Scaling Exact Fourth-Order Cumulant Propagation to 1024 × 16: Zero-Sample K=4 via Factored n³-Memory Transport Cuts Error 11.8× Below Augmented K=3 and Reveals Why Memoryless Closures Stall

ARC White-Box Estimation Challenge 2026 · Submission #336537 (The Only Public Exact Full K=4 at 1024 × 16)

Yan Su (AIcrowd: Susupro) · hi@yansu.me · github.com/logxio/fourth-cumulant (MIT)

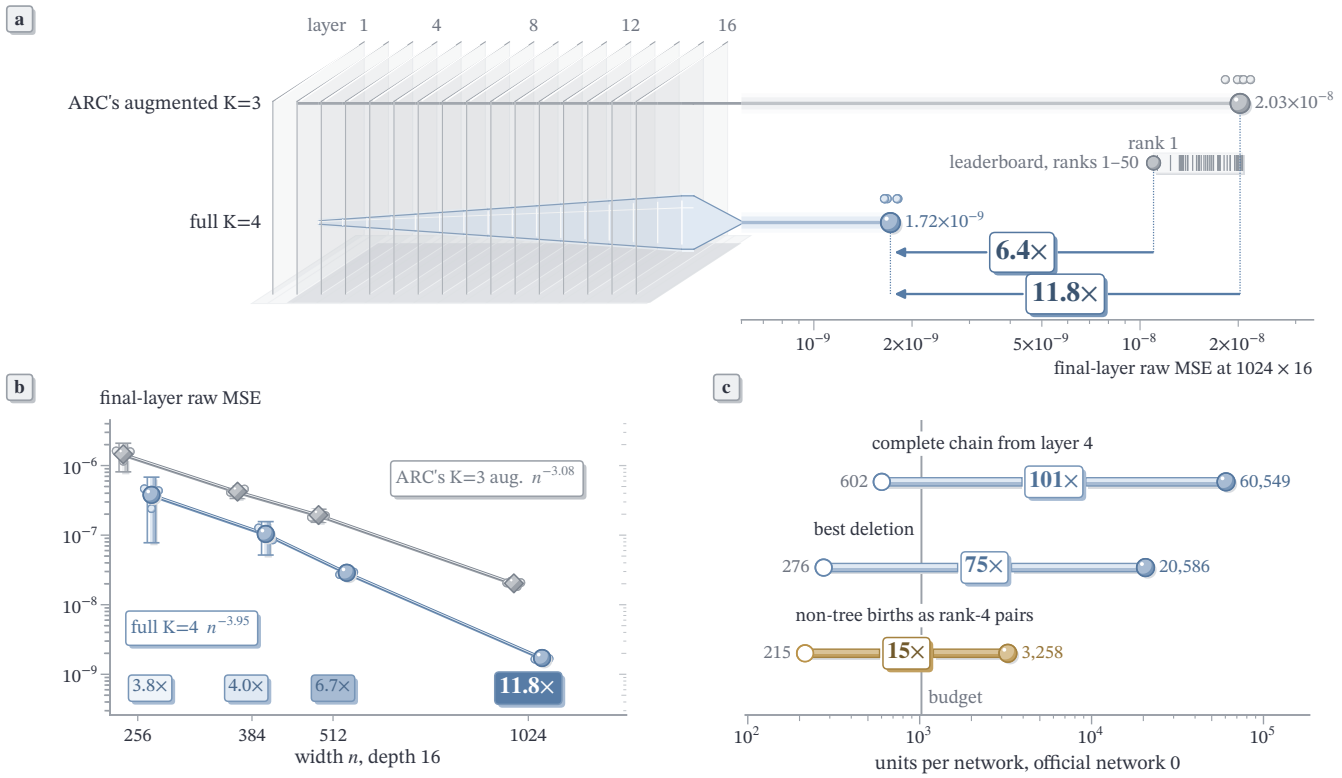


Figure 0: **Exact unaugmented K=4 at 1024 × 16 ends 11.8× below augmented K=3 on all five official networks and 6.4× below the lowest raw error on the leaderboard, its error falling as $n^{-3.95}$.** (a) The 16 layers of the five official networks as the glass plates of Figure 3; the K=4 state's width counts its live source groups, and each beam ends at its final-layer raw MSE on one log ruler (large disc the mean, small discs the networks). Ticks: each entry ranked 1–50 on the official leaderboard (18 October), on its 50 networks. (b) Small discs networks, large marks means, tubes 95% t intervals. (c) Official network 0 from layer 4: counted units (disc), units at which each configuration ties our scored chain (ring).

1. **Zero-sample K=4 at the challenge shape: 11.8× below augmented K=3.** ARC's complete K=4 cumulant recursion, run here for the first time in public at 1024 × 16 (Table 0, Figure 0a), where a Phase 1 write-up measured K=4 at a 61.5 GB memory wall already at 256 × 16 (forum 18154), reaches a final-layer raw MSE of 1.72e-9 on the five official networks, lower than ARC's augmented K=3 on every one, with no samples, fitted constants or tuning. The gap opens with width (Figure 0b), from 3.8× at 256 to 11.8× at 1024, at exponents $n^{-3.95}$ against $n^{-3.08}$: the 4 and 3 of ARC's conjectured bound $c_K(L/n)^K$ at fixed depth.
2. **6.4× below rank 1 of the leaderboard.** Full K=4's 1.72e-9 on the five official networks sits 6.4× below the lowest raw error on the official leaderboard, where every entry ranked 1–50 lies between 1.1e-8 and 2.1e-8 on the board's 50 networks (Figure 0a), the decade of augmented K=3's 2.03e-8.
3. **The bottleneck to the budget sits in two births.** From layer 4 the complete chain counts 60,549 units in this representation, 101× the 602 at which it ties our scored K=3 chain, 96% of them in C·κ3 and non-tree births; storing the non-tree births as rank-4 matrix pairs brings the best deletion from 75× to 15× the units at which it ties (Figure 0c).

For the judges' record, one line per criterion of rules §6.

(i) *Novelty and performance impact.* The first public run of ARC's complete K=4 recursion at the challenge shape, 1024 × 16, reaches a final-layer MSE of 1.72e-9, 11.8× below augmented K=3 on all five official networks (Table 1, Figure 0); Propositions 1 and 2 state why it fits on one GPU and why memoryless closures stall.

(ii) *Clarity and accuracy of the write-up.* All 1,935 numbers of the report sit in `data/numbers.csv` with their source files, and every K=4 code path matches ARC's reference to 1e-15 at n = 40 (Table A4).

(iii) *Ease of determining the actual performance impact from code and write-up.* Every scored step is its own grader submission, so our steps' factor of 0.830 [0.820, 0.839] on mean adjusted error, lower on 50/50 leaderboard networks, is read off eight receipts, and `verify.py` checks the estimator hash and rescores the saved predictions (Table 4, Section 8).

(iv) *The write-up requirement.* Approach (Sections 2–5), development (Section 9) and measured impact (Tables 1, 4, 5) correspond to exactly one graded submission, #336537, whose adjusted error of 3.145e-9 is the lowest among the eight public Phase 2 write-ups posted by 20 October and 5.9% below the next, hyojun_kwon's #335413; #335301 ranks first among the eight write-ups' chains without a learned corrector or mean-correction table, 1.3% below his #334316 (Table 4b).

Summary

Three regimes (Figure M). *Regime 1, exact recursion at toy width.* ARC's paper (Wu et al., 2026) gives the K=3 and K=4 cumulant recursions, basic, augmented and factorized, with experiments at widths 4–256 and 2–12 hidden layers; every K=4 number in this report evaluates those recursions and their term tables, and ARC's reference code checks each of our K=4 code paths to 1e-15. Its dense factor storage would hold about 256 GB per network at 1024 × 16. *Regime 2, budget-pruned K=3 chains with fitted or learned closures.* The graded submissions in all public Phase 2 write-ups run inside the budget, ours included, and none carries full K=4; among the eight posted by 20 October, ours leads (Section 4). *Regime 3, full-scale factored K=4 with readout-guided pruning.* This report runs ARC's exact unaugmented K=4 at the challenge shape: the first empirical confirmation at the challenge scale, 1024 × 16, of the fourth- and third-order width exponents of the $c_K(L/n)^K$ bound conjectured by Wu et al. (2026, Section 5), $n^{-3.95}$ for K=4 (95% interval 3.68–4.08) and $n^{-3.08}$ for augmented K=3 (2.95–3.17) over widths 256, 384, 512 and 1024, intervals that hold 4 and 3 (finding 1). Measuring every projection step through the whole 16-layer chain, as ARC's deduction–projection account of this recursion asks, it meters the first compressed K=4 chains end to end: rank-32 matrix pairs hold **6.95e-9 at 525,045 units, 30.6% fewer than dense** (Section 5).

Two terms. *Zero-Sample Mechanistic Inference* is an estimate computed deterministically from ARC's term tables and the network's weights, with no samples, no fitted constants and no tuning on any network. Every K=4 number in Sections 1–3 is of this kind; a Monte Carlo mean needs 4.89×10^7 forward passes, 746.5 budgets of sampling compute, to match full K=4's 1.72e-9. *Downstream Readout Observability* is the value of a piece of carried state measured by what later layers read from it into the final estimate, not by how exact it is at the layer where it is born. We measure it end to end, as the final-layer MSE without the state over the MSE with it, and Proposition 2 shows that a local norm cannot stand in for it. They are that account's two halves: exact deduction steps, and our test of each projection step.

Why memoryless closures stall. A memoryless closure rebuilds the fourth cumulant from the current layer's state and carries no fourth-order source forward. Handed the K=4 chain's own exact covariance and $\kappa4$ at every layer, the best that any such closure could supply, augmented K=3 reaches 0.29 of its error; the K=4 chain, which carries its sources for life and feeds them back into $\kappa3$, reaches 0.08. The exact (3,1) slice inside our K=3 chain recovers at most 10% of the K=4 chain's log gain, and ending every source's life after two layers costs 109× (Section 3, Table 3c). The gain grows with n/L: at n/L = 8, where 504aldo's findings log concludes that the series is best truncated at K=3, unaugmented K=4 beats augmented K=3 on all six networks frozen before running, and the loss F29 measured belongs to the augmented variant (Section 1).

Toward the budget. The two birth families of finding 3 hold both the units and the gain. Metered by flopscope on official network 0, the chain from layer 4 holds 6.37e-9 at 756,620 units with dense (2,1,1) kernels; the rank-32 pairs above cost 9.1% more raw and cut its distance to the 602-unit tie from 1,257× to 872×. In both, 87–88% of the units are reads of the slice and two-leg groups, each an n^4 product because $\kappa3$ is stored dense: the next cut is $\kappa3$'s representation (Section 5.5).

Inside the budget. The same rule built the scored entry. Our deletion steps on the scored K=3 chain, chain B, multiply the grader's mean adjusted error by **0.830** [0.820, 0.839], lower on 50/50 leaderboard networks, and #336537 reaches **3.145e-9**: of the eight public Phase 2 write-ups posted by 20 October, each bound to one graded submission, it has the lowest adjusted error, **5.9% below the next**, hyojun_kwon's #335413, at 0.762× its raw. #335413 is hyojun_kwon's #334316 plus a learned terminal corrector on a byte-identical state, the same structure as #336537 on our #335301, and chain B alone leads its own class: **#335301 ranks first among the chains of all eight public Phase 2 write-ups that run without a learned terminal corrector or a layerwise mean-correction table**, at adj 3.494e-9, 1.3% below hyojun_kwon's #334316 in adj and 18.9% in raw (Table 4b).

Table 0. Public results across the Summary's three regimes: ARC's paper, exact recursions at toy width; the eight public Phase 2 write-ups posted by 20 October, budget-pruned K=3 chains with fitted or learned closures, ordered by the adjusted error of the graded submission each is bound to (rules §6; pages of 19 October, Table 4b), #336537 first, #335301 first without corrector or mean-correction table; this report's full-scale K=4; and a Phase 1 replication, with how this report relates to each.

work	what it measured	this report
Wu et al., ARC (2026), reference code	K=3 and K=4 recursions, basic, augmented and factorized; experiments at width 4–256 and 2–12 hidden layers	the same unaugmented K=4 at width 1024 and depth 16, made storable (Section 2)
	a conjectured bound $c_K(L/n)^K$ (their Section 5)	exponents 4 and 3 first confirmed at 1024 × 16 (Section 1)
this report, forum 18254	bound submission #336537: adj 3.145e-9, raw 1.853e-8, C/B 0.170	first of the eight write-ups, 5.9% below the next (Table 4b)
	#335301, the same chain without its learned corrector: adj 3.494e-9	first of the eight write-ups' chains without a learned corrector or mean-correction table
hyojun_kwon, forum 18255	bound submission #335413: adj 3.343e-9, raw 2.431e-8	#336537: 0.941× the adj, 0.762× the raw
	#335413 is the frozen #334316 (adj 3.541e-9) plus a learned corrector, state unchanged	without the correctors, #335301 on #334316: 0.987× the adj, 0.811× the raw
	low-rank fourth-order cores and oracle ceilings of pair fourth-order blocks: measured negatives	the complete K=4 chain, carried for life: 11.8× at 1024 × 16 (Section 1)
Artyom_Mazur, forum 18249	bound submission #333534: adj 3.736e-9, a K=3 chain with a scale-mixture κ4 closure and a learned corrector	#336537 lower in adj, raw and C/B: 0.842×, 0.912×, 0.92×
	a per-neuron correction learns at most 4% of a K=3 chain's error	our corrector, its state carried through W: 0.873 of chain B's raw error on the mini networks, 0.887 graded (Section 4.2)
	at 256 × 16, three layers of κ4 blocks plus one scale-mixture block keep 89% of K=4's gain	at 1024 × 16 a three-layer lifetime is 67× worse than full K=4 on network 0 (Section 3)
	a young-κ4 tree reader gives ×0.70 raw on ten public networks at about the cost it saves	
Nurlanggs, forum 18252	bound submission #336225: adj 3.813e-9, V29 with an all-distinct κ4 edge term and a fitted (3,1) law	#336537 lower in adj, raw and C/B: 0.825×, 0.883×, 0.93×
	oracle injections and memory cuts: the all-distinct third cumulant needs rank about 3n/4 and the full depth	the same limit from the K=4 side, where κ3's representation is the next cut (Section 5.5)
il.miscusi, forum 18248	bound submission #336317: adj 5.176e-9 at C/B 0.301, with the lowest raw of the eight, 1.721e-8	#336537: 0.608× the adj at 0.56× the compute
	carrying κ4 with memory makes a K=3 engine 2.25× worse until a scale mode is removed	18249 derives and measures the scale mode (Proposition 4); 18248 removes it in the estimator
	that mode inflates the carried cumulants but leaves every mean unchanged; with it removed, 0.380× (graded, 50/50)	
kai_leal_gonzalez, forum 18246, code	bound submission #334821: adj 5.264e-9, V29 with a two-term (3,1) closure	#336537: 0.597× the adj, 0.894× the raw
	inside V29 the exact (3,1) slice of κ4 lowers final MSE 18.1% on 23 networks	the whole K=4 chain and its births: exact local patches recover part of the gain, the chain the rest (Section 3)
	a two-term closure recovers about a sixth (−3.13% on 77 pre-registered networks, graded 5.42e-9 → 5.26e-9)	
	tree births of the slice give +0.7% to −2.7% at +31% FLOPs	
504aldo, forum 18218, V29	bound submission #330093, V29 itself: adj 5.418e-9	#336537: 0.580× the adj, 0.869× the raw
	F29: augmented K=4 17–20× worse than augmented K=3 at (64, 8) and (96, 12), out of memory at (128, 16)	F29's two cells rerun with unaugmented K=4 (Figure 1a, Table 2b)
	F29 concludes the series is best truncated at K=3 at n/L = 8	unaugmented K=4 wins on all six networks
	F42: augmented K=4 at 1024 × 16 priced at 13,348 budgets	V29 is chain B's starting point (Section 4)
DD_team, forum 18250	bound submission #335813: adj 2.733e-8, a V29-route closure with exact cost cuts, at C/B 0.853	#336537: 0.115× the adj
James Henry, Phase 1	a third-cumulant carrier at cosine 0.83 to the truth: final estimate 4.5× worse than its probe (256 × 32)	the same split between local fidelity and downstream value, at K=4 and 1024 × 16

Origin and fit of every piece. *ARC's:* the K=3 and K=4 algorithms and their term tables; every K=4 number in Sections 1–3 and Figures 1–3 evaluates them deterministically, with no samples, no fitted constants and no tuning on any network. *Ours:* the storage and the n³ forms that run them at 1024 (Section 2), every ablation, patch and late start, and the budget ledger. *Inherited into chain B:* 504aldo's V29 (MIT), with 16 λ values of its κ4 regeneration law (least squares on 8 public networks) and 15 reference ratios of its adaptive rule (its 16 × 13 ridge mean-correction table stays switched off); koushik_rudra's V34 dead-unit pruning, which restricts the third-cumulant leg products to the rows of active neurons, and acco-cyber's V37 cost blocks and final-layer row band (both MIT, one team); Puffi's SSC response closure for the κ4 regeneration law, whose 9 coefficients we refit end to end on 12 public networks. *Ours, scored:* active-set trimming, which extends V34's cut to the columns of the (2,1) slice, young-atom pruning and the cost work of Table 4's steps 3, 4, 6 and 7, with ranks, age gates and thresholds chosen on public networks. *Fitted, scored:* the corrector of #336537 (Section 4.2), 54,561 parameters

trained on the public full split and 2,048 networks of Keenan Pepper's public corpus. Nothing was fitted on the grader's networks, which are disjoint from the public sets.

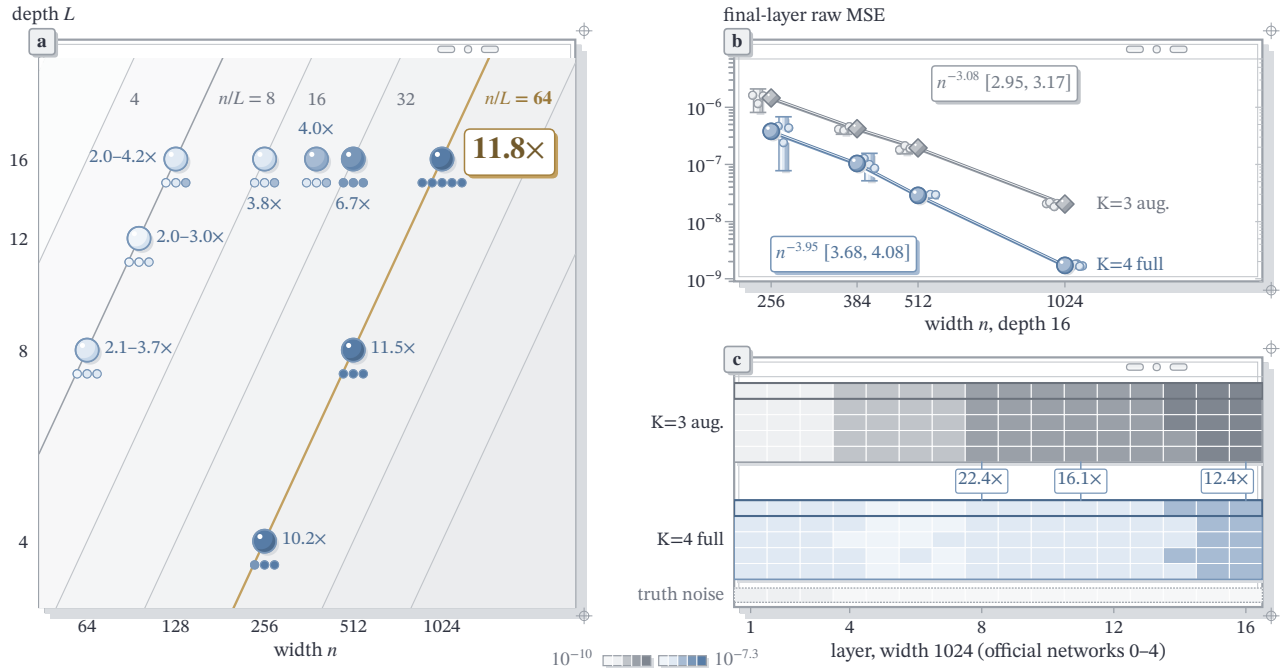


Figure 1: **Exact unaugmented K=4 scales as $n^{-3.95}$ against augmented K=3's $n^{-3.08}$ and cuts final-layer MSE 11.8× below it at 1024 × 16 on all five official networks; the gain follows n/L, 10.2–11.8× for all three shapes at n/L = 64 and 2.0–4.2× on the nine networks at n/L = 8, the ratio at which F29 found augmented K=4 behind.** (a) Every measured shape at its width and depth; the diagonals are lines of constant n/L, the bands between them deeper as n/L grows. A station's shade and label give K=3 aug. ÷ K=4 full as a ratio of mean errors (at n/L = 8 the range over its three networks), the dots beneath it the ratio on each network. (b) Raw final-layer MSE against width at depth 16: small discs are single networks (three per width, the five official networks at 1024), large marks their means, tubes 95% t intervals, chips the fitted exponents with 95% network-resample intervals. (c) Per-layer MSE of the five official networks at width 1024, one cell per network and layer, network 0 on top and outlined; chips are the noise-removed ratio at layers 8, 11 and 16, the bottom strip the truth's sampling noise. Tables 2 and 2b (Appendix A.8) list every value.

