WhestBench Phase 2 write-up: learned corrections, carriers and where the error lives

Oct 25, 2026 · Matthew Ryan Davis

Summary

Our main finding: with the learned corrections switched off (calibrated λ and K31_C stay on, as defined in Finding 1), two very different cumulant carriers have similar average accuracy on a six-net block. Our V29-lineage carrier (~438 GF) and LordProcciuto's e30_a (~329 GF) differ by 3% in mean final-layer MSE. On these nets most of an estimator's edge comes from the learned corrections trained on top.

- **Learning is the edge.** Our learned stack takes our carrier ×0.74 in MSE; e30_a's own mean table takes it ×0.90, and our refit of that table (Q71p) ×0.86.
- **Structure that buys no accuracy is pure cost.** Refitting the cheaper carrier's own mean table on 52 nets, with our structural settings, gave −14% score on untouched nets against our best own-carrier entry. Our per-neuron hook, blend and readout did not move with it. Graded result: 3.084e-9, vs 3.596e-9 for our best own-carrier entry; adding young-atom pruning gave our final entry, 3.062e-9.
- **The corrections have a physical reading.** κ3's correction is a ×1.5–1.9 rescale of the rank-limited feedback channel. κ4's correction is consistent with a missing contribution in the closure: our hypothesis is that the λ regeneration stands in for a κ3⊗κ3 → κ4 birth.
- **Where the remaining error lives.** Off-diagonal covariance error drives the late-layer variance error. κ21 and κ31 births account for most of what is newly created each layer, and exact κ31 is worth ×0.87–0.88.
- **The shape is right; the moments are wrong.** Given true slice moments, four cumulants reproduce E[ReLU] to the sampling floor. Using the carrier's slice mean with every other moment true gives an MSE 75% as large as Q71p's, so most of the error travels through the inherited mean (a counterfactual ratio, not an additive split). The carrier's κ3 decorrelates with depth (corr ≈ 0.7 by L15) in ways none of our per-unit corrections predicted.
- **No tested replacement for the old κ3 tier passed.** Closures, templates, finite memory, low-rank, vector ⊗ matrix and matrix-pair forms cost ×1.02–×119 in final MSE; the near-misses keep most of the state. An even, sign-gauged coherent fit of the κ3 slices still

leaves 26–29% of their energy, spread over hundreds of directions: the remainder has the degrees of freedom of the weight rows. A 64-probe sketch puts 90% of the output's response to readout errors in 32–52 directions per layer (Finding 5).

- **Negative results with numbers.** Sampling corrections, low-rank compression, kink-only carriers, sign-split weights, teacher-forced and end-to-end correction fitting, a carried-state corrector, a structural port of the feedback rescale, truncation extrapolation, stacked small gains, symmetric-billing rewrites and deeper Strassen each fail, and we show by how much.
- **The same state predicts far more than the mean.** With no fitting, unit variance and firing probability are 91–912× cheaper than Monte Carlo at equal accuracy across 61 networks, and linear readouts 111–692× at the last layer. Tails need a per-unit κ3 calibration (useless for the mean) and a saddlepoint formula; with them, last-layer 4σ tails are 22× cheaper than sampling. Training a network collapses its activation cloud into 2–5 directions and breaks the carrier from mid-depth; staged oracle experiments motivate a mixture carrying the full cumulant step per piece, though an affordable implementation remains unvalidated (Finding 6).

Designated submission: #337276 (Q71q-a), score 3.062e-9: final-layer MSE 2.07e-8 at 326 GF (14.8% of the budget), 50/50 public MLPs graded, no failures; selected on the Submissions page on Oct 22. It is Q71p (#336865, 3.084e-9; first graded as #336474) plus young-atom pruning following Susupro's rule (eps 0.005). Built on e30_a (MIT); our changes are listed in Submissions and attribution.

Setup

Each task is a 16-layer ReLU MLP of width 1024 with He-initialised weights and standard normal inputs. The estimator predicts every neuron's expected activation, and it is scored on the final layer.

The score has a floor, and it changes what counts. Score = final-layer MSE × max(0.1, FLOPs/B) with B = 2^41, so the multiplier floors at 219.9 GF. Below that, a cost cut is worth nothing; above it, a cut is worth only the MSE it does not lose. Q71p bills 335.8 GF (multiplier 0.152). The public top entries bill 230–290 GF, and the leading adjusted score is consistent with raw ≈1.3e-8 at the floor. Reaching the floor from Q71p needs −34.5%. The old κ3 tier is 31% of the bill, so the floor cannot be reached from the old tier even by dropping it; most of the cut would have to come from the young tier (D21 readout 27.6%, dense transport 25.5%). Every cost result below should be read against that arithmetic.

- **Score** = final-layer MSE × max(0.1, FLOPs / 2^41), with 2^41 ≈ 2.2e12, computed per net and then averaged. FLOPs are billed by flopscope: dense matmul at 2MKN − MN, elementwise ops at 1 per element, with no sparse discount.
- **Grader limits per MLP:** 120 s wall, 0.4 s residual (client-side) time, 8 GB memory, 5 s setup. Each worker runs its MLPs in one persistent process.
- **Truth:** the organisers' 1e9-sample means for the 1000-net full split (`all_layer_means`). Weights for any net are rebuilt bit-exactly from `mlp_seed` with `derive_seed_streams` .

Protocol (local MSE is final-layer and paired by net; most runs are warm-process, and some confirmation runs were cold):

Role	Nets	Used for
Training	0–51 (moment targets on 0–39, 48–51)	fitting learned corrections
Development	52–59	choosing between variants
Confirmation	60–67, then 100–107	one-shot check of the chosen variant at first use; later experiments reuse 56–67 as development data
Public grader	50 public nets	graded score

Provenance by finding. Where a number came from, so each can be traced to a run.

Finding	Carrier	Learning	Nets	Fit objective	Process	Metric	Raw results
Submission (Finding 1)	e30_a with our refitted 87-feature mean table (R_FB = 4, old-tier ranks 416/224); packaged as Q71p	on	fit 0–51; development 52–59; confirmation 60–67, then 100–107	mean table fitted teacher-forced to per-layer MC means	confirmation blocks cold	final-layer MSE paired by net; score per net then averaged; public grade #336865, flopscope 0.12.3	Codex outputs, Q71p confirmation records; public leaderboard
Finding 2 (channel decomposition)	our V29-line carrier (R_FB = 8), not Q71p	off for the decomposition; stack on for the ×0.74	fits on 0–39; 40–47 held out	per-layer MC κ3 / κ4 regressions, then final MSE	warm	linearised and live final MSE	Codex outputs, Q41–Q47 series
Finding 3 (error anatomy)	our V29-line carrier	on (after learning)	40 (41 as check); held out for this carrier, in-sample for Q71p's table	none (diagnostic)	warm	per-layer moment errors vs MC, 4.19M samples, cross-half products	Codex outputs, Q13–Q16 series
Finding 4 (slice attribution)	Q71p, live, state dumped	on	52–55	none (oracle swaps of exact slices)	warm	linearised final MSE	Codex outputs, Q71 ablation series

Finding	Carrier	Learning	Nets	Fit objective	Process	Metric	Raw results
Finding 5 (old-tier replacements, J spectrum)	exact plain K3 reference chain (lean), not Q71p	off (no learned terms exist)	40 (52 as check)	closure and template coefficients fitted to local readout least squares; one check fitted to linearised final MSE	warm; J on Modal	final-layer MSE vs the exact chain	outputs/sensitivity_I, outputs/spectrum_J, outputs/objective_gap; cost map outputs/sensitivity_G/costmap_summary.json
Negative results table	as stated per row	as stated per row	52–59 unless stated	per row	warm	paired final-layer MSE	Codex comms, by task letter

Screening uses paired per-net ratios against the current reference. A variant is packaged only if it beats the reference on development and on an untouched confirmation block.

Submissions and attribution

We built two lines: our own estimator on 504aldo's public V29, and a port of our learning method onto LordProcciuto's public e30_a. Both bases are MIT; their notices ship in our files.

Entry	Submission	Public score	Final MSE	GF	Base
Q71q-a (designated, selected)	#337276	3.062e-9	2.07e-8	326	Q71p + young-atom pruning (Susupro's rule, eps 0.005)
Q71p (designated until Oct 22)	#336865	3.084e-9	2.03e-8	334	e30_a (LordProcciuto)
Q70m	#336273	3.596e-9	1.80e-8	438	V29 (504aldo)
Q70h	#336114	3.679e-9	1.84e-8	438	V29
U1v	#335918	3.861e-9	—	—	V29

Ours on the V29 line:

- the per-neuron learned hook on variance, D3 and D4 (Q28 / U1);
- the Q58 κ4 feed blend;
- the Q50 final readout;
- the K31_C κ3·T/v term in the (3,1) slice;
- a teacher-forced refit of an 87-feature mean table in e30_a's form (QM);
- engineering: depth-5 Strassen with a Winograd leaf, a pooled-view cache, upkeep routing and setup pre-warm.

Ours on the e30_a line (Q71p):

- the learning-off structural screen and the choice it made: K31_C = 1, λ scale 1.0, λ tilt 0;
- a teacher-forced refit of e30_a's 87-feature mean table on full-split nets 0–51, which replaces the shipped table;
- the attempts that did not pay and are reported below: the Q58 blend (×0.995 MSE) and pre-wick hooks.

