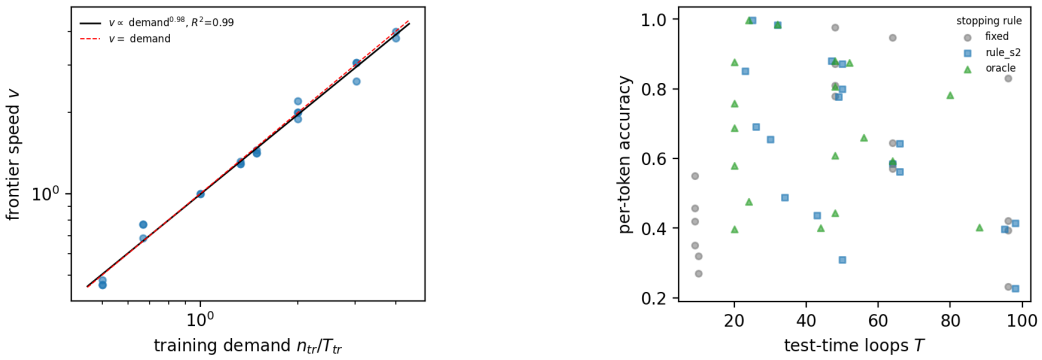

Figure 6: **Left:** the budget law densified—frontier speed vs. training demand over 23 models spanning $8\times$ in demand; the fit is $v \propto \text{demand}^{0.98\pm0.04}$, R^2=0.99 (dashed: $v = \text{demand}$). **Right:** the law as a halting rule—stopping at $T^\star = \lceil n/\hat{v} \rceil$ with $\hat{v}$ estimated in distribution (blue) nearly matches an oracle T sweep (green) and far exceeds the training schedule's own fixed T (grey), while avoiding the overthinking collapse of over-long T.

Table 8: Predicting OOD generalization ($n \in \{64, 128\}$) from strictly in-distribution measurements ($n \leq 32$), 16 seeds per cell. Entries: Spearman ρ (median-split AUROC). Head instruments: frontier speed v_{32} and frontier completion at n=32 under the schedule budget; tail baselines: adjacent-loop cosine, Jacobian alignment; AA = asymptotic alignment (Anil et al., 2022). A dash marks a metric that is *undefined* for that cell, not a missing run: on A_5, speed v_{32} has no seed-level variance to correlate against—the seeds that build a frontier all run it at $v \approx 1$ (the T=n contract), and the seeds that fail never form one, so the binary "is there a frontier" is already captured by completion@32 (AUROC 1.00); on A_5, Jacobian alignment is likewise degenerate under the median split.

Metric [ID, $n \leq 32$]	S_4 log-T	A_5 prop-T	parity prop-T
frontier speed v_{32} [head]	$+.\mathbf{87}$ (.95)	—	$+.38$ (.62)
frontier completion@32 [head]	$+.83$ (.94)	$+.\mathbf{90}$ (1.00)	$+.\mathbf{58}$ (.84)
cosine (adjacent) [tail]	$-.02$ (.44)	$-.77$ (.03)	$+.01$ (.37)
Jacobian align. [tail]	$-.61$ (.10)	$-.79$	$+.11$ (.44)
AA / path indep.	$+.31$ (.60)	$-.66$ (.14)	$+.21$ (.65)

Scope. Claims concern what end-to-end training selects in weight-tied loops on per-position algorithmic sequence tasks at $\leq$ 25M parameters. The S_5 result is a statement about optimization cost of the composition operator at $|G|$=120, not an NC^1 impossibility claim; the $\text{TC}^0 \neq \text{NC}^1$ separation is conditional and forces only super-constant depth. Pretrained depth-recurrent LMs (Huginn, Ouro) are out of scope here; extending τ to them requires resolving tokenization and format confounds and is left to future work.

REFERENCES

Emmanuel Ameisen, Jack Lindsey, Adam Pearce, et al. Circuit tracing: Revealing computational graphs in language models. transformer-circuits.pub, 2025.

Cem Anil, Ashwini Pokle, Kaiqu Liang, Johannes Treutlein, Yuhuai Wu, Shaojie Bai, J. Zico Kolter, and Roger Grosse. Path independent equilibrium models can better exploit test-time computation. *NeurIPS*, 2022.

Arpit Bansal, Avi Schwarzschild, Eitan Borgnia, Zeyad Emam, Furong Huang, Micah Goldblum, and Tom Goldstein. End-to-end algorithm synthesis with recurrent networks: Extrapolation without overthinking. *NeurIPS*, 2022.

David A. Barrington. Bounded-width polynomial-size branching programs recognize exactly those languages in nc^1. *Journal of Computer and System Sciences*, 38(1):150–164, 1989.