1. The Fourth-Order Scaling Law at 1024 × 16

At depth 16, over widths 256, 384, 512 and 1024, the final-layer MSE of ARC's exact unaugmented K=4 falls as $n^{-3.95}$ (95% interval 3.68–4.08) and that of augmented K=3 as $n^{-3.08}$ (2.95–3.17), and at 1024 × 16 the two stand 11.8× apart on all five official networks (Table 1, Figure 1). This is the first empirical confirmation at the challenge scale of the fourth- and third-order width exponents of the $c_K(L/n)^K$ bound conjectured by Wu et al. (2026, Section 5), whose own width scan (their Section 6.3) runs to width 256 at four hidden layers. It is the first reading of Regime 3: the recursion of Regime 1, run at the shape where Regime 2's graded chains compete, and measured, as the deduction–projection account behind it asks of every projection step, through the whole chain (Sections 3 and 5.5).

Metric. "raw" is the final-layer MSE against ARC's 10^9-sample reference, "C/B" is metered flopscope FLOPs over the 2^{41} budget, and the score ("adj") is raw × max(0.1, C/B), averaged over networks. One unit is one n×n×n matmul, $2n^3 = 2^{31}$ FLOPs, so B = 1024 units.

Networks and truth. Official public `mini` networks of `aicrowd/arc-whestbench-public-2026@v2-phase2`, rebuilt from their seeds; the closed-form first layer matches ARC's reference to 7e-10. A numbered network (network 0, networks 0–49) is a mini network unless marked as one of the 1,000 of the public full split. The grader scores 100 other networks, 50 on the leaderboard and 50 sealed; "leaderboard networks" are those 50. Ground truth is ARC's `all_layer_means`. Its noise at layer l is the neuron-averaged activation variance over 10^9: we estimate the variance from 2^{15} inputs per network, and at the output it reproduces the official `avg_variance` to three digits on all five networks. The noise falls from 6.8e-10 at layer 1 to 1.7e-10 at layer 8 and 8.4e-11 at the output. The same variance prices sampling: at 0.084 per output neuron, a Monte Carlo mean reaches full K=4's 1.72e-9 after 4.89×10^7 forward passes of $16 \times 2n^2$ FLOPs, 746.5 budgets of sampling

compute.

Estimators. "K=4 full" is ARC's `factor_k4` (augment = False) with every source type, in our storage (Section 2.3); its output equals the basic algorithm's (the paper's §4.3). "Exact" means this truncated recursion evaluated without further approximation. "K=3 augmented" is ARC's factorized augmented K=3, the strongest K=3 in the reference, run in the reference itself for every number in Tables 1–3 and Figure 1. For the per-layer matrices of Figure 1c the same reference prints every layer of networks 0–4 on one machine; its final layer reproduces Table 1 to within 0.04% on each network, the arithmetic difference between two GPUs. "K=3 simple" is ARC's reference `k3s`. "Chain B" is our scored K=3 chain (Section 4).

Table 1. Raw final-layer MSE at width 1024, official public networks.

network	K=3 simple	K=3 augmented	K=4 cheap n^3 subset	K=4 full	K=3 aug ÷ K=4
0	3.70e-8	2.06e-8	1.01e-8	**1.65e-9**	12.5×
1	4.46e-8	1.99e-8	1.16e-8	**1.81e-9**	11.0×
2	4.59e-8	2.08e-8	1.13e-8	**1.68e-9**	12.4×
3		2.18e-8	1.17e-8	**1.80e-9**	12.1×
4		1.83e-8	1.10e-8	**1.65e-9**	11.1×
mean	4.25e-8 (0–2)	2.03e-8	1.11e-8	**1.72e-9**	**11.8×**

Full K=4 runs 206–299 s per network on one 96 GB GPU, peaks at 60.4 GB of GPU memory and keeps 120 GB of births in pinned host memory (240 GB of host memory in all). The five networks are the independent samples; rerunning networks 0–3 on a second machine reproduces every value to three digits, a check of the computation. Run from the input in the compact representation of Section 5, its portable subset counts at least 77,930 units per network against the 1,024-unit budget (Table 7a).

Width. The exponents are least-squares fits of the log mean error on log n over the four widths, their 95% intervals from resampling the networks within each width 20,000 times; each interval holds the exponent 4 or 3 that the bound $c_K(L/n)^K$ gives at fixed depth. The measured exponent separation is 0.87 (0.74 to 0.96) at finite depth 16. The K=4 error spans 1.9× across the three networks at 256 and 1.10× at 512 and at 1024.

Depth. The per-layer error at layer l is the final error of the same network cut after l layers, the construction of the paper's depth experiments. Full K=4's error above the truth noise is 0.2 times the noise at layer 5, 1.3 times at layer 8 and 20 times at the output, so its measured error in Figure 1c first falls only because the noise falls. With the noise removed, over the nine layers 8–16 the K=4 error grows as $L^{2.9}$ (2.7 to 3.1 across networks 0–4) and augmented K=3 as $L^{2.1}$ (1.9 to 2.3): both slower than the L^4 and L^3 that the bound gives at fixed width, and 0.8 apart. The paper's Appendix D finds the basic algorithm's error close to L^K at width 256 for large L; full K=4, whose output equals the basic algorithm's, grows as $L^{2.9}$ at width 1024. The ratio of the two falls from 22.4× at layer 8 to 16.1× at 11 and 12.4× at 16 (11.5× to 13.1× across networks at 16).

Both axes together. If both errors scaled as their bounds, the ratio of the K=3 and K=4 errors would be $(c_3/c_4)\,n/L$. With the truth noise removed it grows as $(n/L)^p$ with p = 0.90 (95% interval 0.77–0.99) over the four widths and p = 0.75 (0.55–0.90) over layers 11–16, where K=4's error is at least 4.6 times the noise; fits from layer 8, 9 or 10 give 0.80, 0.74, 0.70 (20,000 network resamples, within each width and over networks 0–4). The two series share their n/L = 64 point, the five networks at 1024 × 16 (12.4×).

Same n/L, three shapes (Figure 1a; Table 2 in Appendix A.8). At n/L = 64, three new networks each at 256 × 4 and 512 × 8 give 10.2× (8.5–12.3) and 11.5× (10.8–12.1), and the five official ones at 1024 × 16 give 11.8× (11.3–12.4). At the same widths and depth 16 the ratios are 3.8× and 6.7×: cutting depth by the factor that keeps n/L at 64 raises them 2.7× and 1.7×, to 0.87 (90% interval 0.75–1.00) and 0.97 (0.91–1.04) of the 1024 × 16 ratio (20,000 network resamples). The truth's noise is 4–5% of the K=4 error at the two smaller shapes; a two-batch unbiased MSE gives 10.6× and 11.9×.

F29's cells with this report's K=4 (Figure 1a; Table 2b in Appendix A.8). 504aldo's F29 measured augmented K=4 17–20× worse than augmented K=3 at (64, 8) and (96, 12), could not run (128, 16), and concluded that at n/L = 8 the series is best truncated at K=3. We ran both cells on three new frozen networks each, with ARC's unaugmented `factor_k4` (this report's K=4), augmented K=3 and, where it fits in our 4 GB process limit, augmented K=4. Unaugmented K=4 beats augmented K=3 on all six networks, 2.0–3.7×, and on the three 128 × 16 networks of our earlier order scan, 2.0–4.2×. Augmented K=4 is 1.6–4.0× worse than augmented K=3 on the three (64, 8) networks, the direction F29 reports. At n/L = 8 the loss belongs to the augmented variant; the unaugmented recursion gains there already, and its gain grows to 11.8× at n/L = 64.

2. Breaking the 256 GB Memory Wall: Factored n³-Memory Leg-and-Kernel Transport

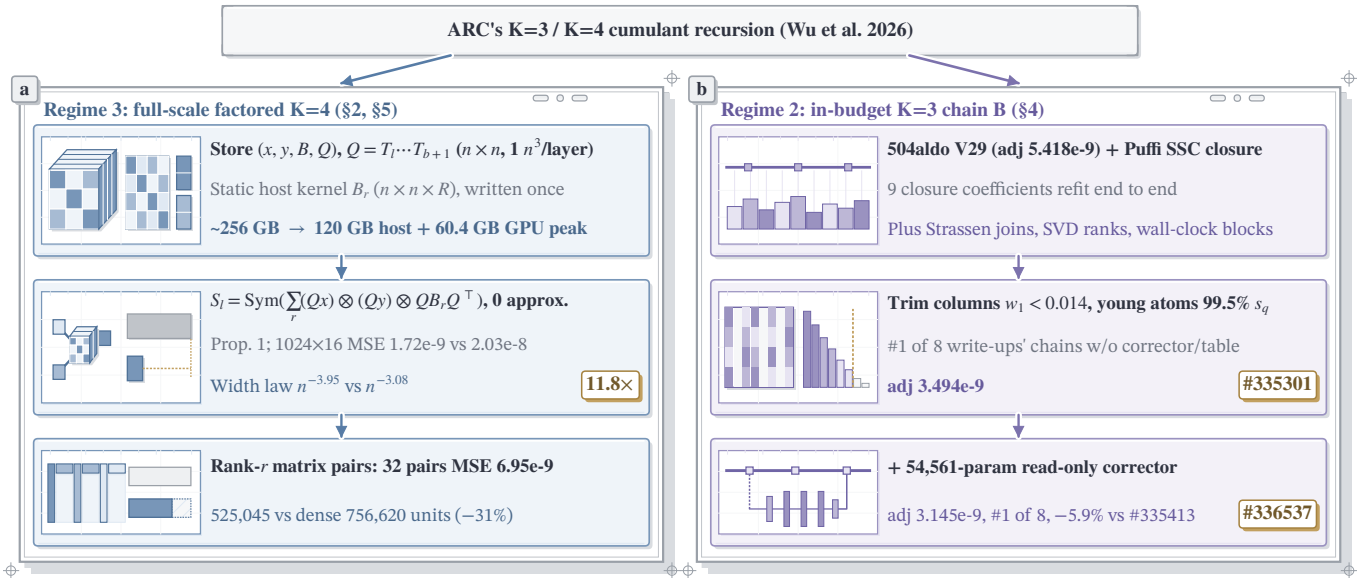


Figure M: **From ARC's recursion, two tracks: factored storage runs exact K=4 at 1024 × 16 on one GPU, 11.8× below augmented K=3; inside the budget, pruning K=3 chain B by what later layers read gives #336537, the lowest adjusted error of the eight public write-ups posted by 20 October, and #335301, first among their chains without a learned corrector or mean-correction table.** (a) Proposition 1 makes the storage exact; the 32 matrix pairs are metered end to end on network 0. (b) Table 4's steps from V29 to #335301; the corrector adds its carried error to the final estimate and leaves the chain unchanged.

This section is the left track (Figure M); Appendix A writes one layer out as pseudocode with every tensor shape.

2.1 Sources

The fourth cumulant of the pre-activations is a sum of sources. Each ReLU layer gives birth to two groups, which are then transported through every later layer by the weights and ReLU slopes.

- **Two-leg group.** Let P be the propagator leg (the identity at birth, multiplied by each later layer's weights and slopes), w_1, w_2, w_3 the per-neuron Hermite coefficients of the ReLU at the birth layer, C_{off} the off-diagonal covariance, and $A = P\,\mathrm{diag}(w_1)\,C_{\text{off}}$. Up to symmetrization over the four indices:
  - star: $\sum_a 4\,w_{3,a}\,A_{:a} \otimes P_{:a} \otimes A_{:a} \otimes A_{:a}$
  - path: $\sum_{a,b} M_{ab}\,A_{:a} \otimes P_{:a} \otimes P_{:b} \otimes A_{:b}$ with $M = \mathrm{diag}(w_2)\,C_{\text{off}}\,\mathrm{diag}(w_2)$
  - C·κ3: $\sum_s \sum_{a,c} Z^{(s)}_{ac}\,A_{:a} \otimes P_{:a} \otimes P^{(s)}_{:c} \otimes Y^{(s)}_{:c}$ with $Z^{(s)} = \mathrm{diag}(w_2)\,X^{(s)}$, where $(X^{(s)}, P^{(s)}, Y^{(s)})$ are the legs of a live κ3 source s.
- **Slice group.** The fourth-cumulant slices that the nonlinearity itself produces: (4,), (3,1), (2,2) and (2,1,1).

2.2 What the next layer reads

The ReLU update at the next layer reads four diagonal slices of κ4: (4,) κ_{iiii}, (3,1) κ_{iiic}, (2,2) κ_{iicc}, and (2,1,1) κ_{iicd}, which feeds the third cumulant. For a source stored as legs and a kernel, a slice that identifies indices on the same side of the kernel is a few n×n matmuls on Hadamard products of legs. A slice that identifies indices across the kernel (a **crossed pairing**) needs the kernel itself: 3 matmuls when the kernel is diagonal (star), 4r matmuls at rank r, n^4 at full rank.

2.3 Storage that fits one GPU

ARC's reference stores each source as $\mathrm{Sym}(\sum_r A_r \otimes B_r)$ with A_r a dense n×n×R factor, about 256 GB by layer 16 at n = 1024 by our count of its shapes. Our storage, Proposition 1, matches the reference to 1e-15 at n = 40 and our n = 128 reference run (1.604e-5) to the printed digits.

Proposition 1 (exact factored source storage). Let a source group born at layer b be $S_b = \mathrm{Sym}\big(\sum_{r=1}^{R} x_{:r} \otimes y_{:r} \otimes B_r\big)$, with birth legs $x, y \in \mathbb{R}^{n \times R}$ and birth kernel slices $B_r \in \mathbb{R}^{n \times n}$, and let every later layer t move each live source by one matrix

T_t on all four indices (the linear step W_t and the carry $\mathrm{diag}(w_1)$, steps 1 and 5 of Algorithm A1). Then at every layer $l > b$, exactly,

$$S_l = \mathrm{Sym}\Big(\sum_{r=1}^{R} (Qx)_{:r} \otimes (Qy)_{:r} \otimes Q\,B_r\,Q^\top\Big), \qquad Q = T_l\,T_{l-1}\cdots T_{b+1}.$$

Storing (x, y, B, Q) in place of the transported factors therefore adds no approximation at any layer. Per layer a group transports Q, one n^3 product, where ARC's form rewrites $2n^2R$ numbers, and the n×n×R birth kernel is written once, to host memory: at $n = R = 1024$, about 256 GB of transported factors become 120 GB of static host kernels and a 60.4 GB GPU peak (Table 1).

Proof sketch. $T^{\otimes 4}$ is linear, sends a product of four vectors to the product of their images, and commutes with Sym because one matrix acts on every index; so one layer maps $x_{:r} \otimes y_{:r} \otimes B_r$ to $Tx_{:r} \otimes Ty_{:r} \otimes TB_rT^\top$, and induction over the layers gives the product Q. Appendix A.9 writes the proof out. Exact factored storage and transport remove the 256 GB dense transport memory wall with no representation approximation, leaving the n^4 reads of $QB_rQ^\top$ as the target of Section 5's low-rank bilinear compression.

2.4 The n^3 leg-and-kernel form

Every two-leg source can be written as n×n legs plus an n×n or diagonal kernel; the (4,), (3,1), (2,2) slices and the (2,1,1) same-side CP (canonical polyadic: a sum of rank-one three-way terms) then come from matmuls, without forming an n^3 or n^4 tensor. These slices agree with the n^4 chain to within 1e-18 (absolute, float64) at $n = 128$ and 256. The form covers same-side reads and the star crossing; the C·κ3 crossing above rank 1 and the non-tree feedback reads stay outside it (Section 5).

2.5 Old sources in a shared basis, batched

We carried the old-source idea of 504aldo's V22 from K=3 to K=4. Legs of sources older than m_0 transports are written $A_s = QF_s$ with one transported basis Q (n×k) and static factors F_s, so the per-layer transport depends on k and not on the number of old sources. All sources of one type form one batched family: legs concatenated along columns, one contraction per layer, an incremental Gram for the basis update. In a flopscope micro-benchmark at width 1024 over 16 layers (entry age 2, $k = 128$, κ3 legs n columns wide, young legs static), batching takes the K=4 source machinery from 647.7 units, 80.4 s wall time and 1.19 s residual time to **566.4 units, 73.0 s and 0.33 s**, inside the 0.4 s residual cap.

2.6 The (2,1,1) → K3 feedback

Each layer the K=4 state pushes a third-cumulant contribution T into the K=3 chain: 20 graph terms in 8 classes of ARC's term table. Kept as CP columns, the newborn part reaches 790,144 columns at layer 15. Two changes make it tractable.

- **Age gate with a three-axis core.** Contributions older than m_{0f} layers are projected on all three axes into one transported basis Q_3 (n×k_3) and summed in a k_3^3 core: 0.42 units per layer at $k_3 = 64$, 3.0 at $k_3 = 128$.
- **Young contributions from their graph-term factors.** A slice group born at layer L reaches layer L+1 only through two n×n quantities, A_{21}, the (2,1) slice of $F_1 = ((Q \circ Q) \otimes Q \otimes Q) \cdot T$, and $dB_{ir} = 6\sum_{a,b} Q_{ia}Q_{ib}T_{rab}$, where Q is the layer's transport:

  $$(4,) = \mathrm{rowsum}(Q \circ Q \circ dB), \qquad (3,1) = 3A_{21} + \tfrac{1}{2}(Q \circ dB)\,Q^\top, \qquad (2,2) = \mathrm{sym}\big((Q \circ Q)\,dB^\top\big)/6,$$

  and (2,1,1) is F_1 plus a dB term, which goes straight into the core. Grouping the tree terms by shared edges (two groups centred on the repeated axis, three on the single axis, two diagonal corrections) costs 22 n×n matmuls at the first feedback layer and 26 at each later one, 386 units over the chain. Over all 16 layers at $n = 128$ this chain matches the reference chain to 4.2e-15 (float64). The non-tree terms of the same table have no such factorization yet; Section 5 shows that they carry the gain.