Credits. V29, the lean K3 chain and the λ law: 504aldo (whest-p2-cumulant-k3, MIT). e30_a, including its fused mean-correction evaluator, compaction and fixed-basis old tier: LordProcciuto (#335981, MIT). The cumulant-propagation reference and term tables: ARC (mlp_cumulant_propagation, MIT). PK2: Hisernberg (V44). We also used ideas from koushik_rudra's and kai_leal_gonzalez's forum posts. Finding 5 compares against public results by Nurlanggs (18252), Artyom_Mazur (18249) and Susupro (18254). The young-atom pruning rule in Q71q-a is Susupro's (18254, logxio/fourth-cumulant); the terminal-corrector architecture we tested is hyojun_kwon's (18255); the alternative-basis Strassen recurrence is Schwartz and Vaknin's (SIAM J. Matrix Anal. Appl., 2023). Code was written largely by LLM agents (Anthropic's Claude and OpenAI's Codex) under the team member's direction.

Provenance. Every graded number traces to a saved archive and source file, identified by SHA-256 (first 8 hex digits here; full hashes, builders and per-number scripts in the accompanying provenance file). Q71p #336865: archive 74cfc5b2, estimator 67671d21, mean table fitted on nets 0–51 (Q71_meanbeta_0-51.json). Q71q #337252 (failed, time limit): archive a1ad3193. Q71q-a #337276: archive 693941de, estimator 6705ce11, mean table refit on the pruned chain on nets 0–51, graded 3.062069e-9 (saved API record; the 50/50 count and 326 GF come from run telemetry, and the Oct 22 selection from the author's account page); a rebuild script produces a validated archive with byte-identical estimator source (archive metadata and the archive hash differ) from an isolated source directory (not yet from a fresh checkout with a fresh dependency install). Q71q-b #337318 (failed, undiagnosed): archive f2cb1aae. Our e30_a base matches LordProcciuto's published file (SHA 6ddd97a9); its header cites upstream V29 commit 1a4083f as its base, but we have not established which commit its final term table comes from, so we cite the file hash rather than the commit. Local comparisons use experiment-specific blocks (52–59, 56–67, 60–67, 100–107, stated with each result). A confirmation block was untouched only at its first use; later results on 56–67 reuse those nets as development data, so no local block is a fresh holdout. The graded public split is the only clean held-out measurement.

Reproducing Q71q-a. One command (rebuild_qa.py, in the accompanying files) copies the saved estimator, licence and change log into a fresh isolated directory, builds and validates the archive, checks the estimator bytes against SHA 6705ce11…, audits imports, and runs a small prediction in an isolated interpreter (python -I -S) with the pinned dependencies; smoke_qa.py repeats the prediction check alone. This verifies an isolated source rebuild, not a fresh checkout with a fresh dependency install and not a full contest evaluation.

Finding 1: without learning, the two carriers are similarly accurate

We switched off every learned component in both estimators, measured both on the same nets, and only then trained corrections. Learning off, e30_a is 3% more accurate than our carrier at 25% fewer FLOPs.

Carrier	GF	MSE, learning off	MSE, learning on	Gain from learning
Ours (V29 line, Q70m structure)	~438	2.46e-8	1.83e-8	×0.74
e30_a as shipped	~329	2.38e-8	2.14e-8	×0.90 (its mean table)
e30_a + our structure choices, refit (Q71p)	~330	2.40e-8	2.07e-8	×0.86

Local results on development nets 52–57 (learning-off columns) and 52–59 (learning-on). The two MSE columns use different net sets, so the gain column is a ratio of block means, not a paired estimate. "Learning off" removes the hook, blend, readout and mean tables but keeps the calibrated closure (fitted λ law and K31_C).

- **Per-net ratio of the two learning-off carriers:** 0.87–1.09. So on this block the two carriers have similar average accuracy; we did not set an equivalence margin.
- **Our ~110 GF of extra structure buys nothing on its own.** It only paid on our line because our corrections were fitted to it.
- **Learning-off structural screens on e30_a** (nets 52–55):
  - K31_C = 1 gives ×0.991;
  - λ scale 1.0 and tilt 0 each give ×0.987;
  - stacked, ×0.977 on 8 nets;

    - no FLOP cut tested held its accuracy (old-tier rank 352: ×1.029 score; dead cutoff 2.5: ×1.40).
- **After learning, the λ preference washes out:** shipped λ with a refit table gave ×0.996. This is why we treat λ as a learning knob, not structure.

Result. Q71p (e30_a + stack + refit table) against Q70m (our carrier + full stack), on untouched nets 100–107: MSE ×1.14, but FLOPs ×0.76, so score ×0.858, better on 8 of 8 nets. The grader agreed: 3.084e-9 vs 3.596e-9.

Protocol lesson. Comparing structures with their own learned corrections attached hides this. Each correction is fitted to its carrier, so a structural change judged with stale corrections looks worse, and one judged after refitting conflates structure with refit noise. Measuring learning off first, then training once, made the comparison clean and cheap: one evaluation per variant instead of a 6–7 hour refit.

Finding 2: what the learned corrections actually do

We decomposed the learned corrections against Monte Carlo truth (up to 2.4e7 samples on held-out nets). They do two physical things: rescale an under-sized κ3 channel, and supply a κ4 birth the closure lacks. They leave per-neuron variance alone.

Variance: nothing to fix per neuron. The carrier's per-neuron variance is within about 0.03% of MC on average over all layers and neurons, and the learned variance correction is about 0. Finding 3's 0.11–0.17% is a different cut of the same quantity: the relative variance error at layers 10–15 only, where it is largest. The remaining variance error is cross-neuron (Finding 3).

κ3: the feedback channel is about half-sized. The carrier's D3 splits exactly into four channels (path, moment, feedback, feed; residual 5e-8).

- The learned correction leaves path and moment alone (gain ±0.01).
- It multiplies the feedback channel by about ×1.5–1.9, flat over layers 4–15.
- A regression of (MC κ3 − D3) on the four channels, with no learning involved, gives the same answer: feedback coefficient +0.65 to +1.02, path and moment within ±0.03.
- In our V29-line carrier (Q70m) the feedback channel is a rank-8 range-finder projection of D21 at birth (Q71p uses rank 4), so it is truncated, not wrong in form.
- Rescaling the feedback legs structurally hurts (×1.5: +2.3% MSE; ×1.9: +5.9%), because D21 is scaled too. Raising the rank helps but costs more (rank 32: −4.2% MSE at +50 GF). The diagonal-only rescale is the cheap form that worked on the V29 line. Ported

structurally to Q71p it fails: ×1.48 with learning off on net 55, ×1.33 on nets 56–59 after a full mean-table refit with learning on (Negative results).

κ4: born by the wrong mechanism.

- The learned κ4 correction removes about 60% of the MC κ4 error at L15. Two terms reproduce it: $D4_err \approx -0.15 \cdot D4 + 0.9 \cdot D3^2/v$ (correlation 0.93–0.97 with the learned correction from L7 on). The $D3^2/v$ coefficient (0.9) is of the order of the quadratic-map value 4/3, though 33% below it.
- Fitted directly to MC κ4, our blend is $d4 \leftarrow 0.61 \cdot d4 + 0.57 \cdot acc + 0.35 \cdot D3^2/v + 0.56 \cdot chain + 0.73 \cdot star$. It uses the regenerated κ4 at about half weight and supplies the rest from κ3 products, from the same layer, one layer back, and accumulated.
- **Reading:** our hypothesis is that the closure has no κ3⊗κ3 → κ4 term. We have not derived that diagram and its coefficient. The λ regeneration, calibrated to total κ4, inflates to stand in for it. The blend moves that content onto the hypothesized mechanism.

The global scaling knobs we tested are already near their optima. Scaling each category of the state update (mean, covariance, κ3, κ4 inputs) finds optima within 0.6% of 1, and a combined gain below 0.2%. The covariance's κ3 input is extremely sensitive: a 1% coherent error costs +25–29% MSE.

Transfer test (the prediction). If the blend fixes a flaw of our λ law, it should do little on e30_a, whose λ and mean table were tuned jointly.

- Fitted on e30_a, the blend is $1.05 \cdot g4 + 0.96 \cdot D3^2/v + 0.02 \cdot chain + 0.04 \cdot star$: essentially e30_a's own $D3^2/v$ term.
- Live MSE ×0.995 on development and ×0.994 on untouched nets, versus −12 to −20% on our carrier.

Finding 3: the remaining error comes from pair-cumulant births

After learning, more than 90% of our final error is in the propagated state, not the readout. We traced it back one link at a time against Monte Carlo (4.19M samples per run, independent halves, cross-half products so MC noise does not count). Net 40, with net 41 as a check. The shares below are linearised, one link at a time, and items 1–4 trace the same error backwards, so they are nested, not additive.