Table 9: Easy-to-hard prefix sums (train 32-bit official split): exact-match accuracy by test length, per seed, $T{=}n$ contract; log-T contract fails everywhere beyond training length as the budget law predicts.

Arm	$n{=}32$	$n{=}64$	$n{=}128$	$n{=}256$	official 512
$T{=}n$, seed 2	1.00	1.00	1.00	0	0
$T{=}n$, seed 3	1.00	1.00	0.96	0	0
$T{=}n$, seeds 0/1 (lottery)	1.00	0	0	0	0
log-T, all seeds	1.00	0	0	0	0
DT conv-RNN (Bansal et al., 2022)	solves 512 (locality-forced frontier)				

et al. Blayney. A mechanistic analysis of looped reasoning language models. *arXiv preprint arXiv:2604.11791*, 2026.

ByteDance Seed. Ouro: Looped language models. *arXiv preprint arXiv:2510.25741*, 2025.

David Chiang and Peter Cholak. Overcoming a theoretical limitation of self-attention. *ACL*, 2022.

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. *NeurIPS*, 2023.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. Knowledge neurons in pretrained transformers. *ACL*, 2022.

Mostafa Dehghani, Stephan Gouws, Oriol Vinyals, Jakob Uszkoreit, and Łukasz Kaiser. Universal transformers. *ICLR*, 2019.

DifFRACT authors. Diffract: Diffusion feature reconstruction and attribution for circuit tracing. *arXiv preprint arXiv:2606.15796*, 2026.

Ying Fan, Yilun Du, Kannan Ramchandran, and Kangwook Lee. Looped transformers for length generalization. *arXiv preprint arXiv:2409.15647*, 2024.

Khashayar Gatmiry, Nikunj Saunshi, Sashank J. Reddi, Stefanie Jegelka, and Sanjiv Kumar. Can looped transformers learn to implement multi-step gradient descent for in-context learning? *ICML*, 2024.

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

Michael Hahn and Mark Rofin. Why are sensitive functions hard for transformers? *ACL*, 2024. arXiv:2402.09963.

Peter Hase, Mohit Bansal, Been Kim, and Asma Ghandeharioun. Does localization inform editing? surprising differences in causality-based localization vs. knowledge editing in language models. *NeurIPS*, 2023.

et al. Kaissis. Step-resolved data attribution for looped transformers. *arXiv preprint arXiv:2602.10097*, 2026.

et al. Kohli. Loop, think, & generalize: Implicit reasoning in recurrent-depth transformers. *arXiv preprint arXiv:2604.07822*, 2026.

Bingbin Liu, Jordan T. Ash, Surbhi Goel, Akshay Krishnamurthy, and Cyril Zhang. Transformers learn shortcuts to automata. *ICLR*, 2023.

Wenquan Lu, Yuechuan Yang, Kyle Lee, Yanshu Li, and Enqi Liu. Latent chain-of-thought? decoding the depth-recurrent transformer. *COLM Workshop on LLM Explainability for Reasoning and Planning*, 2025. arXiv:2507.02199.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. *NeurIPS*, 2022.

William Merrill and Ashish Sabharwal. The parallelism tradeoff: Limitations of log-precision transformers. *TACL*, 2023.

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

William Merrill, Jackson Petty, and Ashish Sabharwal. The illusion of state in state-space models. *ICML*, 2024. arXiv:2404.08819.

Gleb Rodionov and Liudmila Prokhorenkova. Neural algorithmic reasoning without intermediate supervision. *NeurIPS*, 2023.

Avi Schwarzschild, Eitan Borgnia, Arjun Gupta, Furong Huang, Uzi Vishkin, Micah Goldblum, and Tom Goldstein. Can you learn an algorithm? generalizing from easy to hard problems with recurrent networks. *NeurIPS*, 2021.

Tianyi Tang, Wenyang Luo, Haoyang Huang, Dongdong Zhang, Xiaolei Wang, Wayne Xin Zhao, Furu Wei, and Ji-Rong Wen. Language-specific neurons: The key to multilingual capabilities in large language models. *ACL*, 2024.

Eric Todd, Millicent L. Li, Arnab Sen Sharma, Aaron Mueller, Byron C. Wallace, and David Bau. Function vectors in large language models. *ICLR*, 2024.

Petar Veličković, Adrià Puigdomènech Badia, David Budden, Razvan Pascanu, Andrea Banino, Misha Dashevskiy, Raia Hadsell, and Charles Blundell. The clrs algorithmic reasoning benchmark. *ICML*, 2022.

Xiaozhi Wang, Kaiyue Wen, Zhengyan Zhang, Lei Hou, Zhiyuan Liu, and Juanzi Li. Finding skill neurons in pre-trained transformer-based language models. *EMNLP*, 2022.