3. Downstream Readout Observability: Why Memoryless Closures Stall

Definitions. The *downstream readout observability* of a block of carried state at layer l is what the later layers read out of it into the final estimate. We measure it end to end, as the final-layer MSE with the block removed over the MSE with it, every later layer run as written. A *memoryless closure* rebuilds the fourth cumulant at each layer from that layer's state and carries no fourth-order source forward; "third cumulant plus a memoryless fourth" lands at raw 2.13e-8 at 1024×16 in 504aldo's measurements (forum 18218), in the band of augmented K=3.

Proposition 2 (readout observability). Let $F(s)$ be the final-layer estimate as a function of a block $s \in \mathbb{R}^d$ of state carried at layer l, every later layer fixed and F twice continuously differentiable near s, and let $g = \nabla_s F(s)$ be its downstream readout vector. For a truncation $\tilde{s} = s - \delta$:
(a) $F(s) - F(\tilde{s}) = \langle g, \delta \rangle + O(\|\delta\|^2)$, and $|\langle g, \delta \rangle| \le \|g\| \, \|\delta\|$, with equality when δ is parallel to g;
(b) the relative readout loss $|\langle g, \delta \rangle| / |\langle g, s \rangle|$ has no bound in terms of the relative norm loss $\|\delta\|/\|s\|$: for every $\varepsilon \in (0, 1)$ there are s and g for which the norm-optimal rank-one truncation keeps all but a fraction ε of the norm and loses the whole readout, while another rank-one projection loses almost all of the norm and none of the readout.

Proof sketch. (a) is Taylor's theorem and the Cauchy–Schwarz inequality. For (b) take the state $s = \mathrm{diag}(1, \varepsilon)$ and the readout $g = e_2 e_2^\top$: the norm-optimal rank-one truncation (Eckart–Young) keeps $\mathrm{diag}(1, 0)$, so $\delta = \mathrm{diag}(0, \varepsilon)$, $\|\delta\|/\|s\| < \varepsilon$ and $\langle g, \delta \rangle = \langle g, s \rangle = \varepsilon$; the projection onto $e_2 e_2^\top$ keeps $\mathrm{diag}(0, \varepsilon)$ and loses none of $\langle g, s \rangle$. Appendix A.9 writes it out.

The proposition is why local exactness and local energy mis-price carried state. A patch that is exact on one block leaves the truncation in every other block that later layers read, and a basis chosen for energy keeps the directions of largest norm, whatever g weighs. The measurements below are those two failures at the challenge shape, and the readings that settle them are collected in Table 3c.

Section 1 measured how much full K=4 gains. This section finds the state that carries the gain in three ways: by removing one birth type at a time (Table 3, Figure 2a), by cutting how long sources live (Figure 2b), and by handing a K=3 chain the K=4 chain's own exact quantities (Figure 3a).

Table 3. Removing one birth type from full K=4: final MSE ÷ full K=4. Width 1024: official networks 0–1, and networks 5, 6 and 7 frozen before running; width 512: seed 41; widths 192 and 128: three networks each, frozen before running (192 new; 128 the order-scan networks). Frozen columns give mean and range, at 1024 networks 5, 6 and 7 in turn. Monte Carlo truth below 1024.

removed	1024	1024, frozen	512	192 × 16, frozen	128 × 16, frozen
nothing (full K=4), raw MSE	1.73e-9	1.42e-9, 1.61e-9, 1.43e-9	2.71e-8	1.26–2.13e-6	1.22–3.13e-5
crossed pairings of the (2,2) and (2,1,1) slice births	1.04	1.12, 1.05, 1.05	1.04	1.09 (1.06–1.13)	1.09 (1.04–1.15)
path births	1.46	1.43, 1.27, 1.51	1.26	0.98 (0.90–1.13)	0.92 (0.82–0.97)
path births and slice crossed pairings ("portable set")	1.68				
C·κ3 births	**5.39**	**5.17, 4.67, 5.18**	**2.86**	**2.41** (1.55–3.57)	0.99 (0.61–1.28)
path and C·κ3 births	5.83		3.10		
everything except star and same-side slices ("cheap n^3 subset")	6.28		3.65		
the whole fourth cumulant (augmented K=3)	11.7		6.64	3.37 (2.56–4.81)	3.12 (2.00–4.17)

C·κ3 births take over as width grows. Removing C·κ3 births multiplies final-layer MSE by 0.99× at 128 × 16, 2.41× at 192 × 16, 2.86× at 512 and 5.39× at 1024, where it costs more than every other birth type together, and all three frozen 1024 networks (5, 6 and 7) preserve the full birth-removal ordering C·κ3 > path > slice crossings (Table 3). At 128 and 192, where the whole fourth cumulant buys 2.0–4.8× against 11.7× at 1024, removing path births lowers the error on five of six networks (0.82–0.97×, one at 1.13×), and on one 128 network removing C·κ3 births does too (0.61×); path births pay at 512 (1.26×) and 1024 (1.46×), and crossed pairings inside slice births cost 4–15% at every width. Each C·κ3 birth reads through the legs of every live κ3 source, so its read cost grows with the third cumulant (Section 5.3).

Why removing path births helps at small width. Below 1024 the K=4 recursion undercounts star births. On twelve frozen networks, three each at 64 × 8, 96 × 12, 128 × 16 and 192 × 16, we fit the true final-layer means as full K=4 plus reweighted net contributions of path, star and C·κ3 births (each the full prediction minus the prediction without that birth type; weight 1 is the recursion as written). Star births get weight 1.09–2.04 (median 1.43), above 1 on all twelve. Scaling a network's star contribution by the mean weight fitted on the other two networks of its shape lowers its final MSE on all twelve, to 0.44–0.97 of full K=4 (geometric mean 0.75). Path births point away from the missing star content: cos(path, star) < 0 exactly where removing path births helps on 10 of 12 networks, all nine at widths 64–128. At 1024 the undercount closes: on frozen networks 5, 6 and 7, fitted star weights are 1.03–1.07, leave-one-out star scaling gives 0.994–0.998 of full K=4, path and star births are nearly orthogonal with cos(path, star) from −0.09 to −0.03, and removing star births costs 1.79–2.05× (in-sample reweighting of all three types reaches 0.91–0.95 of full K=4).

The gain needs the whole chain (Figure 3a, networks 0–3). Feeding the K=3 augmented chain the K=4 chain's covariance at every layer gives 0.64× of its error, its κ4 alone 0.64×, both 0.29×, and κ4 only on layers 9–15 along the 64 leading directions of the map to the output 0.81×. The K=4 chain itself reaches 0.08×: exact covariance and κ4 at every layer leave 3.6× of the gain to the rest of the K=4 state. The same holds one level down. Inside chain B, without its corrector, the exact (3,1)

slice of κ4 from public 10^8-sample pair statistics gives 0.873 of its raw error over eight mini networks (7 of 8 lower), and in V29 the same slice gives −18.1% (Table 0); against full K=4 on the four of them where it ran, that is at most 10% of the K=4 chain's log gain (Table 10).

Sources must live long (Figure 2b, network 0). Dropping every K=4 source after 2, 3, 4 or 6 layers is 109×, 67×, 41× and 12× worse than keeping all 16, already worse than K=3 at 6 layers; slice births after a late start follow the same law (Section 5.2). Old sources compress where young ones cannot (Figure 2b, below the axis). Entering a shared basis of rank 128 after two transports costs 1.29× (3.90e-9 against 3.03e-9 exact), after one 1.66×, and at birth, at rank 112, 3.72×; at rank 256 after two transports the loss is gone (3.04e-9). Feedback follows the same pattern: newborn T projected into the core gives 5.10–5.57e-9 over $k_3 = 64$–256, and one or two exact layers before a rank-128 core give 4.17e-9 and 4.02e-9 against 3.90e-9 uncompressed.

Which reads need exactness. All four output channels are needed (removing one multiplies the error by 1,148 to 714,978), the star crossing pays only exact, the C·κ3 crossed kernel is close to full rank, and the non-tree term $C_{jk}\,\kappa_{3,ijk}$, which the factorized K=3 chains of the V29 lineage leave out, is worth 1.56× in the portable set (Section 6, rows 3, 4, 8 and 9; Appendix A.4 lists every reading).

Table 3c. Downstream readout observability, five readings: what removing, cutting or patching state does to the final-layer error, each with its key in `data/numbers.csv`.

reading	value	`numbers.csv` key	where
C·κ3 births removed from full K=4	5.39× on official networks 0–1; 4.67–5.18× on three frozen networks	`abl01_no_k3c ÷ abl01_full`; `fz1024_no_k3c_over_full_min, _max`	Table 3
every K=4 source ends after two layers	109× (network 0)	`lifetime2_net0 ÷ portable_net0`	Section 6, row 1
augmented K=3 handed the K=4 chain's exact C and κ4	0.29 of its error; the K=4 chain 0.08	`oracle_C_k4, oracle_full_k4_chain`	Figure 3a
exact (3,1) slice inside chain B	at most 10.0% of the K=4 log gain	`red_k31_share_log_net0`	Table 10
inside the budget: prune what later reads barely use	×0.830 mean adjusted error, lower on 50/50	`lad50_own_steps_adj_ratio_of_means`	Table 4

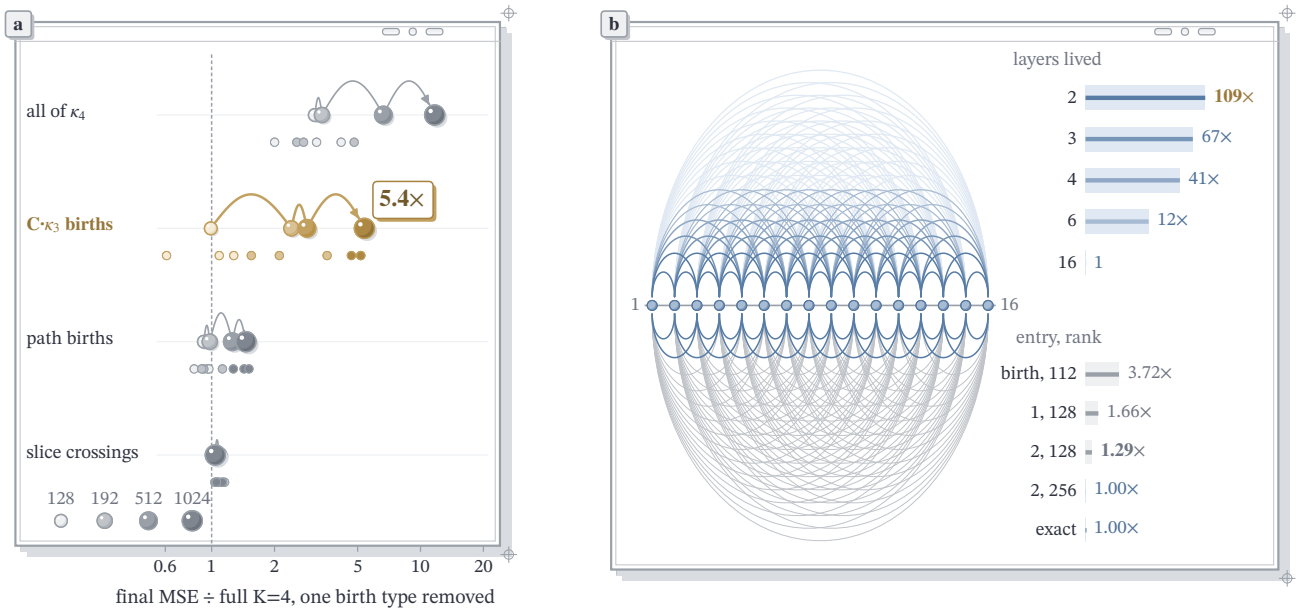


Figure 2: **Carried births, not local accuracy, hold the gain: removing C·κ3 births costs 5.4× on official networks 0–1 at width 1024, the largest single removal, and 4.7–5.2× on three networks frozen before running; letting every source live only two layers costs 109×.** (a) One row per birth type removed from full K=4, final MSE ÷ full K=4: beads are widths 128, 192, 512 and 1024, lighter to deeper and joined in that order; small dots are single networks frozen before running (at 128 and 192, and networks 5, 6 and 7 at 1024); the dashed line is no change (Table 3). (b) The K=4 state of official network 0 at width 1024, drawn as its reads: each arc joins a source's birth layer to a later layer that reads it, shaded by age. Above the axis, the ladder gives final MSE as a multiple of keeping every source for all 16 layers, when each lives 2, 3, 4 or 6; below, gray arcs are the reads made after a source enters a shared basis two transports after birth, and the ladder gives final MSE ÷ exact for each entry age and basis rank, on the same log scale.

4. Compute-Optimal State Pruning Inside the 2^{41}-FLOP Budget

A step here is compute-optimal in the grader's own sense: it stays only where the graded adjusted error falls, each step has its own official receipt, and all seven steps after V29 lowered it (Table 4, Figure 4a, Figure M), so every claim of this section is read off the scorer rather than a local proxy.

4.1 Chain B on the grader

Our scored entry, #336537 (step 8), runs chain B, our K=3 chain, from 504aldo's published V29 onward. Its two pruning pieces apply the rule of Section 3 from the other side: they find K=3 state that later reads barely use and stop computing it.

Table 4. Eight separate official scorer receipts. The first numeric column is the scored quantity: mean adjusted error over 50 leaderboard networks (lower is better). The ratio and paired 95% interval compare each step with its predecessor; "lower" counts networks whose adjusted error falls. Origin separates inherited V29, mixed work, our independent steps, the ported-and-refit SSC closure, and the learned corrector. The last row combines only our steps 3, 4, 6 and 7.

step · origin	adj ($\times 10^{-9}$)	÷ previous [95%]	lower /50	receipt	raw ($\times 10^{-8}$)	C/B
1 V29	5.418			#332957	2.133	0.254
2 mixed	4.449	0.821 [0.813, 0.830]	50	#334990	2.099	0.212
3 ours	4.286	0.963 [0.956, 0.970]	47	#335127	2.128	0.201
4 ours	4.258	0.993 [0.993, 0.994]	49	#335162	2.129	0.200
5 ported	4.030	0.946 [0.942, 0.951]	49	#335185	2.014	0.200
6 ours	3.525	0.875 [0.867, 0.882]	50	#335300	2.067	0.169
7 ours	3.494	0.991 [0.987, 0.995]	37	#335301	2.090	0.166
8 fitted	**3.145**	0.900 [0.890, 0.911]	49	**#336537**	1.853	0.169
ours only: 3, 4, 6, 7		**0.830** [0.820, 0.839]	**50**			

The interventions by step are: 1, published V29; 2, leaf-8 Strassen (also found by ngmitam), joins and dead-row trimming, with koushik_rudra's V34 dead-unit pruning and acco-cyber's V37 cost blocks (mixed); 3, hub level 6, SVD ranks 8/16 and basis 352 (ours); 4, wall-clock blocks with unchanged arithmetic (ours); 5, Puffi SSC closure ported and nine coefficients refit (ported); 6, young atoms, active set and twelve cost cuts (ours); 7, feedback thin legs at rank 4 (ours); 8, a fitted per-layer corrector. The final "ours only" comparison composes four receipts, not a separate submission.

Steps 1–7 cut C/B from 0.254 to 0.166 while raw moves from 2.13e-8 to 2.09e-8, −35.5% in mean adj, lower on all 50 networks. The steps made only of our changes (3, 4, 6, 7) multiply mean adj by **0.830** (95% interval 0.820–0.839), lower on all 50 networks (0.77 to 0.89 each), each a submission the grader ran inside its 8 GB and 120 s; step 6, active-set trimming and young-atom pruning with their cost cuts, gives 0.875 alone. Step 8 branches from chain B: the corrector lowers raw 11.3% at C/B 0.169 against 0.166. The pieces of chain B that are ours:

- **Active-set trimming.** koushik_rudra's V34 already restricts the third-cumulant leg products to the rows of neurons that are almost never active. A column of the (2,1) slice enters every consumer only through its neuron's first Hermite coefficient $w_1 = \Phi(\alpha)$, so we extend the cut to columns: those with $w_1 < 0.014$ skip the hub contraction and the shared-basis lift.
- **Young-atom pruning.** A young third-cumulant source born at layer b has one atom per neuron q of b, which enters three rank-one terms: $X_{1,q} \otimes e_q \otimes Y_{1,q}$, $M_q \otimes e_q \otimes e_q$ and the regeneration feed $X_{3,q} \otimes e_q \otimes y$. Its score is the norm bound $s_q = \|X_{1,q}\|\,\|Y_{1,q}\| + \|M_q\| + \|X_{3,q}\|\,\|y\|$ (for the first source, $3|w_{2,q}|\,\|A_{:,q}\|^2 + \|M_q\|$); at birth the atoms are sorted by s_q, the fewest holding 99.5% of $\sum_q s_q$ are kept, rounded up to a multiple of 32, and the rest are zeroed in every leg. As a single-block change on an earlier chain (at 99.6%) it moved adj 0.963× on the grader (#335175 → #335187).
- **SSC on the regeneration channel.** Puffi's response closure was written for 256 × 32; we put its Sym(C⊗C) slices next to V29's $\kappa 4$ regeneration law at 1024 × 16, decouple the three uses of λ and refit the 9 coefficients end to end on 12 public networks. As a single-block step on the grader (#335162 → #335185) it lowers raw 5.4% with C/B at 0.200 before and after.
- **Cost and wall-clock work.** Strassen depth and leaves, residual and feedback SVD ranks, a narrower shared basis, rank-4 thin legs, buffer layouts.

The grader's receipts for #336537: per-network raw from 1.47e-8 to 2.36e-8 on the 50 leaderboard networks, C/B from 0.166 to 0.178 on all 100, median wall time 85.5 s, slowest 102.1 s (limit 120 s), largest residual 0.357 s (limit 0.4 s), no failures.

Marius Binner's Phase 1 report prices the same trade-off for sampling state (pruning off: MSE −1.9%, compute +45%); chain B prunes deterministic cumulant atoms and scores each change on the Phase 2 grader.

Pieces switched off in the final chain (Table 5). We turned our three pieces off in chain B as submitted, one at a time and all together, with V29, the V37 cost blocks and the SSC closure fixed. Active-set trimming runs on V34's row schedule, so its switch turns V34's dead-unit pruning and V37's final-layer row band off with it: the active-set and all-three rows measure our extension and those row cuts together. A second switch turns off our column cut alone and keeps both. The representation arm puts the shared basis back to 384 columns and the residual and feedback-leg ranks back to 16, and runs every Strassen family at six levels.

Table 5. No corrector, mini 0–49. Each switch-off and its chain-B control run on the same machine per network (the column cut on 4-core CPUs, where B peaks at 5.8 GiB). Main column: ratio of mean adjusted errors, with a 95% paired-network bootstrap interval; median is the median per-network ratio. Peak memory above 8 GiB marks an unshippable counterfactual.

switched off	mean adj ÷ B [95%]	worse /50	median ratio	C/B	peak GiB
nothing: B, #335301	1		1	0.165	5.6
active-set trimming, with V34's and V37's row cuts	**1.160** [1.053, 1.227]	49	1.210	0.199	7.2
our column cut alone, both row cuts kept	**1.035** [1.032, 1.037]	**50**	1.032	0.171	5.9
young-atom pruning	1.044 [0.949, 1.101]	49	1.097	0.182	7.0
representation	1.049 [0.951, 1.109]	48	1.100	0.190	7.9
all three, with V34's and V37's row cuts	**1.278** [1.156, 1.355]	49	1.340	0.234	14.7

All three off raise mean adj 1.28× (95% interval 1.16–1.35), active-set trimming off, with the row cuts, 1.16× (1.05–1.23); each is worse on 49/50. Our column cut alone is worth 1.035× (1.032–1.037), worse on all 50 networks, and 1.034× on networks 4–49, outside the four on which chain B's cost settings were tuned. It wins on compute: without it C/B goes from 0.165 to 0.171, while raw moves 0.998× (0.996–1.000). Young-atom pruning off and representation off worsen 49/50 and 48/50 paired networks, with median ratios 1.097 and 1.100, mean ratios 1.044 [0.949, 1.101] and 1.049 [0.951, 1.109] over all 50, and mean ratios 1.095 and 1.103 excluding network 6 (Appendix A.5). With #335351's corrector held fixed on mini 0–7, the same switches cost 3.72×, 1.97×, 1.03× and 4.11× (Table A3b).