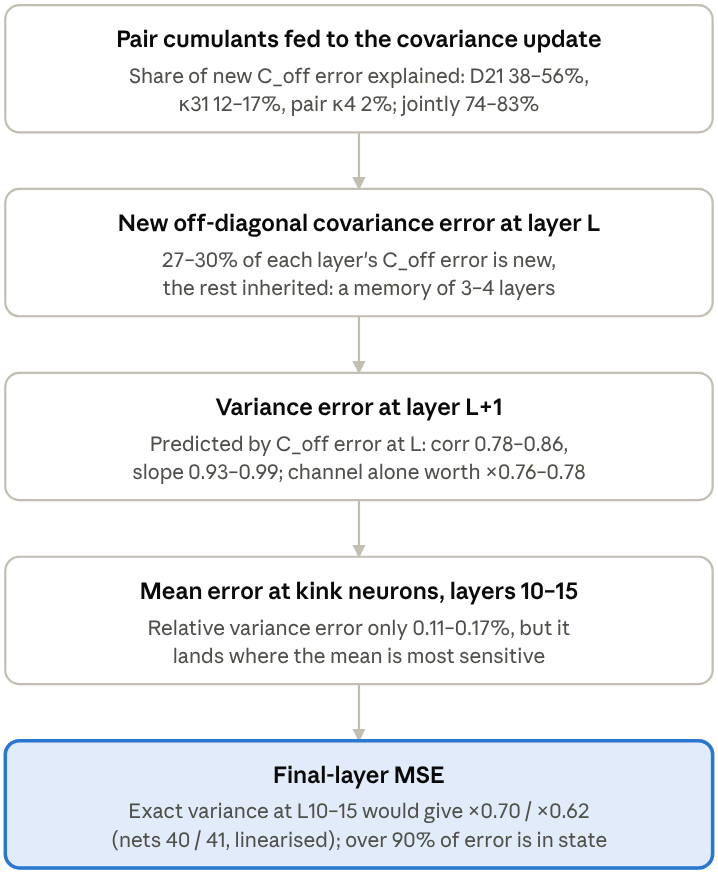
1. **Variance error at L10–15 drives about a third of the final error.** Making variance exact there would give final MSE ×0.70 / ×0.62 (nets 40 / 41, linearised). The relative variance error is only 1.1–1.7e-3, but it lands at kink neurons where the mean is most sensitive.

2. **That variance error comes from the previous layer's off-diagonal covariance.** The C_off error at L predicts the variance error at L+1 with correlation 0.78–0.86 and slope 0.93–0.99. This channel alone is worth ×0.76–0.78 of final MSE. No per-neuron or per-pair-statistic feature explains it (0–1%), which is consistent with our per-neuron learned corrections not reaching it.
3. **About 27–30% of each layer's C_off error is newly created, and the rest is inherited.** That gives the error a memory of 3–4 layers.
4. **The new error is inherited from the pair cumulants fed to the covariance update:**

Input to the covariance map	Share of new C_off error explained	Input accuracy vs MC
Pair κ3, κ(i,i,j) (D21)	38% → 56% (rising with depth)	72–83% of signal
κ31, κ(i,i,i,j)	12–17%	only 47–52% of signal
Pair κ4, κ(i,i,j,j)	2%	93–96%
All three jointly	74% → 83%	
Old-tier rank, term scales, missing Hermite order	⩽ 2% each	

The shares in this table are regression attributions among correlated inputs. They do not partition the new error exactly, and they do not by themselves set the final-MSE gain that fixing any one input would give.

The final error traces back through one pathway



error-budget pathway · net 40, learning on, linearised shares

Read top down: each box is the measured link from the one above it, on net 40 with learning on. The shares are linearised and nested, not additive. Finding 4 adds the last link on Q71p: most of the final mean error is inherited through later layers rather than created at them.

Oracle injections of the exact slice. We replaced our κ31 with MC truth at chosen layers and measured final MSE.

- Exact κ31 at L9–13: ×0.881; at L2–13 (Wiener-shrunk where MC is noisy): ×0.873.
- Exact D21 alone: ×1.006 with learning on, and worse with learning off (×1.054 on Q71p with the mean table off, ×1.024 on the exact plain K3 chain; net 40, layers 9–13). The closure's own D21 error is compensating other errors in the chain, not being absorbed by the learned stack. Exact κ31 plus D21: ×0.830.
- Single layers are uneven: L10 ×0.948, L12 ×0.957, L9 ×0.980, L13 ×0.983, L11 ×0.992.
- This independently matches kai_leal_gonzalez's estimate (exact slice −18% on V29). Like their two-term law, the cheap fixes recover only a small part of it.

What cheap κ31 rebuilds reach.

- Carried κ4 star, chain and pair births raise the explained share of true κ31 from 0.47–0.52 to 0.54–0.61. That gives ×0.963 final MSE at ≥ +25% FLOPs, a net loss.
- The sketchable cheap form gives ×0.982 for +10–20 GF. Early layers alone are worth ≤ 2% even with exact truth.

Where the frontier is. Exact pair κ3 and κ31 would remove about 75–80% of the newborn C_off error, roughly ×0.80–0.85 of final MSE. That needs better pair-cumulant births: a carried κ31, better κ21 birth terms. Not compression, not map form, not per-neuron learning. il.miscusi's carried κ4 with scale-mode extraction (×0.84 vs e30_a at 2× FLOPs) is the public attempt closest to this, with the caveats in open directions. Our earlier observation fits it: the off-diagonal error has a mean-direction rank-2 component $\mu y^T + y\mu^T$ explaining 15–41% of its energy. That is the shape a common scale mode leaves.

Finding 4: the shape is right; the moments are wrong, and most of the error is inherited

Given the true moments of each pre-activation slice, a four-cumulant Gram–Charlier formula reproduces E[ReLU] to the sampling floor at every layer. Q71p's error therefore comes from predicting those moments. Using the carrier's slice mean with every other moment true gives an MSE 75% as large as Q71p's live MSE, so most of the error travels through the mean inherited from earlier layers. This is a counterfactual ratio, not an additive attribution.

Setup. Nets 52–54, 524,288 N(0, I) samples each.

- The true slice mean is computed from the supplied 1e9-sample dataset means at layer l−1, times W_l (an estimate, not an exact expectation).
- Variance, κ3 and κ4 come from the samples, with split-half noise subtracted.
- Q71p ran live with its per-layer state dumped.

Shape test (true moments in, MSE of E[ReLU] over units):

Layer	Gaussian (mean, var)	+ skew, kurtosis (Gram–Charlier)	Sampling floor
1	1.9e-7	1.2e-8	1.2e-8
4	3.0e-7	3.8e-9	3.9e-9
8	2.8e-7	1.6e-9	1.6e-9

Layer	Gaussian (mean, var)	+ skew, kurtosis (Gram–Charlier)	Sampling floor
15	1.9e-7	5.6e-10	4.6e-10

Slices are near-Gaussian: |skew| 0.003 → 0.09 and |excess kurtosis| 0.005 → 0.06, both growing with depth. Even so, the Gaussian-only shape error is about 10× our whole final MSE. With four cumulants the excess over the sampling floor is about 1e-10 at L15, a point estimate without an uncertainty; the Edgeworth κ_3^2 term adds nothing measurable.

Attribution of Q71p's live error (MSE vs dataset truth; each column swaps one carrier moment into the true set):

Layer	Q71p, after mean table	Via slice mean (inherited)	Via κ3	Via κ4
4	2.1e-9	9.6e-10	1.0e-9	5.0e-10
8	7.3e-9	4.7e-9	1.2e-9	7.7e-10
12	1.4e-8	1.0e-8	1.3e-9	1.0e-9
15	2.0e-8	1.5e-8	1.8e-9	1.3e-9

- **The variance term is noise-limited.** It is at most ~2e-9 against a split-half floor of 5e-9 at L15.
- **κ3 degrades with depth.** Correlation of the carrier's κ3 with the true κ3 falls from 0.93–0.99 at L4 to 0.65–0.73 at L15, at about half the true magnitude. κ4 correlation is 0.45–0.77.
- **Per-unit fixes don't recover it.** A per-layer κ3 or κ4 gain fitted on one net worsens held-out E[ReLU]. A ridge on the carrier's per-unit features predicts κ3 better overall (corr 0.95), but weighted by $\partial E/\partial \kappa_3$ it is flat. The κ3 error that matters was not predicted by the tested per-unit features; the dropped three-way cross terms are a candidate, not an identified cause.
- **The cloud is not compressible at our precision.** The post-ReLU covariance's effective dimension falls from 238 (L3) to 47 (L15). Across a layer pair, about 470–540 directions pass with canonical correlation > 0.5. Matching the leaders' cost with a basis would need about 300 directions for the youngest κ3 content; it needs 512–640 at ages 2–3, and full width at ages 0–1.

Implication. The error is a slow build-up of small per-layer injections from mispredicted skew and tails, compounded through the weights. None of our learned per-unit corrections, including the linearised end-to-end fit, reached it. That is consistent with every hook result in Negative results.

Finding 5: no tested replacement for the old κ3 tier passed our gates

The old κ3 tier holds sources four or more layers old in a shared basis (ranks 416 and 224 in Q71p). It is about 31% of Q71p′s FLOPs on net 52; young transport and readout are about 53%. No cheaper replacement we tested passed our gates. The two closest (×1.021 and ×1.056) keep most of the state, so they would need FLOP savings above 2% and 5% to break even, which we did not measure. The rest cost ×2 or more. These results rule out the schemes tested, not approximate carry as a class.

All runs use the exact plain K3 reference chain (not Q71p) on net 40, with net 52 as a check where stated. The closure and template coefficients were fitted to a local readout objective, not to final MSE. One check fitted them to the linearised final MSE instead (one replacement at layer 8, two scale-mode templates, multipliers fitted on net 40 and frozen for net 52): ×2.331 against ×2.332 for the readout fit on net 40, and ×1.984 against ×1.985 on net 52, with the linear prediction within 0.1% of the actual error. For that case the objective was not the problem; we did not repeat it at other layers or with other templates. The chain has no learned corrections, so learning is off. Ratios are final-layer MSE against the exact chain (3.27e-8 on net 40).