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

Yunzhi Yao, Ningyu Zhang, Zekun Xi, Mengru Wang, Ziwen Xu, Shumin Deng, and Huajun Chen. Knowledge circuits in pretrained transformers. *NeurIPS*, 2024.

et al. Yu. Enhancing auto-regressive chain-of-thought through loop-aligned reasoning. *EACL*, 2026. arXiv:2502.08482.

A Localization, transport, and editing under weight tying

The survey lines that dominate interpretability—neuron localization (Dai et al., 2022; Wang et al., 2022; Tang et al., 2024), causal-tracing-guided editing (Meng et al., 2022), and circuit discovery (Conmy et al., 2023; Yao et al., 2024)—all index structure by *parameter address* (layer, neuron, head). Weight tying voids that index: one address serves every loop. Three experiments show what replaces it.

State transplant follows the repair phenotype. Splicing input A's frontier states into input B's run at matched (position, loop)—the state-level analogue of function-vector transport (Todd et al., 2024)—has opposite fates in the two storage phenotypes of Table 3: in the store-once S_4 model the transplanted prefix *sticks* (retained nearly perfectly, essentially never reverting), while the self-healing A_5 model *repairs* the transplant away, reverting almost every position to its own input's answer (Table 10). Together with §7 this completes a three-way separation: *states* transport where storage is passive; the *mechanism* transports universally in weight space (annealing); and *schedules* transport nothing.

Module-resolved cones. Splitting the (position, loop) patch by module localizes the frontier's machinery within the tied block: the first layer's attention carries nearly all cross-position damage (an order of magnitude above the second layer's, which is essentially inert), with both MLPs

Table 10: Frontier-state transplant (A's prefix states spliced into B's run at the frontier, loop t=16): retention vs. repair is governed by the storage phenotype of Table 3.

Model	keeps A prefix	reverts to B	phenotype
S_4 log-T	.97–1.00	.05	store-once
A_5 T=n	.02	.99	self-healing

intermediate—the routing that advances the frontier lives in one attention layer, a module-level claim no fixed-depth circuit method can even express, since its nodes are parameter addresses rather than (module, loop) pairs.

The localization–editing dissociation is architectural. Hase et al. (2023) showed that causal-tracing localization does not predict where weight edits succeed. Under weight tying this dissociation is a theorem—any weight edit hits every loop—and we measure its signature: perturbing FFN rows selected by gradient attribution at a target band of loops produces damage that accumulates monotonically with loop index and shows *zero* band specificity (inside-band and outside-band damage statistically identical). Activation-space localization cannot inform weight-space editing here even in principle; this is why all our localization claims live, and are validated, in activation space (§5). Conversely, it sharpens what our instruments mean: τ and the cones localize *when and where computation happens*, a question that remains well-posed precisely because it is not a claim about parameter storage.

B TAIL BLINDNESS: FORMAL STATEMENT

Proposition 2 (Exact form)**.** *Let $G : \mathbb{R}^{N\times d} \to \mathbb{R}^{N\times d}$ be the loop map $G(h) = F_\theta(h, e)$ for a fixed input embedding e, and suppose the trajectory $h^{t+1} = G(h^t)$ reaches a fixed point: $h^t = h^\star$ for all $t \geq \tau^\star$, with $\tau^\star \leq T - k$. Then: (i) for any $j \geq \tau^\star$ and any $k \leq T - j$, skipping loops $j, \ldots, j+k-1$ (i.e., outputting $G^{T-j-k}(h^j)$ in place of $G^{T-j}(h^j)$) leaves the final state, and hence the decoded output, unchanged; (ii) $J_t := \partial G/\partial h\big|_{h^t} = J_{\tau^\star}$ for all $t \geq \tau^\star$, so any similarity functional of local Jacobians is constant on the tail; (iii) for any feature encoder E and any attribution graph constructed from activations $z^t = E(h^t)$ and linearizations at h^t, all tail-indexed quantities coincide for $t \geq \tau^\star$. None of (i)–(iii) constrains $\{h^t\}_{t<\tau^\star}$ or the map's behavior off the trajectory.*

Proof. (i) $G(h^\star) = h^\star$ implies $G^m(h^j) = h^\star$ for every $m \geq 0$ when $h^j = h^\star$; both branches output $h^\star$. (ii)–(iii) are immediate since all quantities are functions of the (constant) tail state. The final claim holds because the statements quantify only over $t \geq \tau^\star$. □

Proposition 3 (Approximate form)**.** *If instead $\|G(h^t) - h^t\| \leq \epsilon$ for $t \geq \tau^\star$ and G is L-Lipschitz on a neighborhood containing the tail, then skipping k tail loops perturbs the final state by at most $\epsilon \sum_{i=0}^{k-1} L^{T-\tau^\star-i}$, and tail Jacobians differ by at most $\mathrm{Lip}(J) \cdot \epsilon/(1-L)$ when $L < 1$. For contractive tails ($L < 1$) both bounds are $O(\epsilon)$: near-converged trajectories are indistinguishable from converged ones at instrument precision.*

Empirically, all 64 pilot models show relative update norms $< 10^{-2}$ over the final third of the loop budget on solved inputs. Small update norms are consistent with (though do not strictly prove) a contractive tail; we therefore treat the proposition as explaining the observed saturation rather than as a verified premise, and note that models outside the near-converged regime (e.g., the diverging $\mathbb{Z}_{60}$ models of Table 2) are explicitly outside its scope—there, tail instruments can be informative.