Against the other public write-ups. Table 4b compares the eight public Phase 2 write-ups posted by 20 October through the graded submission each is bound to (rules §6). Our bound submission #336537 ranks first of the eight at adj **3.145e-9**, **5.9% below the next** (hyojun_kwon's #335413 at 3.343e-9, 0.941× in adj and 0.762× in raw). Without the learned corrector, **#335301 ranks first among the chains of all eight write-ups that run without a learned terminal corrector or a layerwise mean-correction table**, at adj **3.494e-9** (0.987× in adj and 0.811× in raw against hyojun_kwon's #334316). Against both Artyom_Mazur's #333534 and Nurlanggs's #336225, #336537 is lower on all three graded metrics: adj (0.842× and 0.825×), raw (0.912× and 0.883×) and C/B (0.170 against 0.184 and 0.182).

Table 4b. The eight public Phase 2 write-ups posted by 20 October through the one graded submission each is bound to (rules §6), with #335301, chain B without our corrector, and #334316, #335413 without its corrector: adjusted error, raw error and C/B on the grader's 50 leaderboard networks from each official submission page, read on 19 October 2026, and #336537 over each.

entry	recipe	adj ($\times 10^{-9}$)	raw ($\times 10^{-8}$)	C/B	#336537 ÷ entry: adj, raw, C/B
#336537, ours	chain B with a corrector carried through W	**3.145**	1.853	0.170	
#335301, ours	chain B, no corrector (Table 4, step 7)	3.494	2.090	0.167	0.900, 0.887, 1.01
#335413, hyojun_kwon	#334316 with a learned corrector, state unchanged	3.343	2.431	0.138	**0.941, 0.762**, 1.23
#334316, hyojun_kwon	K=3, factorized κ3 at rank cap 7,680, fitted covariance correction	3.541	2.578	0.137	#335301 ÷: 0.987, 0.811
#333534, Artyom_Mazur	ARC's simple K=3 rewritten, scale-mixture κ4 closure, per-neuron κ4-diagonal corrector	3.736	2.032	0.184	0.842, 0.912, 0.92
#336225, Nurlanggs	V29 with an all-distinct κ4 edge term and a fitted (3,1) law	3.813	2.099	0.182	0.825, 0.883, 0.93
#336317, il.miscusi	e30_a with a carried κ4 core and scale-mode extraction	5.176	**1.721**	0.301	0.608, 1.077, 0.56
#334821, kai_leal_gonzalez	V29 with a two-term (3,1) closure	5.264	2.072	0.254	0.597, 0.894, 0.67
#330093, 504aldo	V29 (Table 4's step 1 reruns it)	5.418	2.133	0.254	0.580, 0.869, 0.67
#335813, DD_team	V29-route closure with exact cost cuts	27.33	3.206	0.853	0.115, 0.578, 0.20

C/B here is each page's "budget used", the mean over the 50 leaderboard networks; Table 4 averages all 100 graded networks (0.169 for #336537, 0.166 for #335301). The leaderboard of 19 October lists #336537, #333534, #336225, #334821, #330093 and #335813 with the same values to its two digits. Two further entries are cited in write-ups without being bound to them: #336226 (il.miscusi, 3.058e-9) and #336611 (hyojun_kwon, 3.044e-9), both unbound entries on LordProcciuto's third-party

e30_a engine with empirical calibration, a public MIT repository graded as #335981 at 3.116e-9. #335301 carries no learned corrector and no mean-correction table; the scalar closure constants of the propagation law (V29's λ and reference ratios, SSC's 9 coefficients) are fitted on public networks, as in every V29-derived chain.

4.2 The learned corrector

At each layer one MLP, shared across layers, reads 118 inputs per neuron and the layer's one-hot: 35 chain statistics of this layer (mean, variance, α, the ReLU coefficients, the κ4 diagonal and others), 8 of the previous layer's and a 32-dimensional state, both carried through W and W∘W, and 3 terms of the incoming correction; three hidden layers of 128 GELU units output the layer's new mean error g_l and the next state (54,561 parameters). The error is carried forward linearly, $u_l = \Phi_l \circ (W_l u_{l-1}) + g_l$, and added to the final prediction; the chain itself is unchanged. *Target:* on chain B's own trajectories, the error each layer adds, $\varepsilon_l = d_l - \Phi_l \circ (W_l d_{l-1})$, with d_l the true mean minus chain B's mean, scaled by its root mean square at that layer. *Data:* 3,048 networks: the 1,000 of the public full split, with ARC's `all_layer_means`, and networks 0–2,047 of Keenan Pepper's public corpus, 1024 × 16 ReLU networks with the challenge's N(0, 2/n) weights, rebuilt from its seeds, and per-layer means from 10^8 samples. The 100 public mini networks chose the configuration (for #335351) and are read after each epoch; the grader's networks are in neither. *Training:* Adam on the squared error of every layer through the whole 16-layer recurrence, 30 epochs, 37 min on one GPU. It gives 0.873× raw on the 100 mini networks, lower on all 100, and 0.887× on the grader's 50 leaderboard networks, lower on all 50 (#336537 against #335301). Artyom_Mazur (forum 18249) bounds what a per-neuron correction can learn at 4% of a K=3 chain's error; a corrector that reads through W and W∘W and carries its state across layers takes 11.3% off chain B's raw error on the grader. Across the 23 public Phase 2 write-ups, two independent teams directly port this (W, W∘W) corrector from our open repository into their own K=3 chains and verify comparable gains on the official 50-network grader: 18285 (#337436, adj 1.668e-9) measures 0.936× raw MSE on its all-safe grader pair (#337433 against #337477), and 18273 (#337389, adj 3.82e-9) measures 0.890× raw and 0.899× adjusted MSE (50/50 networks).

More networks, not more width. #335351 ships the same chain and the same corrector, configuration, seed and schedule, trained on the 1,000 full networks alone: 0.882× on the mini networks, 0.896× graded, adj 3.181e-9 (seeds 1 and 2 give 0.881× and 0.882× on the mini networks, lower on all 100 each time). Expanding the training set to 3,048 networks in #336537 lowers corrected raw on the mini networks by 0.989× (83/100) and graded mean adj by 0.989× [0.984, 0.994] (lower on 37/50, raw 0.990× [0.985, 0.994]) at the same C/B of 0.169, whereas a wider variant with state 64 and hidden 192 (#336536, raw 0.991× [0.986, 0.996]) raises C/B to 0.171 (adj 1.002× [0.997, 1.008]). Appendix A.5 lists every setting.

5. Pareto Frontier: Low-Rank Bilinear Compression and End-to-End Metering

The frontier of this section is the lower envelope of final-layer error against units over every configuration measured on official network 0, inside the budget and beyond it (Figure 5a). Low-rank bilinear compression holds a birth kernel as matrix pairs $\sum_{k \le r} u_k v_k^\top$ along its own direction, and end-to-end metering runs a whole chain with every operation billed by flopscope.

Budget arithmetic. The score is raw × max(0.1, U/1024) for U units per network, and U > 1024 turns the network into an all-zero prediction. To reach the leader's 1.50e-9 (9 October): raw 4.3e-9 needs U ≤ 355, raw 3.0e-9 needs U ≤ 512, and raw 1.72e-9 (full K=4) needs U ≤ 893.

5.1 What K=4 buys inside the budget

We added K=4 pieces that fit in 1,024 units to chain B, alone and in combination, in 71 settings of two kinds (Table 6 in Appendix A.8 lists the best of each piece, Figure 4c places them by cost): read at the output only or fed into the nonlinear moments of chain B so that its mean, covariance and κ3 carry it to every later layer. Every new coefficient comes from the cumulant algebra; ages and ranks were chosen on network 0, then frozen and retested on eight further networks. Each piece keeps n×n state at a fixed age and rank. Appendix A.3 writes out one of these pieces, star births with four distinct indices read at the output (356 units against chain B's 170).

Fed back, the pieces do more on network 0 than read at the output, up to −4.8% for star births with the (4,) and (3,1) slices; over networks 0–3 the same setting gives −1.0%, and network 2 gets 3.8% worse. On the eight frozen networks it gives −1.8% on average (median −2.4%, 95% t interval of the mean −3.6% to +0.03%), from −4.8% on the best network to +1.3% on the worst; the (4,) and (3,1) slices alone improve 7 of 8. Inside the budget the cheap K=4 pieces buy 0.9–1.8% raw on average where they help, and the sign is set network by network; at 976 units the best of them takes adj to 5.6× chain B's. The (2,2) slice at rank 8 and C·κ3 births at rank 4 hold 1.000 on the frozen networks. The 11.8× stays with full-rank C·κ3 crossings, non-tree births whose dense first read costs 3,070.5 units (Section 5.3), and slice groups that must live long (Section 5.2).

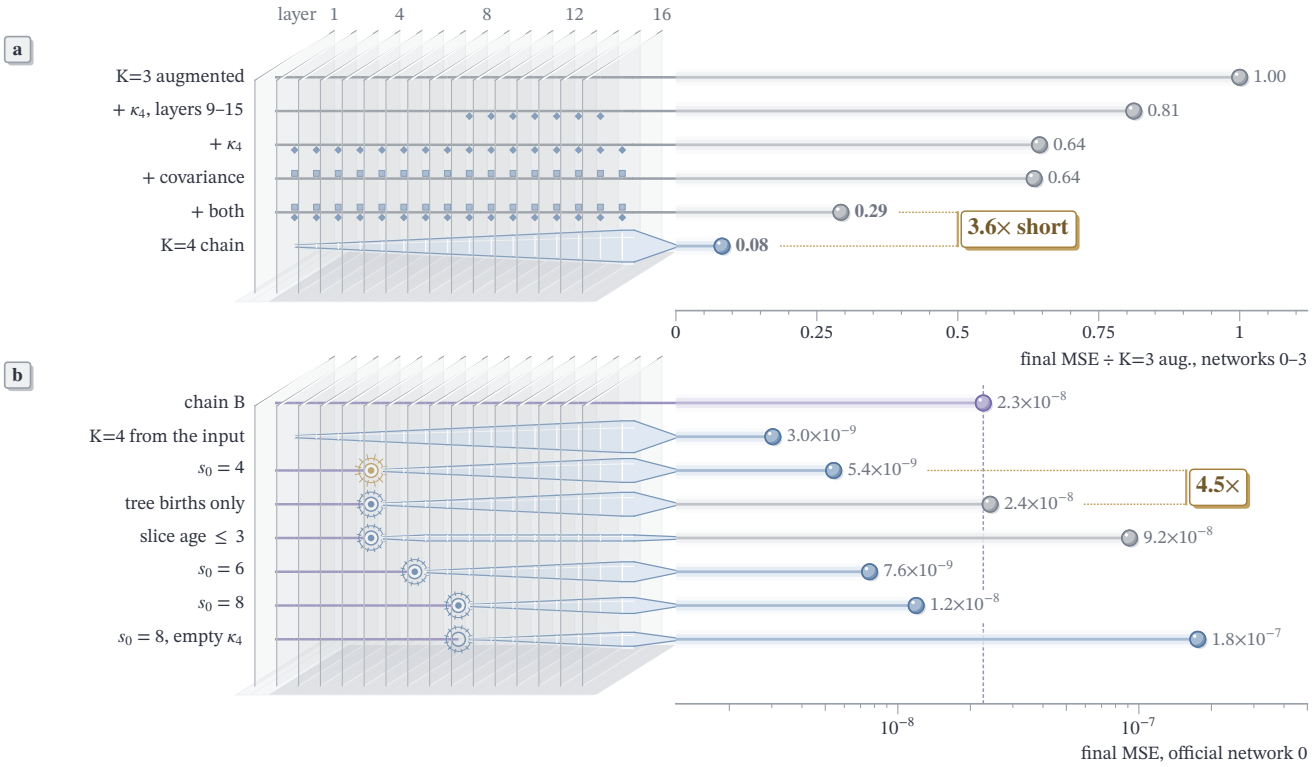


Figure 3: **Memoryless patches stall where the carried chain does not: on official networks 0–3, augmented K=3 handed the K=4 chain's own covariance and κ4 at every layer reaches 0.29 of its error, the K=4 chain 0.08, 3.6× lower; started at layer 4 from chain B, K=4 reaches 5.42e-9, and without its non-tree births it is 4.5× worse.** Each chain runs left to right through its 16 layers, drawn as glass plates: gray is augmented K=3, iris chain B, blue the K=4 state, whose width counts its live source groups (hatched: tree births only); a ring marks where the K=4 state starts, the open ring from an empty fourth cumulant, and each beam leaving layer 16 reads out the final MSE. (a) Official networks 0–3, as a ratio to augmented K=3; squares and diamonds mark the layers at which the exact covariance and κ4 are handed in. (b) Official network 0; the dashed line is chain B alone, and the slice-age row caps how long the slice groups of the tree-only run live.

5.2 Starting K=4 late, and which births it needs

The K=4 chain does not have to start at the input. On official network 0, chain B alone gives 2.26e-8. Running chain B to layer s_0 and the portable K=4 set from there gives **5.42e-9 at $s_0 = 4$**, 7.64e-9 at 6 and 1.19e-8 at 8 (Figure 3b). At the switch the K=4 state starts from what chain B already carries about the fourth cumulant: its (4,), (3,1) and (2,2) slices, its radial (2,1,1) feed as the initial core, and zero in every all-distinct-index entry. At $s_0 = 4$, **non-tree births carry all of the gain**: keeping only the tree-type births of the slice group (the graph terms that factor along shared edges, Section 2.6) gives 2.41e-8, 1.07× of chain B. Capping the slice age of the tree-only runs at 3, 2 or 1 layers gives 4.05×, 7.79× and 13.9× of chain B, so slice groups need long memory; a rank-8 or rank-32 covariance Krylov basis recovers none of the dropped non-tree part (7.80× against 7.79×). Without C·κ3 births the late start gives 1.13e-8, 2.1× worse.

5.3 What the complete chain costs in this representation

The call ledger (Table 7, Figure 4b) prices one computation in two representations. The compact one (legs and kernels, Sections 2.4–2.6) is counted row by row from metered operators, for the late starts of Figure 3b; births at layers s_0–13 are read at every later layer through 14.

- **The real κ3 width.** Chain B hands over three symmetric terms per live κ3 source, each expanded on three first legs, n columns each, one source per earlier layer: a C·κ3 birth at layer b carries 9,216·b columns (36,864, **55,296** and 73,728 at b = 4, 6, 8). One transport and one same-side read of it are 7 matmuls of n×n by n×9,216b, **63b units per (birth, read) pair** plus 3.2 for square factors (metered: 381.2 at b = 6). The C·κ3 row is this cost beyond the n-wide κ3 legs of the source micro-benchmark.
- **The C·κ3 crossing**, 4r matmuls at rank r (Section 2.2), is priced at rank 16, the highest rank measured (1.72× the exact crossing's error), so each total is a lower bound for reaching its column's accuracy.

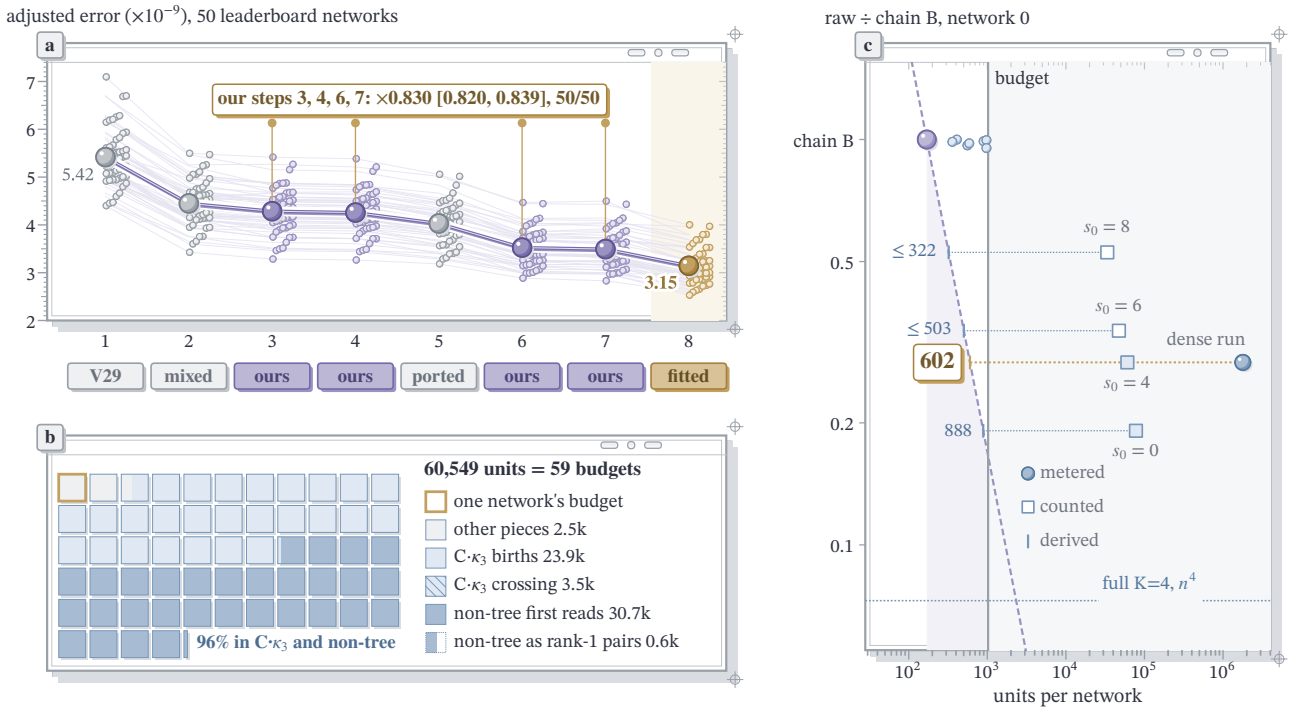


Figure 4: **Our own steps multiply the grader's mean adjusted error by 0.830 [0.820, 0.839], lower on all 50 leaderboard networks; the complete K=4 chain from layer 4, counted at 60,549 units, ties chain B's score at 602.** (a) Eight official scorer receipts: each thin line follows one of the grader's 50 leaderboard networks through the eight steps, large discs are the means, tags the origin of each step, the shaded column the scored entry (Table 4). (b) The complete chain from layer 4 on official network 0, counted in the compact representation (Table 7a): one tile is one network's budget of 1,024 units, filled by component; the last key row prices the non-tree reads as rank-1 pairs, extrapolated from n ≤ 64 (Section 5.4). (c) Cost against accuracy on network 0. The dashed diagonal is chain B's score: inside the budget, marks below it score better, and past the budget line a network scores as all zeros. Small discs are the in-budget K=4 pieces of Table 6; squares are the complete chain from layer s_0 at its counted lower bound (filled at the raw of the compressed run, open at the exact raw, a bound), each dotted line ending at the units where it ties chain B; the dense run (Table 7b) prices every operation of the $s_0 = 4$ configuration.