Replacement for the old tier	Final MSE vs exact	What it shows
Drop it	×26.3	The tier carries most of κ3′s value
Mean-field closure from exactly carried aggregates (unfitted / per-layer oracle fit)	×9.40 / ×9.40	The 5–15% of old κ3 that aggregates miss is what matters
Same closure at the last layer only	×2.08	One approximated layer already doubles the error
Templates built from current state, oracle coefficients (nets 40 / 52)	×11.2 / ×12.8	The tested template dictionary with its local least-squares objective failed

Replacement for the old tier	Final MSE vs exact	What it shows
Exact for ages 4–10, closure for age ⩾ 11 (nets 40 / 52)	×1.19 / ×1.06	Sources 11+ layers old still matter
Exact top 75% of atoms by weight, closure for the rest	×1.056	Even the weak atoms must stay
Low-rank residual after removing the scale mode (rank 2 / 4 / 8)	×5.17 / ×4.99 / ×4.51	The residual is high-rank
Vector ⊗ matrix (mode-1) form, 16 / 64 / 256 terms	×7.19 / ×1.85 / ×1.021	About 256 terms are needed, the size of the basis already carried
No κ3 at all; κ2 and κ4 kept (nets 40 / 52)	×108 / ×119	The even cumulants cannot stand in for the odd one

Why the tested closures fail. Our reading, supported by the residuals below rather than proved: the tested closures are built from aggregate state, and that state misses a network-specific part of old κ3.

- **What the closure gets right.** The closure captures the scale mode well: R^2 is 0.85 on the D21 slice and 0.94 on D3 at deep layers.
- **What it misses.** The final mean is very sensitive to the network-specific, random-sign remainder, which the closure cannot see.
- **Related result.** Artyom_Mazur reports that odd orders have no coherent level on random networks. That is consistent with our reading but does not prove it (18249).

The engine already corrects itself as far as it can. A readout error reaches the output through two channels: the consumer channel, which reads it directly, and the feedback channel, through the new sources it seeds.

- **The channels add linearly.** 28 separately perturbed parts sum to the full effect with cosine 0.99.
- **They largely cancel.** The two channels point in opposite directions, with cosine −0.75. The consumer channel alone gives ×19.2 and feedback alone ×14.2, against ×9.40 together.
- **The cancellation is already near its optimum.** Rescaling the feedback per layer gains 4–10% on the net it was fitted on and nothing on the other net.

The output is sensitive in many directions. Because the final mean responds linearly to readout errors, we measured that response directly. If it had only a few large directions, a compression would only need to be accurate in those.

- **Measurement.** For each layer, we injected 64 random 1% perturbations into the D3 or D21 readout. We then took the spectrum of the 1024 × 64 matrix of output changes: 1,536 runs on Modal, $2.62.
- **Result.** The response needs 32–52 directions per layer to hold 90% of its energy, and 55–63 of the 64 to hold 99% (table below, 64 probes per cell).
  - Net 52 matches net 40 to within two directions at every layer.
  - These are 64-probe sketch estimates, not guaranteed bounds on the true energy ranks. Where we checked, r90 rose from 58 to 64 probes.
  - The normalised squared response (energy) to D3 errors grows 12.5× from layer 4 to layer 14 on net 40 and 16.9× on net 52; the D21 energy peaks at layers 8–12.
- **Error size alone is not the problem.** If the response stays linear at that size, random 10–15% errors at layer 4 would cost only +5% MSE. We checked linearity only at 1–2% perturbations, on net 40 at layer 10. The closures' errors are about that size, yet cost ×9, because they are structured and line up with the directions the output responds to.

Layer	D3 directions for 90% / 99%, net 40	net 52	D21 directions for 90% / 99%, net 40	net 52	D3 response energy, net 40
4	32 / 55	32 / 55	43 / 61	43 / 61	7.9e-5
6	35 / 57	34 / 56	46 / 62	46 / 62	1.8e-4
8	36 / 58	38 / 58	48 / 62	48 / 62	3.7e-4
10	40 / 59	40 / 60	50 / 63	50 / 63	5.7e-4
12	44 / 61	44 / 61	52 / 63	51 / 63	7.8e-4
14	49 / 62	48 / 62	50 / 63	50 / 63	9.9e-4

A smaller exact state would save numbers; we found no way to make it save compute. An exact old tier is a linear time-varying system: sources move forward each layer, new ones are added, and the readout is linear. This holds with the trajectory frozen, so that the

system is linear. Minimal-realisation theory then bounds the observable state after layer l from above by what the remaining readouts can see: about (layers remaining) × n^2, shrinking to n at the output. Building that smaller state in our implementation still takes one element-wise product per (source, later layer) pair. We have not shown that no other algorithm could build it more cheaply.

How this fits outside work.

- **Numerical linear algebra.** The old-κ3 update, $K_{l+1} = (M_l \otimes M_l \otimes M_l)\, K_l + B_l$, is a time-varying tensor Stein (discrete Lyapunov) recursion. Results on low-rank approximability for such equations rest on structure, such as decaying singular values (Kressner and Uschmajew). We see no such structure in random Gaussian transports with their Wick factors. That fits the ranks above, but it is our reading, not a theorem.
- **Sketching.** Tensor sketches such as Higher-order Count Sketch support approximate tensor operations on the sketch itself. We did not test an evolving sketch of the old tier; the probes above measure output response, not a sketch.
- **ARC.** ARC lists the depth cost of cumulant propagation as an open problem (blog).
- **Other public write-ups.** Other carriers report similar limits. These are the authors' numbers, not checked by us.
  - Nurlanggs finds that the all-distinct κ3 needs rank about 3n/4 and the full depth (18252).
  - Artyom_Mazur reports five κ3 compressions that "died on the meter" because building the compressed object cost what it saved (18249).
  - Susupro (18254) puts it as "state is worth carrying when later layers keep reading it".
- **The last lead.** At fourth order, Susupro found that ranking state by its first-order effect on the output did 3.4–22× worse than their baseline chain. What worked instead was matrix pairs that follow the reads, built by unfolding a birth kernel on its repeated index. We tested the third-order analogue and it fails; see the next paragraph.

The matrix-pair test at third order. Each κ3 source is a sum over neurons q of the birth layer: a pair of legs (P_q, P_q) or (T_q, T_q) times a single leg. We unfolded each source's birth kernel on that repeated index q and kept the top r directions in q, which turns the source into r vector ⊗ matrix pairs. This is exact at full rank (×1.000 sanity check). Learning is off and the net is 40.

Kept per source	Old sources (age ⩾ 4)	Young sources (age 0–3)
1 pair	×21.8	×48.4

Kept per source	Old sources (age ⩾ 4)	Young sources (age 0–3)
4 pairs	×21.3	—
16 pairs	×19.0	×37.0
64 pairs	×7.35	×17.4
256 pairs	×2.65	—

- **The kernels have flat spectra.** At birth, one pair holds 0.3–7% of a path kernel's energy and 1–15% of a fan kernel's. Sixteen pairs hold at most 31% and 77%. Susupro's fourth-order kernels kept 79–84% with one pair.
- **The format only saves compute at about one pair per source.** Each pair costs about 2 n^3 per layer to carry, against about 4 n^3 for a source's current legs and readout. At one pair per source the old tier costs ×21.8, close to dropping it (×26.3).
- **Why it doesn't transfer.** Our third-order birth kernels have flat spectra on this unfolding, while Susupro reports a dominant direction for the fourth-order non-tree kernel.

The sign-gauge test. If the incoherent remainder were partly sign scrambling, an even version of the κ3 slices should be far more predictable from aggregate state. Pre-activation κ3 is odd under flipping one weight row; with s_i the sign of neuron i's pre-activation mean, $D3_i \cdot s_i$ and $D21_{ij} \cdot s_j$ (index i appears twice in $\kappa3_{iij}$ and stays signed) are flip-invariant. We fitted the gauged slices with a binned coherent model on even features only ($|\alpha|$, σ, the gauged covariance and its square, 20×4 bins on the signed index, 10×4 on the gauged one) and compared with the same model using signed bins on both indices. Net 40, layers 9–13, engine slices as fed to the nonlinear map; learning off.

Layer	Scalar scale-mode template, R^2	Gauged even fit, R^2	Signed fit, R^2	Residual energy after the even fit	Residual SVD energy at rank 64 / 256
9	0.38	0.707	0.735	29%	0.82 / 0.97
10	0.36	0.706	0.743	29%	0.84 / 0.98
11	0.33	0.736	0.755	26%	0.84 / 0.98
12	0.31	0.711	0.754	29%	0.85 / 0.98
13	0.29	0.734	0.758	27%	0.85 / 0.98

- The even fit leaves 26–29% of D21′s energy against a bar of about 0.2%, and that remainder is itself spread wide: rank 256 captures 97–98% of it, leaving 2–3%, still far above the 0.2% bar. The D3 fit behaves the same way (R^2 0.66–0.73).
- Imposing the flip symmetry costs 2–4 points against the unconstrained signed fit. The gauge is a reparametrisation of what sign-aware features already express; it uncovers no structure.
- The scalar scale-mode template explains 29–39% of the whole D21 at these layers. The higher R^2 quoted for the closure (0.85 on D21, 0.94 on D3) uses the transported aggregate state X and τ, not C and m alone.

Finding 6: the same state predicts far more than the mean

The contest scores only the final-layer mean, but the carrier already holds each unit′s spread, skew and tails in order to get that mean right. ARC′s stated motivation for white-box estimation is low-probability estimation, so we measured what the same state predicts beyond the mean, scored against Monte Carlo at equal compute. The measure throughout is samples-to-match: the carrier′s per-unit MSE against a 1,048,576-sample truth (the truth′s own sampling floor subtracted), converted to the number of MC samples that would give the same error, divided by the 9,900 samples that cost the same compute as one Q71p pass. Above 1, the carrier is cheaper.