C CALIBRATION PROTOCOL AND PILOT DETAILS

τ calibration. The instrument must fire on a known serial mechanism and stay bounded on known shortcuts. Streaming parity (seriality by construction): $d\tau_{\max}/dn = 1.00$. Fixed-T parity (budget-bounded): local slopes 0.39–0.41, correctly capped by T. Proportional-T parity: slopes 0.50–0.70, ordered with seed-level generalization. Learned-PE models, which fail beyond training length, yield

fewer than three solved lengths and are reported as out of instrument range rather than assigned a class. τ requires solved inputs at ≥ 3 lengths; for tight-budget models that fail exact match at long n, frontier speed v (measured per length) substitutes, as in §8.

Frontier-resolved skip. The single-step skip of the pilot is replaced by: skip $k \in \{1, 2, 4\}$ consecutive loops at every start offset, retention measured per example on solved inputs at the largest solved length, stratified by offset relative to the measured frontier. Frontier models show retention 1.0 only for offsets past frontier completion; refit-style models show it everywhere.

Pilot population. 64 models (Appendix D for recipes), all at in-distribution exact match 1.0. Tail readings: single-step skip retention exactly 1.0000 for all 50 models with ≥ 8 solved examples at n=24; adjacent-loop Jacobian probe alignment 0.85–0.99; hidden cosine 0.59–0.94; attention similarity 0.76–0.99. Tie-corrected Spearman against length generalization, within task: all $|\rho| \leq 0.32$ (cosine ≤ 0.23), within-recipe seed contrasts flat. Convergence-time slopes over the same population span 0.39–0.70 and order with generalization in the seed-only recipe (§8).

D TRAINING AND EXPERIMENT DETAILS

Architecture. Pre-norm transformer blocks, $4d$ FFN, GELU; weight-tied across loops; input injection via a learned linear adapter on $[h; e]$; readout head on $\mathrm{LN}(h^T)$. Scales: S (2 layers, d=128, 4 heads, ∼1.6M params), M (2L, d=256, 8H), L (4L, d=512, 8H, ∼25M); untied control U12 (12 distinct layers, T=2, depth-matched to S at T=12). NoPE throughout except the learned-PE arms.

Training. AdamW, lr 3×10^{-4} (2×10^{-4} for L), weight decay 0.01, grad clip 1.0, batch 256 (128 for L). Length curriculum with stages $n_{\max} \in \{4, 8, 16, 32\}$ (6k–12k steps each), promotion at per-token accuracy ≥ 0.98; lengths sampled uniformly within a stage. Loop schedules: log (T=$\lceil \log_2 n \rceil$+3, jitter ±1), prop (T=$\lceil n \cdot m \rceil$ for multiplier $m \in \{0.25, 0.5, 1, 2\}$), fixed ($T$=12), streaming ($T$=$n$+2). Horizon curriculum (used for chained A_5/S_5): the per-position loss is truncated to the first h positions, h ramping $4 \to n_{\max}$ within each stage. Operator-first S_5 curriculum: stages 2:20k, 4:10k, 8:10k, 16:10k, 32:12k at scale M. Annealing arms warm-start from a converged checkpoint of the source schedule and re-train under the target schedule with stages 16:6k, 32:8k.

Population sizes. 64 pilot models; 56-run Stage-2 grid ($S_5/S_4/\mathbb{Z}_{60}\times$ scales $\times$ 4 seeds + streaming arms); expansion to 16 seeds in the three race cells; controls (A_5, matched-order; untied; learned-PE; T-multiplier; streaming±deep supervision; operator-first; annealing); 8 easy-to-hard runs. ∼170 trained models in total, on 8× RTX 3090; individual runs take minutes (S) to ∼1.5h (L, long stages).

Easy-to-hard protocol. Official 32-bit training split (10,000 examples) with random prefix crops $n \in [4, 32]$ for the length curriculum; evaluation on the official 512-bit test split and on generator-matched intermediate lengths; NoPE, scale S, 12k steps.

Metrics. Exact match = all supervised positions correct; per-token accuracy over supervised positions. Generalization outcome in §8: mean per-token accuracy at $n \in \{32, 64\}$ under the schedule-matched loop budget, recomputed uniformly from checkpoints. Statistics: tie-aware Spearman; partial correlations by residual rank regression on adjacent-loop cosine.