- **Non-tree first reads**: three n×n by n×n² contractions, **3,070.5 units** (3n), once per birth, all classes merged. **Young-leg transport** (static in the micro-benchmark): 2 + 4 matmuls per young source and layer.

Table 7a. Counted lower bound for the compact representation, units per network, official network 0, at the real κ3 width. Birth/read pairs and raw error are context for the four schedules. Old sources enter a k = 128 basis after two transports; old feedback enters a $k_3 = 128$ core after one layer. The $s_0 = 0$ accuracy is the portable set on the research K=3 chain. "Tie units" use the compressed raw; ≤ is a bound from exact raw. Every row is counted from metered operators; Table 7b (Appendix A.4) is the end-to-end dense run, and Table 12 meters the compressed configuration end to end.

compact · counted units	s_0=0	s_0=4	s_0=6	s_0=8
chain B, metered prefix	168	168	168	168
K4 sources, κ3 legs n-wide	510	311	227	153
young-leg transport	234	162	126	90
feedback, old core + young tree	519	384	314	244
old slice reads, 26-matmul form	2,730	1,430	936	546
C·κ3 births, real κ3 width	28,028	23,870	18,648	12,642
C·κ3 crossing, rank 16	5,824	3,520	2,304	1,344
non-tree first reads, dense	39,917	30,705	24,564	18,423
counted lower bound	**77,930**	**60,549**	**47,287**	**33,611**
C·κ3 + non-tree share	95%	96%	96%	96%
(birth, read) pairs	105	55	36	21
raw MSE, old pieces exact	3.03e-9	5.42e-9	7.64e-9	1.19e-8
raw MSE, compressed as counted	4.33e-9	6.38e-9		
units to tie chain B adj	888	602	≤503	≤322

What the ledger says (the dense run of Table 7b is detailed in Appendix A.4). In this representation the cost sits where the gain sits: C·κ3 and non-tree births carry the fourth-order gain (Table 3, Section 5.2) and 95–96% of every compact column. Everything else sums to 2,454 units from layer 4, 2.4 budgets. The C·κ3 row grows as $63 \sum_b b\,(L_r - b)$, cubic in depth (L_r the last read layer), since each birth's legs widen by one κ3 source per layer; the non-tree row is 3n units per birth, n^4 arithmetic. Compressing old pieces keeps the source and feedback rows small and costs accuracy (Section 3): from the input 4.33e-9 against 3.03e-9 exact, at ≥ 1,321 units before the table's further rows; from layer 4 the run reaches 6.38e-9 against 5.42e-9, so 602 units tie chain B's score on network 0 (2.26e-8 at 170 units).

5.4 Non-tree births: the gain is in the directions, not the mean

Non-tree births hold all of the late-start gain and half the layer-4 ledger (30,705 units); Table 8 tries cheaper forms, and Appendix A.6 runs direction, sources and long-lived state as one per-layer chain with the shape and units of each step.

Table 8. Non-tree births in cheaper forms: share of their gain kept, (tree only − X) ÷ (tree only − exact), late-start suffix from layer 4. Units at n = 1024; rank-r costs are flopscope counts at n = 16–64 carried along their measured $n^{2.98}$. Last two rows: on chain B itself, no K=4 tree state, raw against chain B on the same networks and machine, metered at 1024; births at layers 13–14, then at the last 1, 2, 4 and 8 layers.

non-tree births read as	directions kept	n, networks	gain kept	units
exact, dense first read (Table 7)	all	1024, 0–1	100%	3,070.5 per birth
averaged over the current layer's weights	none	1024, 0–1	−36%, −46%	3.5 per source
covariance Krylov basis, rank 8 and 32	covariance	1024, 0	none	
Rademacher sketch, rank 1 and 16	random	64, seed 5	40%, 92%	
matrix pairs from the kernel, rank 1	the kernel's	64, seeds 5–7	**83%, 79%, 84%**	11.6 + 6.0 per read
matrix pairs from the kernel, rank 2	the kernel's	64, seed 5	101%	13.7 + 12.0 per read
pairs from chain B's factors, nearest source	chain B's mean	1024, 0–3	raw −0.010%	30.1 per birth, 66.2 in all
pairs from chain B's factors, every source	the kernel's, one power step	1024, 0–1	raw −0.7 to −1.8%, +0.1 to +3.0%	762 per birth; 9.1–46 in one joint space

The mean is empty. Keep every earlier layer at its true weights and average the non-tree contraction over the current layer's weights alone, Gaussians of variance 2/n: Wick pairing leaves two trace sums of $H = P^\top P$ against the birth kernel. At the first read H is diagonal and the kernel lives on all-distinct indices, so the average is exactly zero; over the 55 older sources per network it carries 1.5–3.7% of the readout energy in the (4,) and (2,2) channels and none in the other two, the fluctuation around it 95.6–98.6%. In the chain (chain B to layer 4, networks 0–1) it ends 17–18% worse than tree births alone. The non-tree term does not self-average.

One axis compresses. Unfold the birth kernel $B_{abr} = 6\,\mathrm{sym}_{ab} T_{rab}$ as $F_{r,(ab)}$ and compress only the repeated-index axis with a randomized range finder on F itself, $F \approx UV$, each V_s a full n×n matrix. After transport Q, the pair $A_s = Q\,\mathrm{diag}(U_{:s})Q^\top$, $D_s = QV_sQ^\top$ gives the four same-side slices by matmuls, e.g. $(4,) = \sum_s \mathrm{diag}(A_s) \circ \mathrm{diag}(D_s)$; at full rank it equals the dense read to 2e-16. At n = 64 (16 layers, full K=4 to layer 4, MC truth noise below 0.01% of the gain) rank 1 keeps 79–84% on three networks and 83–99% over four sketch seeds, with a sketch variance of 2.8% of the gain; at layer 8, ahead of the deep layers where the reference's negative-variance clipping acts, it keeps 70%. A Rademacher sketch ignores the kernel's directions and pays in variance: 267% of the gain at rank 1. Extrapolated from these metered runs to 1024, rank-1 pairs cost **557 units for the whole non-tree part from layer 4**, against 30,705 dense (Figure 4b); counted at 1024 on the six non-tree births of Table 11, rank 1 costs 330 units.

Births from chain B's own factors. In chain B the kernel is not stored: each birth is built from chain B's κ3 factors. Along chain B's normalized mean and from the nearest source only, two births with every read cost **66.2** units at 1024 and move raw by 0.010% over networks 0–3; the kernel's own direction, every source in one core and the price of that core have measured replacements, from 99.80–99.99% of the best rank-1 energy to **9.1 units per birth** (Appendix A.6).

What they buy on chain B. With the kernel's direction and every source in place, nothing compressed, births at the last 1, 2, 4 or 8 layers read once give raw 0.993–0.982 of chain B on network 0, an improvement of 0.7–1.8%, and 1.001–1.030 on network 1, a loss of 0.1–3.0%. Kept as K=4 sources in the late-start chain of Section 5.2 with slice births alone, tree births leave 1.07× and 1.11× of chain B's raw on the same two networks and tree with non-tree births reach 0.55× and 0.66× (the portable set, with star and C·κ3 births, 0.24× on network 0); at n = 64 rank-1 pairs living that way hold 79–84% of the gain. The non-tree gain in these tests comes with the K=4 state the births live in, the line of Figure 3 and of Section 6, row 8.

5.5 Where the 602-unit target is bound: two births

Table 7a counts the complete chain from layer 4 at 60,549 units, this representation's counted lower bound, and its compressed run ties chain B's score at 602. We split that distance by state family and measured what each family carries, what a

local ruler says about it, how far deleting state goes and how far compressing it goes: 46 runs of the K=4 suffix on one GPU and 18 runs of chain B with K=4 pieces on CPU, on official networks 0 and 1, eight runs with the two families compressed on network 0, every raw recomputed from its saved prediction, chain B with an exact (3,1) slice on eight networks, and the compressed chain metered end to end by flopscope (Tables 9–12, Figure 5).

Table 9. The $s_0 = 4$ column of Table 7a (official network 0) by state family. Units: Table 7a's rows regrouped. Read weight: the family's share of the first-order read weight summed over every counted (birth, read) pair (defined below). Without ÷ with: final MSE without the family over the chain with it, from layer 4 and, for the same births, from the input (Table 3; the non-tree term in the portable set, Appendix A.4; star births on networks 5–7, Section 3).

state family	units	share	read weight	without ÷ with, from layer 4	without ÷ with, from the input
C·κ3 births (real κ3 width, rank-16 crossing)	27,390	45.2%	33.2%	**2.08**	**5.39**; frozen 5.17, 4.67, 5.18
non-tree births (dense first reads)	30,705	50.7%	**6.4%**	**1.95**; 1.94 with C·κ3 and star off	1.56, $C_{jk}\,\kappa_{3,ijk}$ alone, portable set
tree slices and (2,1,1) → K3 feedback	1,814	3.0%	30.3%	alone, 1.07× chain B's error	slice crossings 1.04
star births, K=4 sources, young legs	473	0.8%	30.2%	1.10, C·κ3 off	1.79–2.05, frozen 5–7
chain B, metered	168	0.3%			
total	60,549	100%		5.42e-9, 4.2× below chain B	1.73e-9

Units and gain sit in the same two families. C·κ3 births hold 27,390 units (45.2%) and non-tree births 30,705 (50.7%). Removing either from the layer-4 chain multiplies its final error by about 2: 2.08× without C·κ3 births, 1.95× without the non-tree births of layers 4–12. Everything else, tree slices with their feedback, star births, the source machinery and chain B's metered prefix, sums to 2,454 units, 2.4 budgets, and does not beat chain B alone: tree slices without the two families end 1.07× worse than chain B, and with star births added 2.87× worse. From the input C·κ3 births are the largest single removal, 5.39× (4.67–5.18× on the frozen networks); the non-tree term $C_{jk}\,\kappa_{3,ijk}$ alone is worth 1.56× in the portable set (Section 3), and removing star births, part of the other 2,454 units, costs 1.79–2.05× on networks 5–7.

Table 10. Chain B (#335301, no corrector) with its (3,1) slice of κ4 replaced at layers 1–14 by the exact one from public 10^8-sample pair moments, the object of forum 18246, against full K=4 on the same network (Table 1). Share of the K=4 gain: ln(chain B ÷ with exact (3,1)) ÷ ln(chain B ÷ full K=4).

network	chain B	with exact (3,1)	ratio	full K=4 ÷ chain B	share of the K=4 gain
0	2.27e-8	1.75e-8	0.770	0.073	**10.0%**
1	2.01e-8	2.02e-8	1.004	0.090	−0.2%
3	2.57e-8	2.07e-8	0.804	0.070	8.2%
4	2.35e-8	2.14e-8	0.911	0.070	3.5%
5, 6, 7, 8	1.97–2.31e-8	1.83–1.94e-8	0.876, 0.926, 0.837, 0.882		
nets 0, 1, 3–8			**0.873**, 7/8 lower		

Exact local content leaves 90% of the gain to the births. The exact (3,1) slice lowers chain B's raw error to 0.873 over eight networks, 7 of 8 lower, in line with the −18.1% that forum 18246 measured inside V29. Against full K=4 on the four networks where both ran, it recovers 10.0% of the K=4 chain's log gain on network 0, 8.2% and 3.5% on networks 3 and 4 and −0.2% on network 1: at least 90% of the gain lives in the K=4 chain's own births (Table 9). The layer-wide patch of Section 3 says the same at full strength: handed the K=4 chain's covariance and κ4 at every layer, augmented K=3 stops 3.6× short of the K=4 chain.

A first-order read weight ranks the gain last. We scored every counted (state, read) pair by the first-order effect of its read on the final mean, through the next layer's mean and variance, and in a second version through κ3 as well. Non-tree births get 6.4% of the summed weight on network 0 (6.5% with κ3, 7.1% on network 1) and carry 1.95×; tree slices and star births get 30% each, and alone they end worse than chain B. Keeping reads in order of this weight is 3.4–22× worse than chain B at every size measured from 600 to 20,452 units, on both networks and both versions (11 configurations), while the reverse order at 20,451 units gives 0.703×. A read's local effect does not price what later layers do with the state: the split that James Henry's third-cumulant carrier showed in Phase 1 (Table 0), here at K=4 and 1024 × 16.

Deleting state stays 45–75× from the tie. The deletion rule on the lower envelope keeps whole lives and a consistent state: tree slices born at every layer, non-tree and star births born early, from layer 4 to e. It gives 0.937× chain B's raw at 8,194 units, 0.833× at 11,296, 0.749× at 14,396, 0.669× at 17,492 and 0.615× at 20,586 (network 1: 0.910× at 11,296, 0.703× at 20,586). Each part is needed: without the late tree slices the same window is 1.80× worse than chain B at 20,256 units, with star births at layers 10–13 in place of the late tree slices 3.21× worse at 20,443, and a late start at layer 10 that keeps every family gives 0.785× at 20,453. Inside the budget the envelope is chain B with K=4 pieces fed back, 0.953× at 976 units (Section 5.1). Chain B's score prices each point: at 0.615× a network may spend 276 units to tie it, and the rule spends

20,586, 75 times as many; at 0.937× the factor is 45. Of the rule's 20,586 units, 18,423 are the dense first reads of its six non-tree births. Deleting births moves along the envelope; reaching 602 at the layer-4 chain's accuracy needs the same state in a hundredth of its units.

Compressing the births moves the envelope. We kept the 20,586-unit rule's births and held its six non-tree births as matrix pairs from their own kernels, the one-axis form of Section 5.4 run at n = 1024 (Table 11). At rank 4 the rule reaches 0.788× of chain B's raw at **3,258 units**; the cheapest deletion point at that raw or lower costs 14,396 units (0.749×), so compression reaches it with 77% fewer units, and the factor over the tie falls from 75× to **15×**. Rank 1 gives 0.942× at 2,493 units, where deletion needs 8,194, and rank 16 gives 0.681× at 6,321; all three join the lower envelope of Figure 5a. The six non-tree births cost 330, 1,095 and 4,159 units at ranks 1, 4 and 16, against 18,423 dense; the rest of the rule, 2,163 units, is led by the old tree slice reads at 1,430. Holding both families as pairs over every birth from layer 4 to 13 reaches 0.555× of chain B's raw at ranks 8 and 8, counted at 9,345 units plus 23,870 for the real $\kappa 3$ leg width of the C·$\kappa 3$ births, Table 7a's 63b units per (birth, read) pair. That width is the next cut. A same-box rerun of the 20,586-unit rule reproduces its raw to 0.02%; every count is in Table 7a's representation on network 0, and the metered run below prices the same state end to end.

Table 11. The two families as matrix pairs from their own kernels (Section 5.4's one-axis form), official network 0, n = 1024. Units are counted as in Table 7a, with the real $\kappa 3$ leg width of every C·$\kappa 3$ (birth, read) pair; the next column leaves that width out. Raw from each saved prediction; tie: the units at which that raw ties chain B's score. Last column: the fewest units of a deletion-grid point (Figure 5a) with no higher raw.

births kept, held as pairs at rank	units	without $\kappa 3$ legs	raw ÷ chain B	units ÷ tie	deletion, units
the 20,586-unit rule, dense	20,586		0.615	75×	
the same births, non-tree rank 16	6,321		0.681	25×	17,492
the same births, non-tree rank 4	**3,258**		0.788	**15×**	14,396
the same births, non-tree rank 1	2,493		0.942	14×	8,194
births at 4–6, both families rank 8	13,427	5,237	0.821	65×	14,396
births at 4–6, both families rank 4	11,926	3,736	0.890	63×	11,296
births at 4–13, both families rank 8	33,215	9,345	0.555	109×	32,687
births at 4–13, both families rank 4	29,893	6,023	0.635	112×	20,586

The discriminating test, metered end to end. We ran the complete compressed chain from layer 4 on official network 0 with every operation executed and billed by flopscope 0.12.3: chain B to layer 4, then old sources merged into one k = 128 core, old feedback into a $k_3 = 128$ core, the (2,2) and (2,1,1) slice kernels dense and every birth kept for life (Table 12). Accuracy holds end to end: the final raw is **6.37e-9**, on the 6.38e-9 at which Table 7a sets the 602-unit tie, and each of its 12 layers is within 0.004–0.14% of the GPU research chain. The meter reads **756,620 units** (C/B 739; 6.2 hours and 114.8 GiB on a shared CPU host), 12.5× this representation's counted lower bound and 0.42× the dense run of Table 7b; network 1, run through layer 14, bills the same units layer for layer and reaches 6.48e-9 there. Table 12 puts the excess in the groups that hold the births of Table 9: the two-leg groups, which carry the C·$\kappa 3$ and star births, read for 193,000 units against 27,863 counted, and the slice groups for 481,007 against 32,135, 345,045 of them the dense (2,2) and (2,1,1) kernels that carry the non-tree births. In this run $\kappa 3$ is stored dense, so every (2,1,1) feedback lands as an n^3 tensor and each of those reads is an n^4 product; Table 7a counts them with $\kappa 3$ and the feedback kept in factors. The metered chain stands at 1,257× the 602-unit tie, and 88% of its units, 665,989, are those n^4 reads; the two Table 12 rows that hold the two families bill 538,045, 71%.

Matrix pairs move the metered frontier. Holding the (2,1,1) birth kernel as matrix pairs along its own direction removes the dense kernel reads and the old-feedback projection. With 8 pairs the meter reads 512,138 units at raw 8.34e-9, the pairs' reads and compression costing 2,133 against 30,705 dense. With **32 pairs, run through all 16 layers, the chain holds 6.95e-9 at 525,045 units** (C/B 513; 3.96 hours and 79.2 GiB on 16 CPU threads), its final layer recomputed from the saved prediction by an independent check (Table 12). Against the dense kernels it bills 30.6% fewer units for 9.1% more raw, and against 8 pairs 16.7% less raw for 2.5% more units; measured against the fixed 602-unit target, the metered chain moves from 1,257× to 872×. The three metered points join as one line on Figure 5a's frontier. Compressing the whole kernel compresses its tree terms with it (8 pairs keep 15–23% of the kernel's energy at birth, 32 pairs 36–46%), where Table 11 and Section 5.4 compress the non-tree part alone. In the 32-pair run the slice groups' (4,), (3,1) and (2,2) parts are read on legs for 265,661 units and the pairs for 5,483, and 87% of its units remain reads of the slice and two-leg groups, n^4 products because $\kappa 3$ is still dense.