Variance, firing and linear readouts, with no fitting. Across 61 networks (0–51 and 55–63), unit output variance is a median 912× cheaper than sampling at layer 8 and 91× at layer 15; firing probability 666× and 112×; the worst single network is 40× or better at every layer tested. Any linear readout $a \cdot z$ has variance $a^T C a$ under the carried covariance, and on nets 52–55 the layer sum, 256 random directions and the top 20 principal directions all come within 0.12–0.20% of a 1M-sample truth, which is the truth′s own noise: at layer 15 that is 488×, 111× and 692× cheaper than sampling, and 480× to over 2,000× at layers 4–8. Gaussian-only readouts are 3–10× worse on every statistic.

Tails need the κ3 fix that the mean could not use. Uncorrected, 2–4σ exceedance probabilities are 40–320× cheaper than sampling at layer 4 but fall below sampling from layer 12, because the carried κ3 drifts to about half its true value by the last layer (Finding 4). A 15-coefficient ridge calibration of per-unit κ3 on the carrier′s own features (D3, D21 row sums, g4, covariance row sums, variance, mean and the partial ReLU moments) raises the κ3 correlation from 0.68–0.73 to 0.97. It does nothing for the mean, which depends on κ3 only through units at the kink where the regression gains nothing, and it is useless for

the contest; but a tail probability depends on κ3 at every unit, so it pays directly. Fitted on nets 0–51 and tested on nine untouched nets, 55–63 (training on 26 nets gives the same test numbers within 3%):

Layer	κ3 correlation, carrier → calibrated	2σ, carrier → calibrated	3σ	4σ (calibrated + saddlepoint)
4	0.925 → 0.978	28× → 161×	40× → 218×	281× → 1,727×
8	0.831 → 0.978	2.8× → 21×	4.2× → 29×	27× → 152×
12	0.728 → 0.975	0.9× → 7.1×	1.3× → 10×	7.9× → 46×
15	0.684 → 0.965	0.5× → 2.8×	0.8× → 4.3×	4.5× → 22×

The far tail also needs a different tail formula. Gram–Charlier is a truncated series and undershoots at 4σ even with the true moments (layer 15: 4.87e-5 against an MC truth of 5.88e-5). A Lugannani–Rice saddlepoint built from the same four cumulants gives 5.76e-5 with true moments and 5.70e-5 with the carrier's calibrated ones; at 3σ the two formulas agree within 10%, so the gain is specific to far tails. With calibration and the saddlepoint, last-layer 4σ tails are 22× cheaper than sampling, against 10× for calibrated Gram–Charlier and 4.5× uncorrected. We did not validate the formula beyond 4σ: importance sampling for 4.5–6σ truth failed (29 of 32 targets had no 6σ hits), and the plain Monte Carlo references we later obtained to 6σ (nets 55–57, 2^25 samples each) were not scored against it.

Where the method stops: trained networks. We trained a network with exactly the contest's structure (16 bias-free 1024^2 ReLU layers, He initialisation, a 10-way readout outside the stack) as an MNIST classifier to 97.1% and ran the unchanged Q71p engine on four checkpoints with Gaussian input. The fresh checkpoint reproduces everything above (layer-15 mean 287–301×, variance 86×, firing 114×; the κ3 calibration fitted on contest nets transfers, 4σ from 4.8× to 24×). Training breaks the carrier from mid-network: after 100 Adam steps the layer-15 variance error is 5% and the mean loses to sampling at layer 15; after 1 epoch from layer 10; after 3 epochs from layer 8, with NaN at layers 14–15. Neither the mean table nor turning learning off changes this. The cause is the activation cloud

collapsing, not the weights (top singular value 2.83 → 3.1–3.3, stable rank 256 → 192–217, dead units 16% vs 21%). The participation ratio of the pre-activation covariance, the effective number of directions the 1024-unit cloud occupies:

Checkpoint	Layer 2	Layer 4	Layer 8	Layer 12	Layer 15	Top direction, layer 15
Untrained	300	195	114	71	46	10%
100 steps	275	132	21	7	4	41%
1 epoch	234	74	10	5	4	36%
3 epochs	160	33	5	3	2	63%

Mean |correlation| between units at layer 8 goes from 0.07 to 0.32, so the joint cross-unit cumulants the closure drops as $O(1/\sqrt{n})$ are no longer small. This collapse is not il.miscusi's global scale mode: in the untrained net the top direction is that mode (squared cosine with the mean direction 0.89–0.97 from layer 8), but in the trained net it forms at layers 4–8 in unrelated directions (0.02–0.42).

Realistic noise on a trained network: a real image plus static. Gaussian input is not what a classifier sees. We fed the trained network (3 epochs) an MNIST test image plus Gaussian pixel static at four levels and scored against 1M-sample Monte Carlo. Here most units are decided by the image: by layer 8 about 60% never fire and 20% always fire, and only 2–15% are uncertain for σ up to 0.5 (56% at σ = 1.0). Q71p fails in a specific way. Its pre-activation means and variances stay accurate (variance error 0.9% at layer 9), but for the few uncertain units its carried cross-unit κ3 implies standardized skews of 4–40, the cumulant correction to the post-ReLU mean goes wrong, and the error feeds forward to NaN at layer 10. A plain Gaussian closure with the exact pairwise ReLU covariance, which treats deterministic units exactly by construction and costs 68 GF (a fifth of Q71p), runs clean to layer 15:

Static σ (standardized pixels)	Uncertain units, L8	Mean, samples-to-match ÷ cost (L4 / L8 / L15)	Variance rel. error, L15	Firing, L8 / L15
0.1	2%	at MC noise floor / 2,100× / at floor (> 500×)	0.16%	2.5× / 3.3×
0.25	6%	811× / 230× / at floor (> 500×)	0.41%	1.5× / 1.1×

Static σ (standardized pixels)	Uncertain units, L8	Mean, samples-to-match ÷ cost (L4 / L8 / L15)	Variance rel. error, L15	Firing, L8 / L15
0.5	15%	142× / 51× / 115×	1.3%	0.7× / 0.25×
1.0	56%	27× / 2.9× / 1.3×	8.3%	0.08× / 0.06×

The cost denominator is the Gaussian closure's own (about 2,000 forward samples). Up to σ = 0.5 the mean and variance are cheaper than sampling at every layer; at σ = 1.0, static as strong as the image itself, over half the units are uncertain again and the advantage shrinks toward the Gaussian-input regime (mean 1.3× at layer 15, variance error 8%); firing probability is cheaper only at low static, because when nearly every unit is decided, sampling's own error on firing is already tiny. A three-component mixture halves the error but triples the cost. The control runs the other way: on the untrained network with the same input, the Gaussian closure is 31× at layer 15 against Q71p's 114×. Cumulant corrections pay on random-like clouds and hurt on collapsed ones, so a practical carrier for trained networks should switch them off per unit when the carried skew is implausible.

A weights-only training indicator. The participation ratio of the Gaussian-closure covariance, computed from the weights alone in one pass, reproduces the sampled one at every checkpoint (layer 2/4/8/12/15: untrained 303/198/121/83/61 vs sampled 300/195/114/71/46; 100 steps 279/135/22/6.8/4.1 vs 275/132/21/7/4; 1 epoch 238/77/11/5.1/4.0 vs 234/74/10/5/4; 2 epochs 195/42/3.8/1.7/1.5; 3 epochs 165/35/5.6/2.5/2.1). Over the same checkpoints the weight spectra barely move (mean stable rank 257 → 203, top singular value 2.82 → 3.21, with nothing visible at 100 steps, when the layer-8 participation ratio has already fallen from 121 to 22). So the propagated statistics register training long before spectral weight measures do. Whether they track the quality of training (test accuracy, generalization) rather than its amount needed networks of equal training effort and different quality. We built 14 after the contest (label noise, training-set size and input-noise augmentation, two seeds): on the held-out seed, our propagated-statistic measures ranked overfitting far worse than input-gradient size (granulated Kendall τ 0.11 against 0.78), so the answer so far is no.

A tested route to a fix. A trained cloud is random-like once about 20 dominant directions are removed, and the carrier is accurate on random-like clouds, which motivates a mixture split along those directions. Staged oracles (81 pieces, split in three at each of layers 3–6

along their analytic top direction) give a layer-15 variance error of 42% with everything analytic, 35% with the true split shape, 9.2% with true parent moments at split time and regression children, 11% with parent moments at about the cumulant carrier's accuracy, and 7.6% with everything true, against 62% for Q71p alone. Where a piece is cut barely matters; its accuracy when cut matters most. Giving each piece its own per-unit κ3 and κ4 leaves the error at 42%, so what the pieces lack is the cross-unit cumulants that Q71p carries as D21 slices, the same place Finding 3 puts the mean error on random nets. The remaining build is a mixture whose components each run the full cumulant step; the open question is whether a few hundred of them can cover a roughly 20-dimensional core below the cost of sampling.

Negative results

Each of these was measured end to end on held-out nets unless noted. Ratios are final-layer MSE (or score where FLOPs change) against the reference of the time.