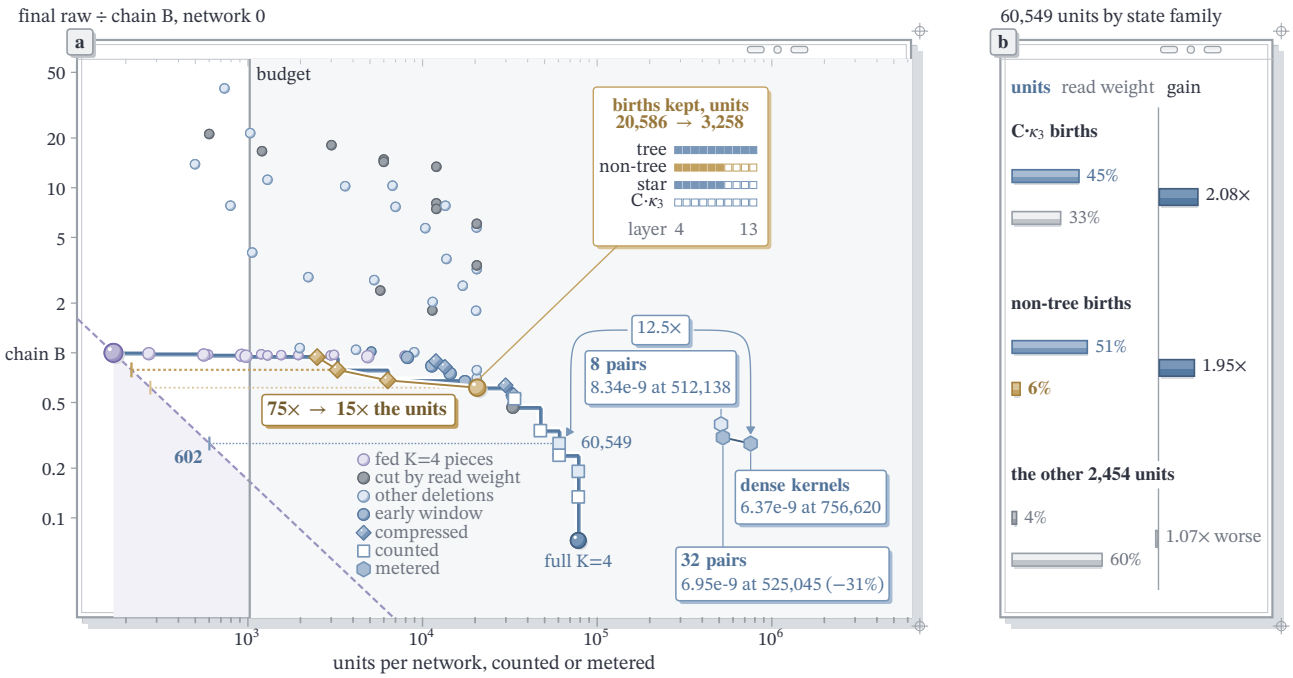


Figure 5: **One compute–error frontier on official network 0, inside the budget and beyond it: deleting state stalls at 75 times the units its accuracy may cost (0.615× of chain B's error at 20,586 units), rank-4 non-tree pairs cut that to 15 times (0.788× at 3,258), and the metered whole chain falls from 756,620 units with dense kernels to 525,045 with 32 matrix pairs at 6.95e-9.** (a) Every measured configuration on official network 0, final raw over chain B's against counted units; the stair is the lower envelope over both batches, the best raw at no more units. The dashed diagonal is chain B's score, so a point's horizontal distance to it is the factor by which its units exceed what its accuracy may cost; past the budget line a network scores as all zeros. Gold: the 20,586-unit rule and, along the gold line, the same births with non-tree pairs of rank 16, 4 and 1, the dotted lines ending at their ties (Table 11); blue diamonds hold both families as pairs. Squares are counted (Table 7a); hexagons are metered end to end by flopscope (Table 12) and joined as the metered line, each capsule naming its run: the compressed chain from layer 4 with dense (2,1,1) kernels, 6.37e-9 at 756,620 units, and the same chain with its (2,1,1) births as 32 matrix pairs, 6.95e-9 at 525,045 (31% fewer units), and as 8 pairs, 8.34e-9 at 512,138. The arc joins the counted chain from layer 4 (60,549 units) and the dense metered run at one raw: the meter reads 12.5× the count. Chain B's disc and the fed K=4 pieces give the frontier's scored side. The card shows which births the rule keeps, layer by layer. (b) The layer-4 column of Table 7a by family: share of the units, share of the first-order read weight and the gain, final error without the family over with it; for the other 2,454 units, the tree slices alone against chain B. Tables 9 and 10 list the values.

Table 12. The compressed chain from layer 4, metered end to end by flopscope 0.12.3 on official network 0, against Table 7a's $s_0 = 4$ count, row group by row group: the (2,1,1) birth kernels dense (final raw 6.37e-9, C/B 739) and as 32 matrix pairs over all 16 layers (6.95e-9, C/B 513). Rows rounded.

Table 7a rows	counted	dense, metered	32 pairs, metered	what the metered runs do
chain B to layer 4	168	25	25	chain B's prefix to layer 4 (168 is its whole network)
C·κ3 and star births, sources, young legs, crossing	27,863	193,000	193,000	young two-leg groups read exactly, six n^4 products a read; old ones in one $k = 128$ core
feedback, old core and young tree	384	33,724	0	dense: old feedback in a $k_3 = 128$ core; pairs: no projection
old slice reads, 26-matmul form	1,430	135,961	265,661	slice groups on legs; with pairs, their (2,2) parts as well
non-tree first reads, dense	30,705	345,045	5,483	dense: (2,2), (2,1,1) kernels for life; pairs: the 32 pairs
outside the count		48,864	60,877	κ3 expanded once (12,389), linear steps, pair compression, the rest
total	60,549	756,620	**525,045**	12.5× and 8.7× the count; 1,257× and 872× the 602-unit tie

Three measurements, one next cut. The family split puts 96% of the counted units and about 2× of the error in each of the C·κ3 and non-tree births (Table 9). The first compression takes the non-tree births to rank-4 pairs at 15× the tie, while

two-family pairs still pay 23,870 units of κ3 leg width (Table 11). The meter puts 88% of 756,620 units in reads of the slice and two-leg groups, 71% in the two rows that hold the two families, each read an n^4 product because κ3 is dense, and 32 matrix pairs still leave 87% of 525,045 units in those reads (Table 12). All three point at the representation of κ3: a dense old part plus young factors, with the (2,1,1) feedback written into the young factors as their source in place of an n^3 tensor, so that two-leg and slice reads multiply factors at n^3, the form in which Table 7a counts them at 63b units a pair and 26 matmuls. The non-tree births already have their compressed form, matrix pairs along the kernel's own direction on the non-tree part alone, 330–4,159 units for six births at ranks 1–16 against 18,423 dense (Table 11). The other 2,454 units, four times 602, need a shared core for old slice groups that keeps the channels a transport-energy basis loses (Section 6, row 17), as old sources and old feedback already compress at 1.29× and 1.03×. The test of each cut stays the whole chain, with each birth's lifetime and feedback, as ARC's deduction–projection account asks of any projection step and as James Henry's carrier (Table 0) showed for the third cumulant.

6. Falsified Closure Hypotheses and Structural Boundaries

□ Readings are at width 1024 on the official networks unless a row notes otherwise, all measured here; each row states a common closure or cost hypothesis, the reading that falsified it and the structural invariant it reveals.

#	Hypothesis tested	Reading at 1024 × 16	Structural invariant revealed
1	Capping K=4 source age at m = 2, 3, 4, 6 keeps the gain	109×, 67×, 41×, 12× worse	*Uniform lifetime cuts fail; sources must live long*
2	K=4 sources in the shared basis at birth keep the gain	3.72× worse (1.13e-8)	*Young sources must stay exact before shared projection*
3	A rank 1–16 star crossed pairing keeps the gain	18× to 15× worse	*The diagonal star crossing must stay exact (3 matmuls)*
4	A rank 1–16 SVD of the C·κ3 crossed kernel keeps the gain	2.21× to 1.72× worse	*The C·κ3 crossed kernel stays near full rank*
5	Dropping all two-leg crossed pairings keeps the gain	18× worse (5.47e-8)	*Crossed index pairings carry essential state content*
6	Truncating feedback CP columns to 128 keeps the gain	6.76e-7, 223× worse, 195,308 units	*Column truncation on CP terms loses the feedback state*
7	Newborn feedback in a three-axis core keeps the gain	5.10–5.57e-9 for $k_3 = 64$–256	*Core rank alone does not repair newborn projection loss*
8	Keeping tree feedback exact and projecting the rest pays	6.48–7.23e-9; refit basis keeps 0–29%	*Tree and non-tree feedback terms cancel when split*
9	Removing one K=4 output channel keeps the gain	1,148× to 714,978× worse	*All four output channels close the recursion together*
10	Star births plus same-side n^3 slices keep the K=4 gain	3.65× worse than full at 512, 6.3× at 1024	*Omitted births carry more weight as width grows*
11	A learned final-layer readout on 1,000 networks pays	best 0.979×; re-propagated, 1.09–1.60×	*Carrying state beats output fits (Section 4.2: 0.887×)*
12	More K=3 cost work (Table 4, step 3) keeps paying	adj −3.7%, raw +1.4%	*The K=3 cost search we ran saturates near 3.7%*
13	A sixth Strassen level in V29 (leaf 16) is cheap enough	C/B −5.4%, 7.3 GB, 1.25–1.9× wall time	*The 8 GB memory and 120 s wall-time limits bind first*
14	Truncated SVD of young K=3 sources (k = 16) pays	256: cost −11%, raw +16%; 1024: >180 s	*Cheap rank cuts lose readout accuracy and wall time*
15	Patching K=3 with the K=4 chain's C and κ4 keeps the gain	0.29×; κ4 late, 64 directions, 0.81×	*Memoryless patches miss the carried K=4 chain (0.08×)*
16	Starting K=4 at layer 8 from an empty κ4 keeps the gain	1.76e-7, 15× worse than seeded	*Late starts must inherit chain B's κ4 slices*
17	Old slice groups in a k = 64 energy basis keep the gain	2.76e-7, 51× worse; 84% of norm lost	*Local transport energy picks the wrong readout directions*
18	C·κ3 births as CP columns on chain B are cheap enough	55,296 columns; 110 + 271 units a pair	*Read cost grows with the rank of the carried κ3*
19	One CP pool of 2,048–5,120 atoms for κ3 keeps the gain	72× to 135× worse; uncapped 0.994×	*κ3 does not compress into a global pool of CP atoms*
20	A late start with tree births only, or a Krylov r 8/32 basis, keeps the gain	1.07× of chain B; 7.80× against 7.79×	*Late-start gain sits in non-tree births*

7. Three Structural Laws of High-Order Mechanistic Estimation

The measurements on both sides of the 2^{41}-FLOP budget reduce to three empirical laws that govern when high-order cumulant state improves final-layer predictions and where its cost sits.

Law 1, unaugmented fourth-order scaling. The fourth-order gain grows with n/L at the exponents of ARC's bound: Wu et al. (2026, Section 5) conjecture that the error of the recursion truncated at order K lies near $c_K(L/n)^K$, so moving from

$K = 3$ to unaugmented $K = 4$ gains a factor that grows with n/L, whereas the loss that 504aldo's F29 measured at $n/L = 8$ belongs to the augmented variant while on the same cells the unaugmented recursion gains. At depth 16, full unaugmented K=4's final-layer MSE falls as $n^{-3.95}$ (95% interval 3.68–4.08) and augmented K=3's as $n^{-3.08}$ (2.95–3.17), matching the 4 and 3 of ARC's conjectured bound $c_K(L/n)^K$ at fixed depth (Table 2, Figure 1b); at 1024×16 the ratio is 11.8× on all five official networks (11.0–12.5×, Table 1), and at $n/L = 8$ unaugmented K=4 beats augmented K=3 on all six frozen networks (2.0–3.7×) while augmented K=4 loses 1.6–4.0× there (Table 2b, Figure 1a). This law would be refuted if unaugmented K=4 failed to beat augmented K=3 on frozen networks at $n/L = 8$, or if a width scan at depth 16 placed unaugmented K=4's scaling exponent inside augmented K=3's 95% interval 2.95–3.17.

Law 2, downstream readout observability. The value of a piece of carried state is the amount later layers read from it into the final estimate rather than how exact it is at the layer where it is born, because to first order the final-layer error of a truncation δ of a state s is $\langle g, \delta \rangle$ with g the downstream readout vector, so a truncation chosen by local norm or local exactness does not control $\langle g, \delta \rangle$ (Proposition 2). Augmented K=3 handed the K=4 chain's own exact covariance and $\kappa 4$ at every layer reaches 0.29 of its error, while the K=4 chain itself reaches 0.08 on official networks 0–3 (Section 3, Figure 3a); the exact (3,1) slice of $\kappa 4$ inside our K=3 chain recovers at most 10.0% of the K=4 chain's log gain (Table 10); ending every K=4 source after two layers costs 109× on official network 0 (Section 6, row 1), old slice groups in a $k = 64$ energy basis lose 84% of norm and end 51× worse (Section 6, row 17), and ordering reads by a first-order read weight is 3.4–22× worse than chain B (Section 5.5). This law would be refuted if a memoryless fourth-order closure reached the K=4 chain's 0.08× of augmented K=3's error on official networks 0–3 at 1024×16, or if old slice groups in a $k = 64$ energy basis matched the exact chain's final error on official network 0.

Law 3, two-family bottleneck. Beyond the budget, 96% of the K=4 units from layer 4 and about 2× of the error each sit in C·$\kappa 3$ births and non-tree births, making the representation of these two families the measured next cut: each C·$\kappa 3$ birth reads through the legs of every live $\kappa 3$ source so its read cost grows with $\kappa 3$'s width (Section 5.3), the non-tree terms have no tree-term factorization yet and are read densely (Section 2.6), and everything else sums to 2,454 units, 2.4 budgets. From layer 4 the complete chain counts 60,549 compact units, 96% of them in two birth families, C·$\kappa 3$ births (27,390 units, 45.2%) and non-tree births (30,705, 50.7%), and removing either multiplies the layer-4 chain's final error by about 2 (2.08× and 1.95×, Table 9); the chain ties our scored entry at 602 units (Table 7a), and storing the non-tree births as rank-4 matrix pairs takes the best measured point from 75× to 15× the units at which it ties (Table 11, Figure 5a). This law would be refuted if a K=4 subset that excludes both C·$\kappa 3$ births and non-tree births beat chain B's adjusted error inside 1,024 units on official network 0 at 1024×16, or if whole-birth deletion reached the 602-unit tie line without compressing those two families.

Limitations: compute and memory. Full K=4 at 1024×16 counts at least 77,930 units per network in its portable subset against the 1,024-unit budget (Table 7a), and each run requires 60.4 GB of GPU memory and 120 GB of births in pinned host memory. In Table 12, the complete metered chains run on official network 0 (3.96–6.2 hours and 79.2–114.8 GiB on a CPU host), with official network 1 traced through layer 14 at 6.48e-9. Reaching the 1,024-unit budget requires the next engineering step identified in Section 5.5: writing the young (2,1,1) feedback directly into a factored $\kappa 3$ basis so that slice and two-leg reads contract at n^3 instead of n^4.

Implications for mechanistic estimation and trained networks. The unit of mechanistic progress is carried state that later layers read rather than local closure order. Closures that ignore downstream readout alignment lose the chain's gain, including memoryless fourth-order closures (0.29× against the chain's 0.08×; Section 6, row 15), local-norm truncation (84% of norm lost, 51× worse; row 17, Proposition 2), and uniform lifetime cuts (109×, 67×, 41×, 12× worse; row 1), whereas end-to-end gains come from cross-layer factored transport (exact K=4 at 1024×16 on one GPU, Section 2), a dense old part of $\kappa 3$ plus young factors (the measured next cut for the 87–88% of metered units in n^4 reads, Section 5.5), and readout-guided pruning (Table 4). Together, exact factored K=4 at 1024×16 (1.72e-9, 11.8× below augmented K=3), readout-pruned chain B (#336537 at 3.145e-9 and #335301 at 3.494e-9, first in both classes across the eight public Phase 2 write-ups posted by 20 October), and independent Phase 2 replications across seven teams (porting our W, W∘W corrector at 0.936× and 0.890× grader raw MSE in 18285 and 18273, testing our C·$\kappa 3$ birth on the exact (3,1) slice at 31 to 47% explained energy and 13 to 19% of the reborn remainder against 7.5 to 7.8% from carried K=3 legs in 18281/6, and citing our 1024×16 measurements and column cut in 18264, 18272, 18284 and 18249) show that mechanistic estimation advances by preserving high-observability carried state end to end. As a testable prediction derived from these random MLP results for trained networks, learned weight anisotropy can concentrate higher-order deviations in selected directions, hence local-norm truncation may lose directions that downstream layers read; exact factored transport $Q_{b \to l}$ and the first-order readout inner product $\langle g, \delta \rangle$ offer a mathematical framework to track them.

8. Reproduce

One command on a laptop. `python3 laptop_exact.py` runs ARC's exact K=4 and augmented K=3 live on three 128×16 networks with every FLOP charged by flopscope: K=3 ÷ K=4 of 4.168×, 2.002× and 3.179×, in 14.87 s of wall

time and at most 1,570.9 MiB on an Apple M5 laptop. `python3 verify.py` then checks every released reading against `data/numbers.csv` in 0.05 s, and `REPRODUCE.md` lists both outputs line by line.

The code is part of this submission, one MIT repository: `estimator.py`, #336537 byte for byte, with its SHA-256 and the grader's package versions in `manifest.json`; the research code under `lab/`; the Table 5 ablation under `ablation/`; and `data/numbers.csv`, which marks every numbered value of this report as measured or derived and rebuilds every table and Figures 1–5. `REPRODUCE.md` and Appendix A.7 give each command, the hardware it needs and the value it must print, and Table A4 maps each claim to its script and check; the frozen checks of Table 2b and of Table 3 up to width 192 rerun on a laptop CPU in about 40 minutes with `bash report/data/frozen_s16/run.sh`, and `verify.py` rescores the saved predictions of the frozen 1024 column. `THIRD_PARTY.md` names the origin of each file.

9. Use of LLMs

The author designed the methodology, propositions, and experimental architecture, led the research, approved every submission, and reviewed all experimental results and this report. LLM coding assistants were used for code implementation, cloud job orchestration, verification scripting, and manuscript editing. Every K=4 code path is verified against ARC's reference implementation to 1e-15 at n = 40 (Section 2), and `verify.py` checks all 1,935 numbers in `data/numbers.csv` and recomputes each MSE from saved predictions. The author takes full responsibility for the accuracy and completeness of this report and code.

References and provenance

Every link was opened and checked against the claim it supports on 15 to 26 October 2026.

- W. Wu, V. Lecomte, M. Winer, G. Robinson, J. Hilton, P. Christiano (ARC), *Estimating the expected output of wide random MLPs more efficiently than sampling*, arXiv:2605.05179v2, 2026, and ARC's reference implementation at commit 93d091a4.
- 504aldo, V29 repository at commit 1a4083f and its findings log: F29 and F42 (Table 0, Table 2b).
- Phase 2 write-ups: kai_leal_gonzalez, *The (3,1) slice of κ_4 in cumulant propagation*, with code, logs and preregistration; il.miscusi, #336317; Artyom_Mazur, #333534; DD_team, #335813; Nurlanggs, #336225; hyojun_kwon, #335413; 504aldo, V29, #330093; subsequent write-ups porting, testing or citing this work: huang_chung_yi, #337436; pranay212, #337389; dogus_ozel, #337205; yeongjunyoo, #336955; michael_anderson7, #337578; abhimanyu_singh, #337475.
- hyojun_kwon's Phase 1 write-up: ARC's K=4 at a 61.5 GB memory wall at 256 × 16 (finding 1).
- Puffi's Phase 1 algorithmic submission (SSC closure form, step 5); acco-cyber's V34 and V37 sources at commit 42459cd (koushik_rudra's dead-unit pruning and cost blocks, step 2); James Henry's Phase 1 replication; Marius Binner's Phase 1 report; Keenan Pepper's Phase 2 corpus of 14,048 networks at revision cc3405a (networks 0–2,047 train the corrector, Section 4.2); ARC's deduction–projection post; LordProcciuto's e30_a engine (MIT, #335981).

Appendix A. One layer, its shapes, and what each knob costs

A.1 One layer of full K=4

Algorithm A1 runs one layer l of full K=4 at width n, from the state that enters it to the two kernels it stores.

The births of the last layer never propagate and are not stored, so a network holds 120 GB of kernels at the end. Steps 3 and 6 are the n^4 arithmetic of the full chain: the transported kernel $Q_g B_g Q_g^\top$ and the birth sums are formed densely at every layer.