Idea	What we measured	Result	Why it fails
Sample-based correction on top of the closure	MC, linearised-net control variate, antithetic pairs; net 52	MC noise ≈ 1000× the closure error at budget (2.5e-5 at 4096 samples vs 1.7e-8); CV ×1.25, antithetic ×1.11; optimal blend no gain	Per-neuron variance 0.067 needs millions of effective samples
Low-rank + sparse operands	Top-r SVD + largest q% entries, every product family	Rank 10% leaves 30–90% product error	Operands have flat spectra and dense remainders
Kink-only carrier	Drop the on-neuron relay; then e30_a's on-side source columns	Leg error 0.66–0.99 after 3–9 layers; on-side drop ×1.29–1.50 MSE for −3 to −4.5% FLOPs	On-neurons carry κ3 memory for 6+ layers; the relay has rank 100–225

Idea	What we measured	Result	Why it fails
Split weights by sign (W^+, W^-)	Regress residual error on per-neuron sign features, 16 nets, leave-one-net-out	−0.4% explained	An order-r cumulant transforms with r copies of W; the sign features we tried explain none of the error
More per-neuron learned features	~135 products, ratios and powers of the hook's inputs	≤ 1% of the residual (κ4 2–3.5%)	Residual is cross-neuron state, not a per-neuron function
Learned hook fitted teacher-forced	Ground-truth inputs at each layer, then live deployment, on e30_a	Minimal D3 hook ×1.001; 16-feature hook ×1.145	Trained on states the model never sees live; Q28 was fitted live
Partial port of a fixed-basis old tier	Full matched refit of every learned stage	−2.4% FLOPs, MSE ×1.37	e30_a's savings come from pieces that only work together
Young-transport column compaction alone	Net 40 pilot	−1.6% FLOPs	Most of e30_a's saving is row and contraction compaction we already had
κ22 squared-κ21 vertex	MC-confirmed term (coefficient 0.85), end to end	×0.997	The λ calibration already absorbs the level
Per-neuron radial κ31 shape (b1·v_i·Σ_ij)	On our carrier with learned stack	×1.002	Our K31_C term already carries most of the two-term law's gain

Idea	What we measured	Result	Why it fails
End-to-end table increments	Each layer's 87 table features propagated to L15 by the linearised map; 16×87 increments fitted to final error on live Q71p dumps, 26 nets, 5-fold	Held-out ×1.002–1.009; in-sample ×0.73–0.98	Inherited error is per-net state, not predictable from propagated features. On teacher-forced dumps the same fit shows a spurious ×0.24 (truth leaks through the forced input means)
Live sequential D3 hook on e30_a	Layer-by-layer live fit with upstream corrections on; probe after L8, nets 52–53	Score ×1.054	Where κ3 matters it is not predictable from per-unit quantities (Finding 4)
Rich-hook component ablation on e30_a	Variance, D3 and κ4 parts alone, each with a mean refit, nets 52–59	×1.001 / ×1.379 / ×0.994	Only κ4 helps, and below the 1% gate
Q50 readout on e30_a	Live capture of the last two layers, 36 features	×0.9997	The mean table already absorbs it
Final-error calibration of correction strengths	Three strengths fitted through live final predictions	×0.9992	Little left in the fixed correction directions
Stacking the sub-gate wins	κ4 strength + Q71b blend + one live mean-table refit, nets 52–59	×1.50, all 8 nets worse	Gains don't add; a one-pass refit of all tables shifts the states it was fitted on
Sum-of-cubes κ3 readout	flopscope 0.12.1 billing of one source's D21 readout	3.0 units vs 2.0 in mixed form	The D21 readout is never a plain Gram, so it gets no symmetric discount

Idea	What we measured	Result	Why it fails
Depth-6 Strassen on e30_a	Per family and all families, 7.6 GB RSS cutoff	Hub family ×0.996 on 3 nets, 7.67 GB on the 4th; all families over 7.6 GB	Memory-bound; at most ~0.4%
Sign-gauged even κ3 slices	Binned coherent fit of D21·diag(s) and D3∘s on even features only; net 40, layers 9–13, learning off	R^2 0.71–0.74 on D21, 0.66–0.73 on D3; 26–29% residual energy, spread over hundreds of directions	The odd remainder has the degrees of freedom of the weight rows; a gauge re-labels it and does not shrink it
Feedback rescale ported structurally to Q71p (Task L)	Diagonal rescale of the feedback channel, alone and with the feed and κ4 terms, learning off on net 55; then the full 87-feature mean table refitted on 0–51 and validated live on 56–59, learning on, against official Q71p	Learning off ×1.48 / ×1.65 / ×1.64; after refit ×1.33 (1.28–1.42 per net)	The V29-line ×0.74 does not transfer: on Q71p the regressed feedback increment is near zero at layers 4–5 and negative from layer 7
Carried-state corrector (Task M, after Susupro 18254)	Per-neuron MLP with a 32-dim state carried through W and W∘W, read-only on the final mean; 2.4 GF; trained on nets 0–51, validated on 52–59 with learning on, replacing and alongside the mean table	Replacement ×1.127; alongside ×1.002–1.003 (0.986–1.024 per net)	No gain on our stack. Untested: feeding corrections back into variance and D3, full-depth state, a larger corpus

Idea	What we measured	Result	Why it fails
Richardson extrapolation across history length	Exact plain K3 chain, net 40, learning off: K = 6, 8, 10, 15; combine $m_{K'} + a(m_{K'} - m_K)$ with a from the convergence ratio or least squares	Error correlation 0.77–0.78 between consecutive K; ratio-derived a makes MSE worse; LS a on the calibration net gains 4–5%; readout-only variant 0.4%	Truncation errors share a direction but are not collinear enough for norm-ratio extrapolation
Alternative-basis Strassen (Schwartz–Vaknin) at six levels	Pooled metered component with the last level fused; then the whole Q71q-a chain, broad (all families) and selective (products $\geq 1024^3$ only, depth 5 elsewhere); net 52, then paired 53–59, learning on	Component: +0.06% bill, 24% faster at 1024^2. Chain: broad 99 s wall, score ×1.012; selective 79.5 s, 329 GF, score ×1.007 (×1.0066 on 53–59); uploaded as #337318 and failed with an undetailed grader error	The grader pays for FLOPs, not speed; across the real shape mix the basis transforms and per-level additions are metered and outweigh the fused-leaf saving, while metered calls rise 50% (32k vs 21k per net). Predictions move by only 1.6e-6 RMS

Idea	What we measured	Result	Why it fails
Terminal residual corrector (after hyojun_kwon, 18255)	Substitute for their unpublished 33-feature map: 24 engine features + their two W-transported messages, 26→64→64→2 GELU (6,018 params), cost-weighted final-MSE objective; captured on Q71q-a's deployed trajectory, trained on 0–51 (epoch chosen on 40–51), scored on 56–67	Raw ×0.9995 (6/12 nets better); ×1.001 once its own 0.49 GF is billed	Flat at 52 training nets, which is what their reported scaling (gains from 256 to 760 nets) predicts; neither confirms nor rules out their result. A 300-net capture did not fit the budget
Puffi SSC closure slices (via Susupro 18254) on Q71q-a	Nine published Sym(C⊗C) coefficients ported unchanged (bit-exact at zero); learning-off screen on 52–55; then a 52-net capture, mean-table refit on 0–51 and paired learning-on 56–67 against Q71q-a	Learning off ×0.933 MSE (all four nets better) for +0.12 GF; after refit ×0.9988 (8/12 better, worst ×1.048)	The refit mean table absorbs the gain, as it did the λ and feedback-rescale screens: on this stack, structural closure fixes that do not change what the table can see mostly duplicate what it already learns

A hypothesis behind several of these. With the incoming activations fixed, pre-activation κ3 changes sign under $W_l \to -W_l$, so its average over a symmetric weight ensemble is zero. That alone does not make it unpredictable from weight-dependent state. Our evidence is empirical: the aggregate features we tried capture the global scale mode but miss a random-sign remainder (Finding 5). We take this as a hypothesis for why the memoryless κ3 rebuilds we tried stalled near half the true signal. κ4 and κ31 are fourth order and even under that flip, so the argument says nothing about the κ31 failures. The working λ law for κ4 is an empirical result, not a consequence of parity. One direct test of

the parity reading is in Finding 5: making the κ3 slices even by a sign gauge leaves the same 26–29% incoherent remainder, which is what a remainder shaped like the weight rows predicts.