A.2 Which numbers come from n^4 arithmetic and which from n^3

Table A1. The computations behind each number, at n = 1024.

computation	arithmetic per layer	state	numbers
full K=4: the n^4 chain	two n^4 products per group to read $\kappa 4$; n^4 to move $\kappa 3$	one n^3 kernel a group	Tables 1–3, Figures 1–2, 3a
K=4 subsets in the n^4 chain	the same code with groups or pairings dropped	as above	Table 1 col. 4, Table 3
n^3 forms of single pieces	matmuls on n×n legs: star, same-side reads of any core, young feedback (22–26 a layer)	n×n legs, an n×k basis	Sections 2.4–2.6, compact rows of Table 7
compact complete chain, counted	the n^3 forms, C·$\kappa 3$ reads at 63b units a pair, crossings at 4r, dense non-tree first reads at 3n units (n^4)	legs, bases, a $k_3{}^3$ core	Table 7, Figure 4b
late-start chain	chain B to layer s_0, then the n^4 chain	as the n^4 chain	Figure 3b, Section 5.2, dense rows of Table 7

computation	arithmetic per layer	state	numbers
chain B and its branches	factorized K=3 with Strassen products, n^3	5.3 GiB a network	Tables 4–6
non-tree matrix pairs	r n^3 per read	r n^2 a birth	Table 8

For the cheap n^3 subset of the second row, the n^3 forms reproduce every slice to 1e-18 at n = 128 and 256.

A.3 How C·κ3 and non-tree kernels are built

C·κ3 births at layer b, in the order the compact form runs them.

1. Take every live third-cumulant source s with legs $(X^{(s)}, P^{(s)}, Y^{(s)})$, each n×$r_s$.
2. Form its core $Z^{(s)} = 12\,\mathrm{diag}(w_2)\,X^{(s)}$ (n×r_s). The birth is Sym(A⊗P⊗$P^{(s)}$⊗$Y^{(s)}$) through $Z^{(s)}$, with A = diag(w_1)C_{off} and P = I on one side (n×n) and $P^{(s)}$, $Y^{(s)}$ on the other (n×r_s).
3. At each later layer, transport the four legs by that layer's Q: U = QA, V = QP, S = Q$P^{(s)}$, T = Q$Y^{(s)}$.
4. Same-side reads: $d_A = U \circ V$ (n×n), $d_B = S \circ T$ (n×r_s), $a_Z = d_A Z$, $b_Z = d_B Z^\top$; then (4,) = rowsum($a_Z \circ d_B$), (3,1) = $[(a_Z \circ S)T^\top + (a_Z \circ T)S^\top + (b_Z \circ U)V^\top + (b_Z \circ V)U^\top]/4$, (2,2) = sym($a_Z d_B^\top$)/6, and (2,1,1) as the three-leg terms (a_Z, S, T)/6 and (b_Z, U, V)/6, which the K=3 chain takes as legs. Summed over chain B's live sources, r_s is 9,216·b columns, and a transport with a read is 7 wide matmuls: 63b units (Table 7).
5. Crossed reads: one SVD $Z \approx FG^\top$ at rank r, kept for the birth's life; at each read, for q ≤ r, x_q = sym(U diag(F_q) $V^\top$) and y_q = sym(S diag(G_q) $T^\top$); (2,2) gains $\frac{2}{3} x_q \circ y_q$ and (2,1,1) the pair (x_q, y_q): 4r matmuls.

Non-tree births at layer b (Section 5.4).

1. T = the new (2,1,1) slice minus its tree terms (Section 2.6), n×n×n in the research chain.
2. Kernel $B_{abr} = 6\,\mathrm{sym}_{ab} T_{rab}$, unfolded as F (n×$n^2$) with rows on the repeated index r.
3. Direction: start from chain B's normalized mean pre-activation and take one power step, $u \leftarrow F(F^\top u)$, normalized.
4. Pair: V = $F^\top$u as a symmetric n×n matrix; after transport Q, A = Q diag(u) $Q^\top$ and D = Q V $Q^\top$.
5. Reads: (4,) = diag A ∘ diag D; (3,1) row i = $[(\mathrm{diag}\,A)_i D_{i,:} + (\mathrm{diag}\,D)_i A_{i,:}]/2$; (2,2) = sym(diag A diag $D^\top$)/6; (2,1,1) = the pairs (diag A, D) and (diag D, A), each /6. Rank r repeats steps 3–5 on r directions.

Appendix A.6 runs the same births from chain B's factors, without forming T or F.

Star births read at the output (Table 6, third row). Chain B represents κ4 radially, which leaves out star births with four distinct indices. Their contribution to the (4,) slice at layer l, for one birth at layer b < l, is

$$q = 4\,\mathrm{rowsum}\big[(P\,\mathrm{diag}\,w_3) \circ (U^{\circ 3} - 3\,U \circ S_2 + 2\,S_3)\big], \quad U = VC^\top,\ S_k = V^{\circ k}(C^{\circ k})^\top,\ V = P\,\mathrm{diag}\,w_1,$$

with C the zero-diagonal covariance and w_1, w_3 the ReLU coefficients at layer b, P the transport from b to l along chain B's trajectory, and powers taken entrywise; the three terms are the inclusion–exclusion that keeps the three leaves distinct. With q_l summed over the four most recent births, the mean shift is carried forward linearly, $d_l = \Phi_l \circ (W_l d_{l-1}) + q_l \circ (w_4/24 + D_3\,w_7/144)$, the Edgeworth terms in κ4 and κ3κ4, with Φ_l the slopes, D_3 chain B's diagonal κ3 and w_4, w_7 the ReLU's fourth- and seventh-derivative expectations; the output is chain B's plus d_{16}. It meters 356 units (chain B: 170).

Algorithm A1. One layer of full K=4. Gold lines give what the compact form of Sections 2.4–2.6 does in the same step; the right column gives each step's arithmetic, tagged where it is n^4, and the shape of each read slice.

	in	mean μ (n), covariance C $(n\times n)$, third cumulant κ_3 $(n\times n\times n$, dense in ARC's reference chain$)$; the fourth cumulant as its repeated-index slices plus source groups g, each a birth leg x_g $(n\times n$, or $I)$, a birth kernel B_g $(n\times n\times n$, in pinned host memory$)$ and one cumulative transport Q_g $(n\times n)$	
1	linear	$\mu \leftarrow W\mu$; $C \leftarrow WCW^\top$; $\kappa_3 \leftarrow \kappa_3 \times_1 W \times_2 W \times_3 W$; $Q_g \leftarrow WQ_g$	$n^4 + n^3$ per g
		▶ *compact* every leg $(n\times r) \leftarrow W$·leg; old legs share one basis Q $(n\times k)$, moved once	
2	ReLU	$s = \operatorname{diag} C$; $w_1, \dots, w_7$ $(n$ each$)$, the ReLU's Hermite coefficients at (μ, s)	
3	read κ_4	**for** every group g:	two n^4 per g
		$X = Q_g x_g$, $Y = Q_g$ $(n\times n)$; $\tilde{B}_g = Q_g B_g Q_g^\top$ $(n\times n\times n)$; $d_A = X \odot Y$, $d_B = \operatorname{diag}_{12} \tilde{B}_g$ $(n\times n)$	
		$\kappa_4^{(4)}$ += rowsum$(d_A \odot d_B)$	n
		$\kappa_4^{(3,1)}$ += $\tilde{B}_g \cdot d_A/2 + \big((X \odot d_B)Y^\top + (Y \odot d_B)X^\top\big)/4$	$n\times n$
		$\kappa_4^{(2,2)}$ += $(d_A d_B^\top + d_B d_A^\top)/6$ + crossed pairing	$n\times n$
		$\kappa_4^{(2,1,1)}$ += same-side and crossed blocks, 128 rows at a time	$n\times n\times n$
		▶ *compact* kernel kept as an $(n\times r)$ core Z; a transport and a same-side read are 7 matmuls of $n\times r$; crossed reads use Z's rank-r SVD, $4r$ matmuls; star kernels are diagonal, 3 matmuls	
4	term table	ARC's term sums over $(\mu, C, \kappa_3$, the four κ_4 slices, $w) \to \mu', C', \kappa_3'$ and the new repeated-index slices; the $(2,1,1)$ slice of step 3 enters κ_3'	
		▶ *compact* feedback older than m_{0f} layers in one $(k_3\times k_3\times k_3)$ core; young tree terms from two $(n\times n)$ graph-term factors, 22–26 matmuls; non-tree terms by a dense first read	
5	carry	$Q_g \leftarrow \operatorname{diag}(w_1)\, Q_g$ for every existing group	
6	two-leg birth	$x = \operatorname{diag}(w_1)\, C_{\text{off}}$ $(n\times n)$; $B_{jkl} = 12$ path $+ 4$ star $+ 12$ $C\cdot\kappa_3$ terms, each a product of C_{off} slices or of κ_3 with its w factors $(n\times n\times n)$; $Q = I$	n^4
		▶ *compact* star → legs (x, I), diagonal kernel $4w_3$; path → core $12\operatorname{diag}(w_2)\, C_{\text{off}} \operatorname{diag}(w_2)$ $(n\times n)$; $C\cdot\kappa_3 \to$ core $12\operatorname{diag}(w_2)\, X^{(s)}$ $(n\times r_s)$ for every live K=3 source s	
7	slice birth	$B = \kappa_4'^{(4)} + 4\kappa_4'^{(3,1)} + 3\kappa_4'^{(2,2)} + 6\kappa_4'^{(2,1,1)}$ of the new slices, on the diagonal $(n\times n\times n)$; $Q = I$	
		▶ *compact* $\kappa_4'^{(4)}$, $\kappa_4'^{(3,1)}$ diagonal; $\kappa_4'^{(2,2)}$ as core $3\kappa_4'^{(2,2)}$ $(n\times n)$; $\kappa_4'^{(2,1,1)}$ on its K=3 legs, core $6x$	
8	store	both new kernels to host memory, at $n = 1024$ in float32	4 GiB each

A.4 What each compression knob costs

Table A2. Compression knobs of the K=4 chain. Units, wall time and peak memory (GiB): the compact K=4 source machinery alone (flopscope, 16 layers, n = 1024, κ3 legs n wide, 2 CPU threads, C·κ3 crossing at rank 1; in brackets at rank 4). Raw: the same knob in the n⁴ research chain (portable set), official networks 0 and 1, against their exact 3.03e-9 and 2.77e-9; feedback rows against the chain with uncompressed feedback, 3.90e-9. Each error is read at the output, after every later layer has carried it.

piece	knob	units [rank 4]	wall	GiB	raw, nets 0 / 1	÷ exact
OLD SOURCES	basis k = 128 from age 2	648 [1,044]	80 s	0.44	3.90e-9 / 3.97e-9	1.29 / 1.43
	k = 256 from age 2	1,052 [1,594]	84 s	0.48	3.04e-9 / 2.99e-9	1.00 / 1.08
	k = 384 from age 2	1,541 [2,279]	87 s	0.51	3.05e-9 / 2.81e-9	1.01 / 1.02
	k = 128 from age 3	678 [1,148]	95 s	0.44	3.38e-9 / 3.18e-9	1.12 / 1.15
	k = 128 from age 4	705 [1,243]	109 s	0.45	3.19e-9 / 2.91e-9	1.05 / 1.05
	k = 384 from age 4	1,365 [2,150]	115 s	0.51	3.02e-9 / 2.77e-9	1.00 / 1.00
	batched, k = 128 from age 2	566	73 s		3.90e-9	1.29
	batched, k = 128 from age 1	522	59 s		5.03e-9	1.66
	batched, k = 112 at birth	432	43 s		1.13e-8	3.72
C·κ3 CROSSING	SVD rank 1 / 16	4r matmuls a read			6.70e-9 / 5.20e-9 (net 0)	2.21 / 1.72
STAR CROSSING	rank 1–16 / exact	3 matmuls a read, exact			5.57e-8 – 4.58e-8 (net 0)	18 – 15
FEEDBACK	three-axis core at birth, k_3 = 64–256	0.42 / 3.0 a layer at k_3 = 64 / 128			5.10–5.57e-9 (net 0)	1.31–1.43
	exact 1 / 2 layers, then $k_3 = 128$	16,363 / 30,564 young (CP); 386 tree			4.17e-9 / 4.02e-9 (net 0)	1.07 / 1.03
NON-TREE BIRTHS	matrix pairs, rank 1 / 2 (n = 64)	11.6 + 6.0 / 13.7 + 12.0 a read			gain kept 79–84% / 101%	
	joint-space birth, k = 4 / 64	9.1 / 46.4 a birth			birth kept 17% / 71%	

The old-source rows show the age law of Section 3 as a price: at k = 128 the loss falls from 1.29× to 1.05× as entry moves from age 2 to 4 for 57 more units, and k = 256 from age 2 removes it on network 0 for 404 more. The scored chain's own

knobs (shared basis 352, residual and feedback ranks, thin legs at rank 4, young-atom cut at 99.5%, active-set threshold $w_1 < 0.014$) carry their cost, memory and adj in Tables 4 and 5.

Which reads need exactness. The four output channels close the recursion together: removing one from the portable set (2.98e-9) multiplies its error by 1,148 to 714,978, from 3.42e-6 for (2,1,1)→K3 to 2.13e-3 for (4,)→mean and variance, a measure of closure rather than of shares. The star crossed pairing has the diagonal kernel w_3 and costs 3 matmuls exact, while rank 1–16 truncations are 18× to 15× worse. The C·κ3 crossed kernel is close to full rank: at rank 16 the worst layer keeps 11% of its energy and the output error is 5.20e-9 (1.72×), at rank 1 6.70e-9 (2.21×). The non-tree term $C_{jk}\,\kappa_{3,ijk}$, which the factorized K=3 chains of the V29 lineage leave out, is worth 1.56× (4.72e-9 against 3.03e-9). Tree and non-tree feedback terms travel together: at n = 128, layer 13, two non-tree terms of size 61.6 and 63.7 cancel to a total T of 0.9, so keeping the tree terms exact and projecting the rest at birth (6.48–7.23e-9) is no better than projecting all of it (6.46e-9); the triangle loop term costs 6% (3.14e-9 against 2.98e-9). Unless stated, these readings are on official network 0 at width 1024, where the portable set gives 3.03e-9 (2.90e-9 over networks 0–1, 2.98e-9 with the triangle term at rank 32).

Table 7b. Separate end-to-end **dense** run at $s_0 = 4$, same official network 0. Chain B's prefix was metered; the run's 203,289 tensor operations were priced by flopscope shape/dtype rules. It prices the dense research code, which reads every source as a dense tensor at every layer; Table 7a counts the compact form of the same computation.

dense run component	priced units
chain B to layer 4, κ3 expanded once, seeds, transport	46,777
reads of every young and old source	1,701,100
shared basis: solve, project, rebuild n×n	33,795
C·κ3 loops, repeated-index corrections, coupling, rest	2,280
end-to-end run total	**1,783,952**

The dense run of Table 7b in detail. On one 96 GB GPU it takes 444 s after chain B's state is loaded, at 56.7 GiB of GPU and 182 GiB of host memory. Over the chain, C·κ3 birth loops cost 35 units, repeated-index corrections 72, the coupling of the K=4 state to chain B's lower moments at each nonlinearity 1,303, and the rest of the fourth-order nonlinearity with the shared birth construction 870: 2,280 in all, 3.8% of the compact lower bound. The shared basis costs 33,795 because this code rebuilds each projected source as an n×n matrix, and reads take 95.4% of the 1.78 million because it reads every source as a dense tensor at every layer, the n^4 arithmetic (Section 2.3) that the compact rows replace.

A.5 How the corrector was trained

Table A3. The corrector in #336537 (Section 4.2), as trained; brackets give #335351's corrector where it differs. e_l: the root mean square of ε_l over the training neurons.

item	setting
CHAIN	chain B (#335301) as submitted, corrector off, with dense products in place of Strassen (equal to float32 rounding); every layer of every network keeps chain B's mean, the ReLU slopes Φ_l and the input columns below
NETWORKS	training: public full split 0–999 and Keenan Pepper's corpus networks 0–2,047 [full split only]; public mini 0–99: chose #335351's configuration (state width, inputs, activation, epochs), which #336537 keeps, and are read after each epoch
TARGET	d_l = the true mean at layer l minus chain B's mean, in float64 with the weights rebuilt from each seed
	the true means: ARC's `all_layer_means` on the full split, the corpus's 10^8-sample means on the corpus (noise floor 9.0e-10 at the output, by its card)
	the error layer l adds, $\varepsilon_l = d_l - \Phi_l \circ (W_l d_{l-1})$, $\varepsilon_0 = d_0$; learned as ε_l/e_l, with e_l from 2.9e-5 at layer 1 to 6.5e-5 at layer 15 [1.7e-5 to 6.5e-5]
INPUTS PER NEURON	35 of layer l: pre-activation mean, variance and α, the ReLU's density, slope and Hermite coefficients, chain B's mean and cumulant diagonals after the nonlinearity, the regeneration λ
	of the same 35, row and column sums (plain, absolute, squared) of the off-diagonal covariance and of the (2,1), (2,2) and (1,1) slices. 16: eight of them at layer l−1 through W_l and $W_l \circ W_l$. 64: the 32-dimensional state through W_l and $W_l \circ W_l$
	3: $a = W_l u_{l-1}/e_l$, $\Phi_l \circ a$ and $(W_l \circ W_l)(u_{l-1}/e_{l-1})^{\circ 2}$. 16: the layer's one-hot
NORMALIZATION	each of the 51 chain columns: sign(x)·ln(1 + 10^3|x|), then centred and scaled per layer and column by its mean and standard deviation over the 3,048 × 1,024 training neurons [1,000 × 1,024]
	the state through tanh; the 3 correction terms by e_l
NETWORK	134 → 128 → 128 → 128 → 33, GELU; output 1 is g_l/e_l, outputs 2–33 the next state; 54,561 parameters
LOSS	mean over layers 0–15 of the squared error of g_l/e_l against ε_l/e_l; each network runs through all 16 layers with u and the state carried, gradients through the whole recurrence
OPTIMIZER	Adam, one-cycle schedule to 2e-3 with 15% warm-up, 4 networks a batch, 30 epochs, 22,860 steps [7,500], seed 0, TF32 matmuls, weights in float16 on the GPU
SELECTION	after each epoch, the corrected raw over chain B's raw, summed over the mini networks; the lowest is kept: epoch 30, the last, 0.873 (0.947 after epoch 1) [epoch 30, 0.882 (0.973)]
SHIPPED	standardization tables, per-layer biases with the one-hot folded in and e_l folded into the weights: 58,376 float32 numbers in base64 inside `estimator.py`, equal bit for bit to the trained export; outside them the file is #335351's line for line

item	setting
	its NumPy forward matches PyTorch to 4.8e-3 of the largest correction [1.4e-6]
	the shipped file on mini 0, on a laptop: raw 1.93e-8, against 1.95e-8 on the training path
COST	trajectories of 3,148 networks at 35–177 s each, about 100 single-thread workers on one 2 × A40 machine; training 37 min on one 48 GB A40; 2.0 h from launch to the embedded file at $0.85 an hour
	[1,100 trajectories in 34 min on 96 workers, 168 s and at most 2.0 GB each; training 116 s on one 96 GB GPU; 48 min from boot]; on the grader C/B goes from 0.166 to 0.169
RESULT	mini: 0.873 of chain B's raw, lower on 100 of 100 networks (0.76–0.98 each) [0.882, 0.77–0.99]; it leaves 0.50 of the new error's energy at layer 1 and 0.94 at layer 15 [0.95]
	grader: 0.887, lower on 50 of 50 (0.81–0.97) [0.896, 0.81–0.99]; seeds 1 and 2 [mini 0.881 and 0.882, 100 of 100 each]

Table A3b. Table 5's switches with #335351's corrector held fixed, mini 0–7 only. Each ratio is corrected switch-off ÷ corrected chain B on those eight networks: the corrector stays as trained while the chain beneath it changes, so the ratio includes the shift of the state the corrector reads.

switched off	fixed-corrector adj ÷ corrected B [95%]	worse /8
nothing: corrected B	1	
active-set trimming, with V34's and V37's row cuts	3.72 [3.36, 4.12]	8
young-atom pruning	1.97 [1.73, 2.20]	8
representation	1.03 [1.01, 1.05]	7
all three, with V34's and V37's row cuts	**4.11** [3.65, 4.68]	**8**

Network 6 sensitivity. On network 6, a layer-15 outlier dominates the mean ratio's sensitivity, with the top 20 neurons accounting for 79.6% of the squared-error difference between chain B and the active-off arm. Across the ablation runs, all paired arms are compared against each machine's own control, and paired medians and win rates summarize the 50-network panel alongside the aggregate means (Table 5).