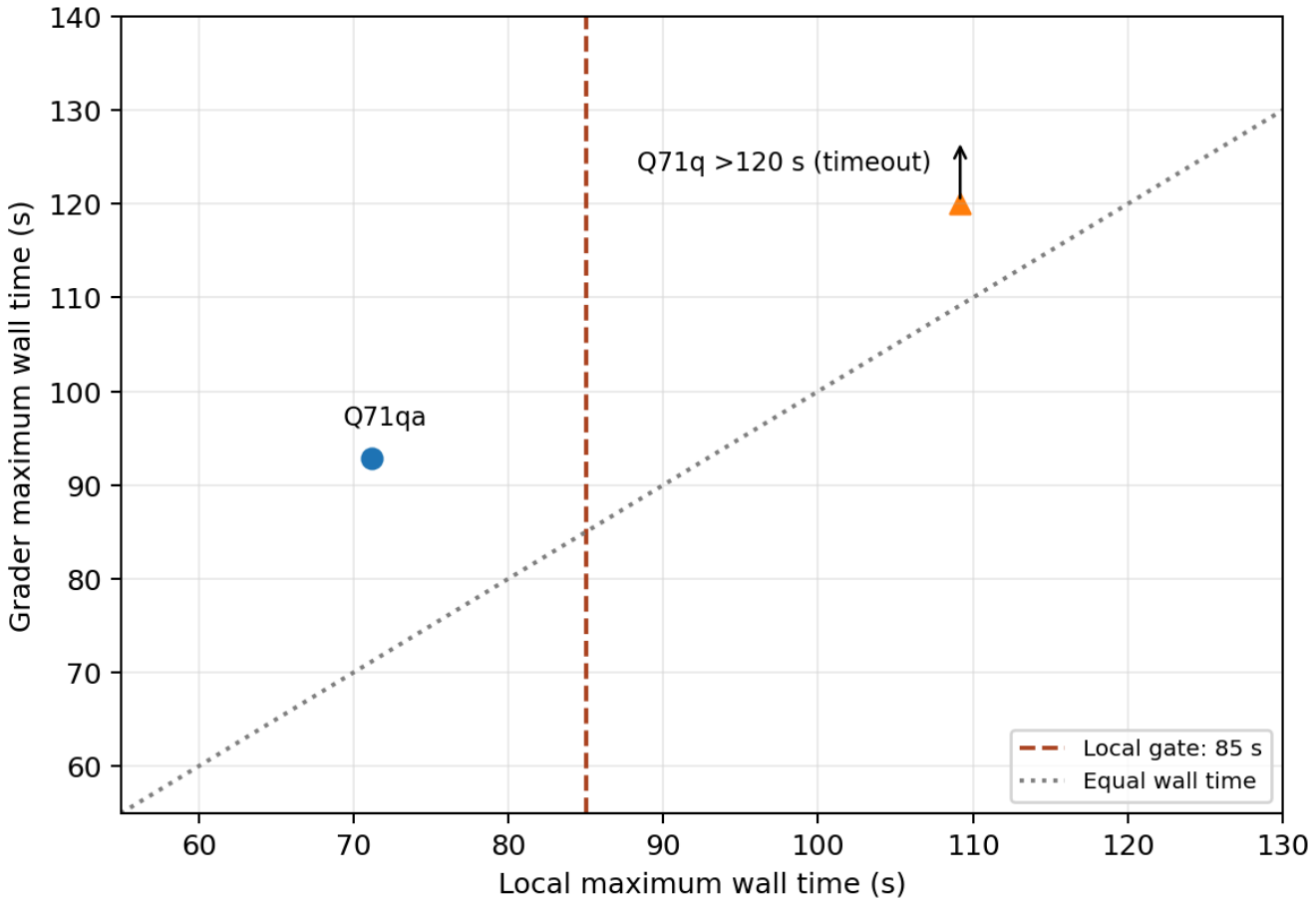
Grader engineering notes

These cost us failed submissions before we understood them; they may save others time.

- **Residual time is round trips, not FLOPs.** Locally, residual ≈ 35 µs per op, and on the grader every array slice is also a client–server round trip. A cache of views on pooled Strassen buffers cut round trips by 31% (44.5k → 30.7k per MLP) with bit-identical predictions. A routing change that raised op count by 46% failed every MLP on the 0.4 s cap.
- **The first call on each worker is different.** Pool creation and view building made the first MLP on each of the 5 workers run 0.41 s residual, against 0.32 s for later ones, and fail. Pre-creating pools, views and buffers for every shape in `setup()` (2.7 s, under the 5 s limit) fixed it: cold/warm residual ratio 1.035.
- **Server throughput varies.** On one graded run, 24 of 100 MLPs ran over 100 s wall. On those, server kernel time per billed GF was 15% higher at the same FLOPs, in index blocks, consistent with host contention. Keep at least 10–20 s of wall headroom; a forecast guard that switches to a cheaper route is a useful safety net.
- **Telemetry is on the submission page.** Per-MLP wall, server kernel, overhead, residual and FLOPs are embedded in the page as JSON. They let you compare your own runs, and others' public submissions, at the level of seconds per billed GF and op count.
- **Bit parity across platforms is the wrong gate.** Windows and Linux numpy/OpenBLAS gave final-output differences of 0.5–1.6% of baseline error with identical FLOPs. We replaced bitwise parity with a numerical policy for training and packaging: score change ⩽ 0.5%, prediction delta ⩽ 2% of baseline error, feature error ⩽ 0.2%.
- **Memory headroom matters for persistent workers.** Our builds peaked at 5.5–6.1 GB against the 8 GB limit. Process-local caches must be sized for the largest shape, not grown per call.
- **The grader host is about 1.3× slower than our laptop on this chain.** Q71q-a's local paired wall max of 71 s became 93 s on the grader; a build at 97–109 s locally (#337252) exceeded 120 s on every MLP and was rejected as REPEATED_MLP_FAILURE. Depth-6 Strassen cut the bill 4% but cost 50% more wall through small-op overhead, so the

meter saving was never collectable. Our packaging rule became: local paired wall max ⩽ 85 s, measured in a persistent process against the last graded build in the same session.

- **Local selection over-predicts by under a point.** The 52–59 block projected ×0.9865 for Q71q-a; the public 50 gave ×0.993. We treat local gains under 1% as unconfirmed until graded.



Grader values are Claude-reported; Q71p omitted (missing data).

Young-atom pruning, ported from Susupro (18254). Each young κ3 source carries one atom per neuron of its birth layer, as a column of every leg; transport and readouts are linear in the kept columns. Their rule scores each atom at birth by a bound on what later layers can read from it, $s_q = \|X1,q\|\|Y1,q\| + \|M_q\| + \|X3,q\|\|y\|$, keeps the fewest atoms holding 99.5% of $\Sigma\, s_q$ (rounded up to a multiple of 32) and zeroes the rest in every leg. The ranking is by read-norm, not energy; our own October center pruning ranked by energy, applied to all ages, and lost (+11% MSE at 0.999 energy kept).

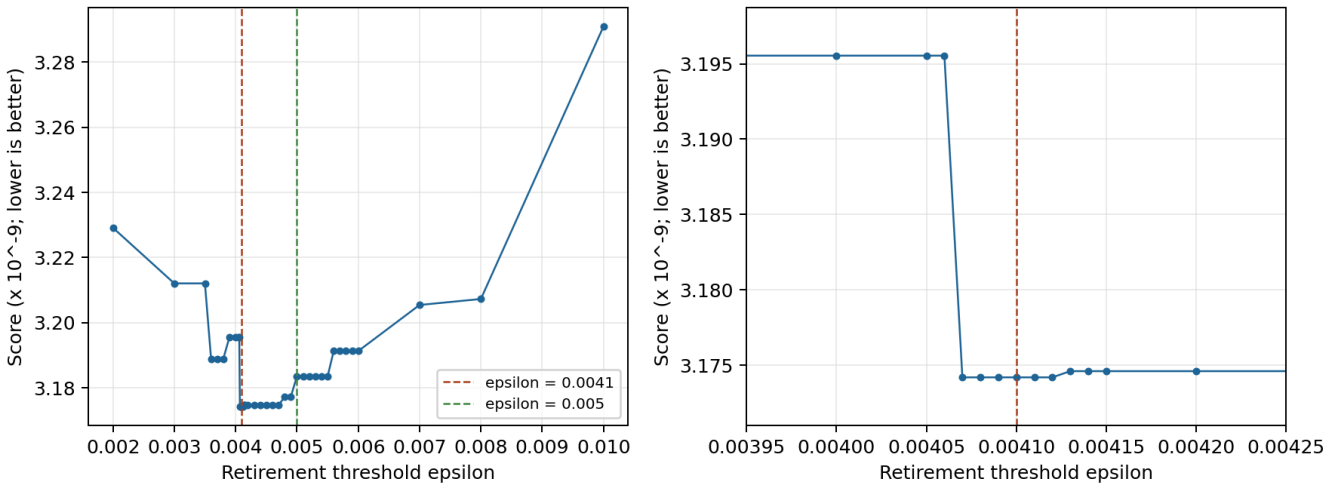
Chain, learning off (nets as stated per row)	FLOPs	MSE	Score
Their public chain, pruning on vs a near-no-drop control	×0.916	×0.997	×0.913
Q71p port, eps 0.005 (nets 52–55 mean)	×0.975	×1.010	×0.985
Q71p port, eps 0.002	×0.990	×1.005	×0.995
Q71p port + depth-6 pooled Strassen (net 52)	×0.956	×1.009	×0.964

- On our chain the rule finds a third of what it finds on theirs: e30_a's dead-unit and column compaction already removes most of the atoms it would drop. The port reproduces Q71p bit-exactly at eps 0.
- Re-scoring atoms after birth by the same bound (ages 2–3: MSE ×1.058 for −1.2% FLOPs; every layer: ×1.136) makes things worse: the atoms that have decayed are still being read.
- The source-column cut at $\Phi(\alpha) < 0.014$, which is −3.5% at raw ×0.998 on their chain, costs ×1.06 MSE for −3.6% on ours and drops out of the combination.
- The same net with no learning on either side: their chain 1.746e-8 at 368 GF, Q71p 2.121e-8 at 336 GF. Their engine is 18% more accurate at 10% more cost before any correction. That gap is structural (their SSC closure port, −5.4% raw; the K=4 terms their chain keeps) and is where the accuracy difference between V29 descendants now sits.

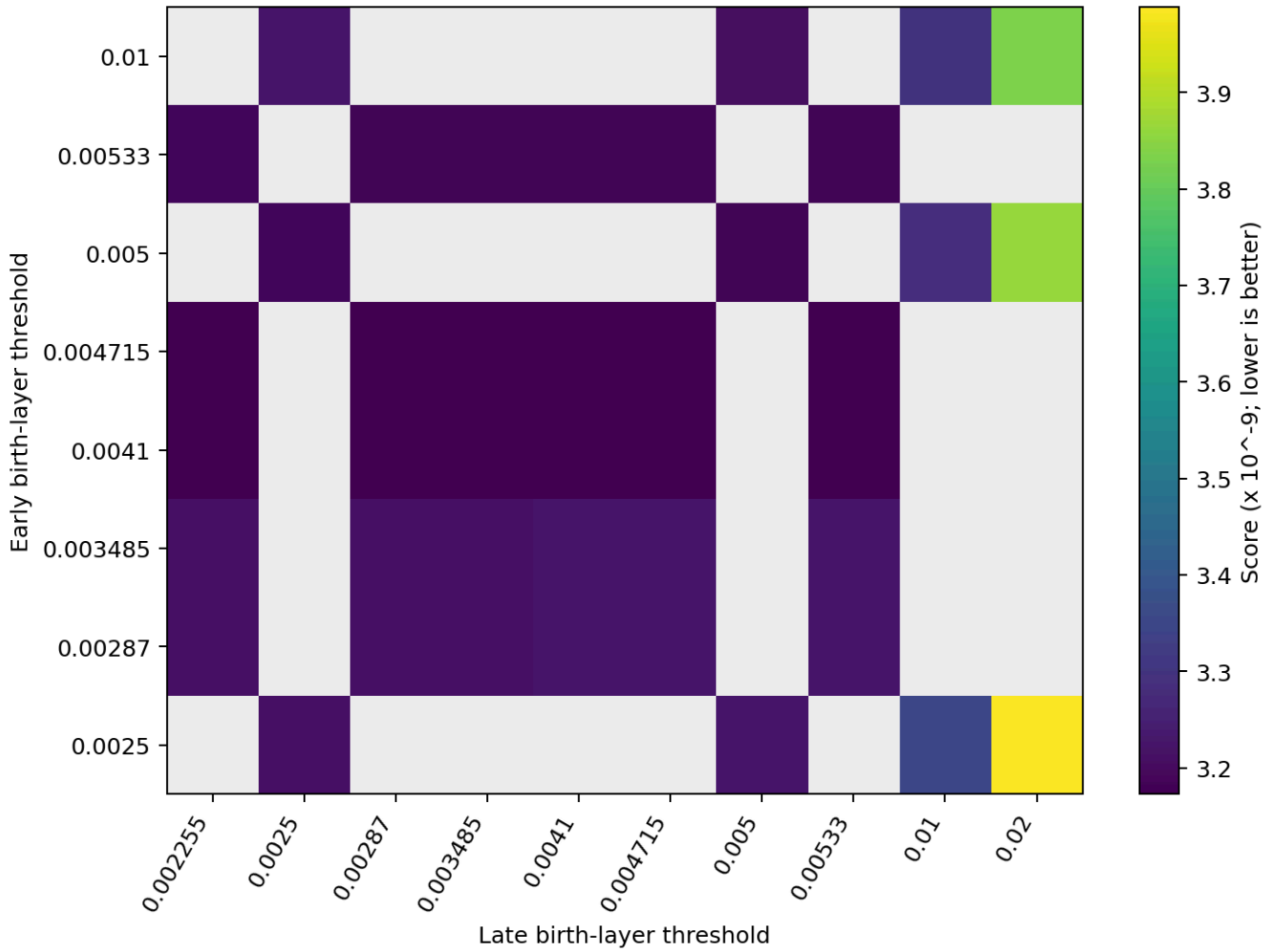
Outcome on the grader. The pruning + depth-6 candidate (eps 0.005, mean table refit on 0–51 of the pruned chain) validated at ×0.960 learning-off and about ×0.97 learning-on locally, but every MLP timed out on the grader (#337252: local wall 97–109 s, grader > 120 s on all four attempts). The same pruning on Q71p's own depth-5 arithmetic, with the same refit table, went up as Q71q-a (#337276): public score 3.062e-9 against Q71p's 3.084e-9 (×0.993), raw final MSE 2.07e-8, 326 GF (14.8% of budget), grader wall 78–93 s. Locally the paired block 52–59 had projected ×0.9865; the grader gave ×0.993, so our selection block over-predicts gains by about 0.7 points.

Threshold and birth-layer grids. A fine scalar sweep put the learning-off optimum at eps 0.0041 (×0.9805 vs ×0.9834 at 0.005, net 52). Two birth-layer grids (nine cells around 0.005; 25 cells around 0.0041 plus a five-cell edge extension) found their best cell within 0.04% of the scalar rule, and every cell that loosened the late-layer threshold failed the ×1.03 MSE cap; the threshold does not want to vary with birth layer. Rounding retained atoms to multiples of 64 instead of 32 cost 2%; re-scoring by age cost more than it saved;

eps 0.01 did not recover under its own refit (corrected 0.01 vs corrected 0.005: ×1.10). Combined with depth-6 Strassen the candidate is memory- and wall-bound, not FLOP-bound (see Grader engineering notes).



Network 52; learning off. Right panel enlarges the fine sweep.



Network 52; learning off. Gray cells were not measured; grid spacing is categorical.

The learning-off optimum did not survive learning. Q71q-c, eps 0.0041 on Q71q-a′s arithmetic with Q71q-a′s mean table carried over unchanged, scored ×1.0043 against Q71q-a on 56–67 (4 of 12 nets better, worst ×1.041), and its predictions moved by 8% of the baseline error, so the table fitted at 0.005 does not transfer even to a slightly different retained set. Without a fresh capture and refit (not worth its cost for an expected gain under 0.5%) the scalar 0.005 stands. Retirement is a joint selection over each source′s birth coordinates, shared across all of its legs, so the per-layer counts we report cannot attribute the score change to any one leg.

Limitations and open directions

The gap that matters is to the leaders: about 1.3–1.5e-9 public score, at 230–290 GF with about 40% lower MSE than ours at every layer. Nothing in this write-up closes it.

Limitations

- **The designated estimator′s own delta over e30_a is small.** It is a structural choice (K31_C = 1, λ 1.0 / 0) and a refit table, worth about −14% score against our own best and −1% against e30_a′s public score; young-atom pruning, adapted from Susupro′s rule, adds a further −0.7%. The contribution we claim is the analysis in Findings 1–5, not that delta.
- **Most MC anatomy uses one or two nets** (40, 41). The effects are large and consistent across those two, but a wider check is open.
- **Local estimates ran about 4% above the grader** for the same package; ratios between variants transferred well.
- **The beyond-the-mean results rest on 61 random bias-free ReLU networks with Gaussian input and one trained network.** The saddlepoint and readout-covariance tests use three to four nets, so their samples-to-match are bounds; biases, non-Gaussian inputs, other activations and convolution or attention are untested, and the learned parts (κ3 calibration, mean table) are specific to the input distribution they were fitted on.
- **Two uploads failed on the grader for arithmetic reasons** (wall time, and an undiagnosed error on the alternative-basis build) after passing every local check; the grader′s own telemetry was the only reliable judge of cost-side changes.
- **The original live-fitted Q28 hook was not fully ported to e30_a.** Teacher-forced hooks and one live sequential D3 hook were tried there, and both failed.

Open directions, in the order we would try them

1. **Better pair-cumulant births.** The prize is measured: exact κ31 at layers 9–13 gives ×0.881 on net 40, and exact κ31 with exact D21 ×0.830, the 75–80% of newborn covariance error identified in Finding 3. The affordable forms fall short. Carried κ4 sources on the existing legs explain 54–61% of true κ31 against our 47–52% and give ×0.963 for at least +25% FLOPs; sketched forms give about ×0.97–0.98 for +10–20 GF. The leading-order tree births close only 10–20% of the deep κ31 gap; the rest looks like κ3-born κ4, at about 12 n^3 per κ3 component per layer. The one open measurement is κ31 carried at layers 9–13 only, where most of the gain sits (layer 10 alone ×0.948 exact). The public combination of a carried κ4 core with scale-mode extraction (il.miscusi, ×0.84 MSE at 2× the FLOPs) is unverified on our stack; their own report says the extraction hurts e30_a with its mean table on (4.73×) and does not combine with the λ law, and acco-cyber found a κ4 core fed into the κ31 readout dead (+133%). Susupro also reports that the non-tree birth C_jk κ3_ijk is worth 1.56× in their K=4 chain and is absent from V29-lineage K=3 chains; Q71p′s nonlinear-map term table confirms the absence (every term is an own-index or same-pair product of two-index slices; nothing contracts a third index). Our September prototype measured that term at 1–2% of the carried D3 in a K=3 chain and dropped it as n^4; its value when it feeds κ4 is untested here.
2. **Shared-mode extraction.** Treat the input-scale mode (and any other collective mode) exactly and carry only the remainder. We tested this on the old κ3 tier (Finding 5): the scale mode is identified cleanly and dominates the norm (R^2 up to 0.82 on old D21, 0.90 on old D3, with one consistent ε per layer), but the remainder is broadband, needing rank 121–228 for 95% of its norm, and εS plus a rank-8 remainder costs ×4.5. Extraction does not shorten κ3 history on our stack. il.miscusi reports ×0.38 MSE from scale extraction with a carried κ4 at twice the FLOPs, and that it hurts e30_a with its mean table on (4.73×); whether a carried κ4 changes what the remainder needs is untested here. Per-unit learned fixes on e30_a, including a live layer-by-layer fit, are closed (Finding 4; Negative results).
3. **A different representation.** The leaders′ telemetry shows 0.29–0.42 s of server kernel per billed GF against our 0.19, 20–50% more ops, and residual near the cap. Their budget goes into many small or elementwise operations rather than large products. A sample-based ensemble does not explain it (dense matmul is fast per FLOP), and neither does a kink-only carrier (≤ 5% FLOPs on e30_a). A search of the literature and every public write-up found no cheaper exact representation of old κ3 (Finding 5). Susupro′s matrix-pair compression, the one lead the search turned up, fails at third order (×21.8 at the one-pair rank where it would save compute; Finding 5). A Susupro-style carried-state corrector, a 32-dim per-neuron state carried through W and W∘W

and read out on the final mean, gives ×1.002–1.003 against Q71p on nets 52–59 with learning on after training on 0–51, and ×1.127 as a replacement for the mean table. It does not confirm that lead on our stack; what it leaves untested is a corrector fed back into variance and D3 rather than read out at the end.

Follow-up after the contest. Our tests of rare input-noise failures, backdoor detection and training-health measures on trained networks are written up separately in Rare noise failures in trained ReLU networks.