A.6 Non-tree births on chain B as one chain

Algorithm A2. Non-tree births on chain B as one chain, in the order a layer runs them, shapes at $n = 1024$; the births enter at layer b. Blue lines give the measurement that holds each step; the right column gives the step's units at $n = 1024$.

	`in`	chain B at layer b: mean μ (n), covariance C $(n\times n)$, ReLU coefficients $w_1, \dots, w_7$ (n each); κ_3 as 6 old sources with factors $(352\times n)$ in the shared basis Q_c $(n\times 352$, 3 of them in a 224-column tier inside it) and 7–8 young sources with their own legs $(n\times r_s,\ r_s \le n)$	
1	`sources`	young: their legs; old: as they are, or projected into span(Q_c) and summed into one symmetric core G $(352\times 352\times 352)$ ▶ *measured* $n = 512$, layer 13: G with the 7 young sources against all 13 original ones, relative squared error 8.7e-12 at birth, at most 3.1e-11 at the first reads	
2	`kernel`	F $(n\times n^2)$: $B_{abr} = 6\,\mathrm{sym}_{ab} T_{rab}$, T the new $(2,1,1)$ slice minus its tree terms, unfolded on r; never formed: $F^\top u$ $(n\times n)$ and $F\Omega$ $(n\times k)$ are contractions of the source factors, C and w ▶ *measured* factor births against dense ones: 4e-16	
3	`space`	weighted covariance of every source's factors $(n\times n)$; a randomized range finder keeps k directions Q_k $(n\times k)$; every source projected into Q_k and summed into one core $(k\times k\times k)$ ▶ *measured* 9.1 units a birth at $k = 4$, keeping 17% of the birth; 46.4 at $k = 64$, 71%; the same birth with every source as it is, 762 units	9.1–46.4
4	`direction`	u = chain B's mean pre-activation, normalized; $u \leftarrow F(F^\top u)$, normalized, once, on F itself or on the k-space core ▶ *measured* $n = 64$, every source: 99.80–99.99% of the kernel's best rank-1 energy; the mean alone, 29–55%	
5	`pair`	$V = F^\top u$ $(n\times n$, symmetric); the birth is (u, V)	
6	`each later layer` l	Q = transport from b to l $(n\times n)$; $A = Q\,\mathrm{diag}(u)\,Q^\top$, $D = QVQ^\top$; the $(4,)$, $(3,1)$, $(2,2)$ and $(2,1,1)$ reads from (A, D) by matmuls, as in A.3 step 5 ▶ *measured* metered at $n \le 64$ and carried along $n^{2.98}$	6.0 a read
7	`state`	(A, D) stays a K=4 source next to the tree births and is read at every layer through 15; the K=4 state feeds chain B's mean, covariance and κ_3 at each nonlinearity (Section 5.2) ▶ *measured* $n = 1024$, networks 0–1, slice births exact: 0.55× and 0.66× of chain B's raw, tree births alone 1.07× and 1.11×; $n = 64$, three networks: rank-1 pairs kept for life hold 79–84% of the gain; extrapolated from layer 4: 557 units, and 1,969 for the tree births read with them	557 + 1,969
	`or: read once`	the same (A, D) read only at $b + 1$ and handed to chain B's κ_4 and κ_3 consumers ▶ *measured* $n = 1024$, networks 0–1: 0.982–1.030 of chain B's raw	

Steps 1, 2 and 4–6 ran together on chain B at 1024, every source as it is and nothing compressed, each birth read once: the last line, −0.7% to −1.8% on network 0 and +0.1% to +3.0% on network 1. Step 3 puts steps 1–2 into one k^3 core at 9.1–46.4 units against 762, keeping 17–71% of the birth. Step 7 holds the gain: carried for life next to the tree births, the same births take chain B's raw to 0.55× and 0.66×. This chain run at 1024 for every birth's whole life, at 557 + 1,969 units extrapolated, is the non-tree target of Section 5.5.

A.7 Reproducing each claim

The commands below regenerate every table, figure and ablation from the repository root:

```
python3 -m venv .venv && .venv/bin/python -m pip install -r requirements.txt
source .venv/bin/activate
python3 smoke.py                                                    # scored estimator
python3 make_tables.py all && python3 make_figures.py               # tables and figures
mkdir -p outputs
(cd outputs && python3 ../lab/cand_ssc/fit_knobs.py ../lab/cand_ssc/jc/pert.json inrun \
    ../lab/cand_ssc/data/j1_a_final.npz ../lab/cand_ssc/data/j1_b_final.npz \
    --ridge 0.01 --cfgdir lab/cand_ssc/jc)
python3 ablation/summarize.py && python3 ablation/summarize.py --panel 50
python3 verify.py                                                  # saved-result checks
python3 ablation/run.py --arm b --learner plain --width 128 --indices 0    # one Table 5 cell
python3 ablation/run.py --arm col_off --learner plain --indices 0-7        # column cut, 6 GiB
python3 ablation/matrix.py --width 1024 --indices 0-7               # 80 cells, 16 GiB

python3 -m pip install -r requirements-research.txt
R=refs/mlp_cumulant_propagation
git clone https://github.com/alignment-research-center/mlp_cumulant_propagation.git $R
git -C $R checkout 93d091a4c26c042bfffa28f2e76a81bc0aba94bb && python3 -m pip install -e $R
export AW_DATA=data PYTHONPATH=lab:lab/cand_k4x:lab/cand_k4x/s11:lab/cand_k4x/s12
export PYTHONPATH=$PYTHONPATH:lab/cand_k4x/annealed:tools:$R/src
bash report/data/frozen_s16/run.sh                                  # Tables 2b and 3, CPU
python3 lab/k4full.py public 0,1,2,3,4 cuda f32                     # Table 1, full K=4, GPU
python3 lab/ref1024.py 0,1,2,3,4 k3a cuda f32                       # Table 1, aug. K=3, GPU
# Table 3: AW_SPEC names the birth types kept (grammar in the docstring)
AW_SPEC='star,slice4+slice31,!slice22+slice211' python3 lab/k4full.py public 0,1 cuda f32
AW_SPEC='path+star,slice4+slice31+slice22+slice211' python3 lab/k4full.py public 5 cuda f32
python3 lab/k4n3.py check --n 8 --rank 4 --device cpu               # n^3 slices against n^4
```

Table A4. Claims, how to regenerate them, and the check behind each.

claim	number	script	check
full K=4 at 1024	1.72e-9	`lab/k4full.py`	the reference to 1e-15 at n = 40
K=3 aug at 1024	2.03e-8	`lab/ref1024.py`, `lab/depth_s1/ref_depth.py`	the reference; per layer, final to 0.04%
width, depth, n/L	Fig. 1	`lab/kscan.py`, `report/layer_noise.py`	20,000 resamples; noise = official `avg_variance`
ablations	Table 3	`lab/k4full.py` with `AW_SPEC`	grammar in the docstring
frozen checks	Tables 2b, 3	`report/data/frozen_s16/run.sh`	saved predictions rescored; spec full equals the reference at n = 128
frozen at 1024	Table 3	`lab/k4full.py` with `AW_SPEC`, networks 5–7	twelve saved predictions rescored against the truth
star weights	Section 3	`report/data/c45/build.py`, `make_figures.py`	60 saved final layers rescored against `sources.json`
n^3 form	1e-18	`lab/k4n3.py check`	against the n^4 chain, float64
young feedback	4.2e-15	`lab/k4n3_young_direct_chain.py`	16 layers against the reference chain, float64
oracle patches	Fig. 3a	`lab/var_k4_channel.py`	20 frozen-script reruns: zero difference
late start	Fig. 3b	`lab/cand_k4x/b_suffix.py`	$s_0 = 0$ is the reference at n = 16; tree reads to 5.6e-16
frozen retest	Table 6	`lab/cand_k4x/r3/run_one.py`	held-out networks 4–11; 64 MSEs recomputed
ledger; the run	Table 7	`lab/cand_k4x/meter_b_birth.py`	layer 6 metered; all 203,289 operations priced
non-tree births	Table 8	`lab/cand_k4x/{annealed,s11,s12}/`	Wick mean $\leq$ 2.4 s.e.; pairs to 2e-16, factor births 4e-16
bottleneck reduction	Tables 9–10, Fig. 5	`lab/reduce_s1/select_run.py`, `lab/reduce_s1/make_readout.py`	64 raws recomputed from their saved predictions; the ledger reproduces Table 7a's 60,549, 47,287 and 33,611
first compression	Table 11, Fig. 5	`lab/reduce_s2/run.py`, `lab/reduce_s2/make_readout.py`	8 raws recomputed from saved predictions; per-layer family counts checked; full rank restores the dense kernel same-box control within 0.02%
metered chain	Table 12, Fig. 5	`lab/e2e_s1/chain.py --meter`, `lab/e2e_s1/write_tables.py`	seven small-width checks in float64, all within 2.8e-12; billing shim 329/329 each layer within 0.14% of the GPU chain
eight write-ups, same grader	Tables 0, 4b	`make_figures.py` (`nb_*` keys)	ten official submission pages saved 19 October, against that day's leaderboard
scored entries	Table 4	the submitted estimators	grader receipts of all eight steps, network by network
corrector	Table A3	`lab/corpus_lt/`, `lab/lt2_train.py`, `lab/lt2_embed.py`	#336537's embedded weights equal the export bit for bit; re-embedding it rebuilds the file byte for byte
own pieces off	Table 5	`lab/ablate_s1/eval.py`, `lab/ablate_s2/`, `lab/ablate_s3/`	50 paired networks per no-corrector arm; new-machine trace checked against its own control
		`ablation/run.py --arm col_off`	column cut: 100 final layers rescored

A.8 Tables drawn in Figures 1 and 4

Figure 1a draws Tables 2 and 2b and Figure 4c draws Table 6; the tables give every value with its interval.

Table 2. The gap across width at depth 16, and across shapes at n/L = 64: mean raw final-layer MSE over networks with its 95% Student t interval; for the ratio at n/L = 64, a 95% interval from 20,000 paired network resamples. Truth: 5×10^7-sample Monte Carlo at 256–512 × 16, 1.5×10^9 samples at 256 × 4, 1.2×10^9 at 512 × 8, the official 10^9-sample truth at 1024 × 16.

n × L	networks	K=3 aug	K=4 full	K=3 aug ÷ K=4	truth noise
256 × 16	seeds 11–13	1.45e-6 [8.1e-7, 2.1e-6]	3.80e-7 [7.8e-8, 6.8e-7]	3.8×	1.6e-9
384 × 16	seeds 21–23	4.22e-7 [3.4e-7, 5.1e-7]	1.04e-7 [5.2e-8, 1.6e-7]	4.0×	1.7e-9
512 × 16	seeds 41–43	1.94e-7 [1.5e-7, 2.4e-7]	2.88e-8 [2.5e-8, 3.3e-8]	6.7×	1.9e-9
1024 × 16	official 0–4	2.03e-8 [1.9e-8, 2.2e-8]	1.72e-9 [1.6e-9, 1.8e-9]	11.8× [11.3, 12.4]	8.4e-11
fit	MSE $\propto n^{-\alpha}$	$\alpha = 3.08$ [2.95, 3.17]	$\alpha = 3.95$ [3.68, 4.08]	gap 0.87 [0.74, 0.96]	
256 × 4	seeds 14–16	5.67e-8 [3.5e-8, 7.9e-8]	5.54e-9 [3.5e-9, 7.5e-9]	10.2× [8.5, 12.3]	2.3e-10
512 × 8	seeds 44–46	4.00e-8 [2.4e-8, 5.6e-8]	3.48e-9 [2.6e-9, 4.3e-9]	11.5× [10.8, 12.1]	1.5e-10

Table 6. K=4 content added to chain B: raw final MSE over chain B's on the same network and machine, and units per network (metered for the star births read at the output; otherwise chain B's metered 170 plus the counted matmuls of the added piece). Net 0: where settings were chosen; nets 0–3: ratio of mean raws. Held-out split: evaluated on networks 4–11 of the public full split with all settings fixed on networks 0–3; mean and median of the eight ratios, and networks of 8 that improve.

K=4 content added to chain B	read	net 0	nets 0–3	frozen	median	better	units
(4,) slice (diagonal κ4), last 4 layers	output	0.999					170
C·κ3 births, age 4	output	1.000					409
star births, four distinct indices, age 4	output	0.987	0.997	0.990	0.992	6	356
star births, age 4	fed back	0.967		0.990	0.987	6	559
(4,) and (3,1) slices, age 4	fed back	0.978		0.991	0.989	**7**	590
(2,2) slice at rank 8, age 3	fed back	0.990		1.000	1.000	4	897
star age 4, slices age 1, (2,2) at rank 2	fed back	0.965		0.986	0.979	5	919
C·κ3 births at rank 4, age 2	fed back	0.996		1.000	0.998	5	975
star births + (4,) and (3,1) slices, age 4	fed back	**0.952**	**0.990**	**0.982**	**0.976**	5	976
full K=4, reference chain (Table 1)	whole chain	0.073	0.076				n^4

Table 2b. The n/L = 8 cells of F29, three new networks each, frozen before running. Ratios of final-layer MSE per network; antithetic Monte Carlo truth of 1.7×10^7 samples, whose noise is below 0.2% of every error.

n × L	K=3 aug ÷ K=4 full	K=4 aug ÷ K=3 aug	F29, K=4 aug ÷ K=3 aug
64 × 8	2.1×, 3.7×, 2.7×	1.6×, 2.3×, 4.0×	20×
96 × 12	2.5×, 2.0×, 3.0×	above 4 GB in one process	17×
128 × 16	4.2×, 2.0×, 3.2× (earlier order scan)		out of memory

A.9 Proofs of Propositions 1 and 2

Notation. For a matrix $T \in \mathbb{R}^{n\times n}$ and a four-index tensor S, $(T^{\otimes 4}S)_{ijkl} = \sum_{i'j'k'l'} T_{ii'}T_{jj'}T_{kk'}T_{ll'}S_{i'j'k'l'}$; Sym averages a tensor over the 24 orders of its indices; for vectors u, v and a matrix B, $u \otimes v \otimes B$ has entries $u_i v_j B_{kl}$.

Proposition 1. *Lemma 1.* $T^{\otimes 4}$ commutes with Sym: permuting the indices of S and then applying T to each index gives the same tensor as applying T first and permuting after, because one matrix acts on every index; averaging over the 24 orders keeps the equality. *Lemma 2.* $T^{\otimes 4}(u \otimes v \otimes B) = (Tu) \otimes (Tv) \otimes (TBT^\top)$, since $\sum_{i'j'k'l'} T_{ii'}u_{i'}\, T_{jj'}v_{j'}\, T_{kk'}T_{ll'}B_{k'l'} = (Tu)_i\,(Tv)_j\,(TBT^\top)_{kl}$. *Induction.* At birth the claim holds with $Q = I$. If it holds after layer $l-1$ with Q', then layer l maps the source to $T_l^{\otimes 4}S_{l-1}$; by Lemma 1, linearity and Lemma 2,

$$T_l^{\otimes 4}S_{l-1} = \mathrm{Sym}\Big(\sum_r (T_lQ'x)_{:r} \otimes (T_lQ'y)_{:r} \otimes (T_lQ')B_r(T_lQ')^\top\Big),$$

which is the claim with $Q = T_lQ'$. *Storage.* ARC's form keeps, per group, $A_r = x_{:r} \otimes y_{:r}$ as a dense $n \times n$ slice for each r and the transported B_r, $2n^2R$ numbers rewritten at every layer. The stored form keeps x, y ($2nR$) and B (n^2R) fixed after birth and rewrites only Q (n^2), one $n \times n \times n$ product T_lQ' per layer. The two-leg groups of Section 2.1 have this form with

$y = I$ (for C·κ3 births, $B_a = \sum_c Z_{ac} P^{(s)}_{:c} Y^{(s)\top}_{:c}$), and slice groups with legs Q and their slice kernel. At $n = R = 1024$ in float32 each birth kernel is 4 GiB, and the groups born at layers 1–15 hold the 120 GB of Table 1; the births of the last layer never propagate. □

Proposition 2. *(a)* By Taylor's theorem there is a point ξ on the segment from $\tilde{s}$ to s with $F(\tilde{s}) = F(s) - \langle g, \delta\rangle + \frac{1}{2}\delta^\top \nabla^2 F(\xi)\,\delta$, so $F(s) - F(\tilde{s}) = \langle g, \delta\rangle + O(\|\delta\|^2)$ with the constant set by the largest Hessian norm on the segment; the Cauchy–Schwarz inequality gives $|\langle g, \delta\rangle| \le \|g\|\,\|\delta\|$, with equality exactly when δ is parallel to g. For the final layer, take $F = F_i$, the estimated mean of output neuron i, with readout vector g_i: the final-layer MSE depends on the truncation only through the readouts $F_i(\tilde{s}) - F_i(s) = -\langle g_i, \delta\rangle + O(\|\delta\|^2)$. *(b)* Take the 2×2 state $s = \mathrm{diag}(1, \varepsilon)$ with the Frobenius inner product and the readout $F(s) = s_{22}$, so $g = e_2 e_2^\top$ and (a) holds with no remainder. The best rank-one approximation of s in the Frobenius norm (Eckart–Young) is $\mathrm{diag}(1, 0)$: its truncation $\delta = \mathrm{diag}(0, \varepsilon)$ has $\|\delta\|/\|s\| = \varepsilon/\sqrt{1+\varepsilon^2} < \varepsilon$ and $\langle g, \delta\rangle = \varepsilon = \langle g, s\rangle$, the whole readout. The rank-one projection onto $e_2 e_2^\top$ keeps $\mathrm{diag}(0, \varepsilon)$: its truncation $\mathrm{diag}(1, 0)$ has relative norm $1/\sqrt{1+\varepsilon^2}$ and readout 0. As ε is arbitrary, no function of $\|\delta\|/\|s\|$ that vanishes at 0 bounds the relative readout loss; padding with zeros places the same pair in a state of any dimension, and by (a) it carries over to a nonlinear F whose gradient is g. □

What the propositions price. Proposition 1 establishes exact factored storage across all 16 layers (matching ARC's reference to 1e-15 at n = 40), and Proposition 2 links local truncation directly to downstream readout observability (Table 3c). An energy basis is a norm-optimal truncation and loses the readout of row 17 of Section 6 (84% of the norm, 51×); an exact (3,1) slice sets $\delta = 0$ on one block and leaves the others, recovering at most 10.0% of the K=4 log gain; a read weight taken through the next layer alone is a local stand-in for g and ranks the non-tree births last although they carry 1.95× (Section 5.5).