

Estimating the Activations of Random Deep ReLU Networks Without Sampling: Cumulant-History Moment Closure Under a Metered Compute Budget

Suliman Tadros (AICrowd: [suliman_tadros](#)) — ARC White-Box Estimation Challenge 2026, Phase 2 algorithmic-contribution write-up

This write-up corresponds to exactly one submission: #333755 (official score $1.919444765208421e-9$; final-layer MSE $1.5706e-8$ at a FLOP fraction of 0.122; 100/100 networks graded; residual wall time median 0.311 s, max 0.335 s on its first run).

Abstract

We estimate every layer's per-unit post-activation mean of a random width-1024, depth-16 He-initialised ReLU network under Gaussian input, in about $2.6e11$ metered FLOPs per network (12% of the 2^{41} budget), with no Monte Carlo sampling. At this budget sampling is hopeless: reaching the mean-squared error of our method ($1.6e-8$) would need several million forward samples, hundreds of times the budget. Our estimator propagates an explicit moment state (mean, dense covariance on active units, a compressed history of third cumulants in canonical-polyadic form, and an isotropic fourth-order term) through the network, closes the ReLU non-linearity with Gaussian-plus-Edgeworth corrections, and adds small learned correction heads. We contribute (i) a low-rank third-cumulant history that makes third-order effects tractable at width 1024 and separates our pipeline from Gaussian moment propagation, which is about 330x less accurate; (ii) exact-linear refits of the output-side heads and Gauss-Newton calibration of physical closure constants on held-out networks; (iii) an analysis of how computation is billed and timed when every array operation is a metered client-server round trip, with exact rewrites and a round-trip-reducing vectorisation that together removed about 6% of FLOPs at no accuracy cost; and (iv) a diagnostic study of where the remaining error comes from: 81% of it is born in the last six layers, the Edgeworth closure is exact to $\sim 1e-10$ when fed the true pre-activation moments, and injecting true moments would cut the error by 60% - but the gain vanishes once those moments carry 0.1% noise, which bounds what any learned correction of this family can achieve.

1. Introduction

White-box estimation asks for a quantity of a neural network's behaviour - here the expected activation of every unit - computed from the weights alone, without running the network on samples. The challenge instance is a random multilayer perceptron of width $n = 1024$ and depth 16 with He-initialised weights $W_l \sim N(0, 2/n)$ and no biases, driven by $x \sim N(0, I)$. For every layer l and unit k we must return $E[\text{relu}(z_{\{l,k\}})]$. The score of a network is the final-layer mean-squared error times $\max(0.1, F/2^{41})$, where F is the number of floating-point operations counted by the competition's metering library (flopscope); the official score averages this over networks. Two runtime gates apply per network: 120 s of wall time and 0.4 s of "residual" time, i.e. Python time spent outside metered operations; a network that breaks either gate scores as all-zero predictions.

The problem is hard for three reasons. First, sampling is out of reach: matching an MSE of $1.6e-8$ needs several million forward passes, hundreds of times the budget, and control variates do not help: in the composed arc-cosine spectrum at depth 16 only about 19% of the per-sample variance is in degree one, while a useful control variate would need to explain more than 99.9%. Second, a Gaussian moment propagation - the natural analytic baseline - is about 330x too inaccurate, because finite width makes pre-activations increasingly non-Gaussian with depth, mostly through the units that sit near the ReLU kink. Third, exact higher-order propagation is unaffordable: the third cumulant of a layer is an $n \times n \times n$ tensor. Our approach keeps a compressed history of the third cumulant that is cheap enough to transport through sixteen layers and accurate enough to recover most of the non-Gaussian correction.

Prior work and credits. Our implementation began from the AICrowd starter kit and builds on ARC's public work on cumulant propagation through random MLPs (Wu et al., arXiv:2605.05179; code at github.com/alignment-research-center/mlp_cumulant_propagation, MIT). Fitted constants were derived only from the public ARC WhetBench dataset (CC BY 4.0). The Strassen-Winograd identity is an established algorithm, not a claimed contribution. The contributions below are what we added to that framework or measured about it.

Contributions:

1. Cumulant-history moment closure: births of rank-one third-cumulant factors at ReLU kink units, exact transport through the weights, weighted-SVD compression of aged blocks, and diagonal/mixed-slice read-outs feeding mean and covariance updates.
2. Learning and calibration on held-out networks: Gauss-Newton calibration of closure strengths; linearised and exact-linear refits of correction heads (the returned final layer is linear in the terminal head constants).
3. Compute engineering under per-operation metering: a cost model for Strassen-Winograd depth, size-thresholded buffer reuse, and exact "skip unused outputs" rewrites (read-out rows, SVD-factor reuse), plus the lesson that interpreter round trips and new large server-side arrays, not FLOPs, bound further savings.
4. Error anatomy and limits: error-birth decomposition, oracle substitution of true moments, tolerance of the gain to moment noise, and a survey showing why sampling, Gaussian-plus-learned-residual and non-local learned heads cannot close the gap.

2. Background and problem setting

For a pre-activation z with mean m , variance v , third cumulant k_3 and fourth cumulant k_4 , the Edgeworth expansion of the ReLU mean is $E[\text{relu}(z)] = m \Phi(a) + s \phi(a) + k_3 * g_3 / 6 + k_4 * g_4 / 24 + \dots$, $a = m/s$, $s = \sqrt{v}$, with $g_3 = -a \phi(a)/v$ and $g_4 = (a^2 - 1) \phi(a)/(v s)$ the third and fourth derivatives of the Gaussian ReLU mean; analogous expansions give the second and higher post-activation moments. Units far from the kink ($|a|$ large) are essentially linear or dead and are insensitive to k_3 , k_4 ; units near the kink are where non-Gaussianity matters. Propagating a layer is linear in the post-activation statistics: the next pre-activation mean is $W^T E[\text{relu}(z)]$, its covariance $W^T C_{\text{post}} W$, and its third cumulant the three-way contraction of the post-activation third cumulant with W . We verified on a development network with $1e8$ -sample truth that, given the TRUE pre-activation m , v , k_3 , k_4 per unit, this first-order Edgeworth closure reproduces the true post-activation means to $\sim 1e-10$ MSE (below sampling noise), so the difficulty lies entirely in propagating accurate moments, not in the closure formula (section 6).

3. Method

3.1 Moment state and layer update

Per layer we keep, for the pre-activations: the mean m , the covariance C (dense on "active" units with $a = m/s > -3.25$, only the variance on the deep tail), a third-cumulant history (3.2), and an isotropic fourth-order ("radial") scalar that models the finite-width kurtosis common to all units. Post-ReLU moments use the Gaussian-conditional formulas plus Edgeworth corrections from the third-cumulant diagonal $td_k = T(k, k, k)$ and the radial term. The next layer's mean and covariance follow by the linear map W , the covariance through a symmetric blocked product $W^T C W$ that computes only the upper blocks.

3.2 Third-cumulant history in canonical-polyadic form

At every layer, each unit near the kink ($|a| < 3$) "births" a rank-one star factor of the post-activation third cumulant: the part of T_{post} generated by that unit's non-linearity, expressed through the unit's own response and its covariance column. A birth block stores, for its pivot units, vectors x (the pivot coordinate), y (the

covariance column) and coefficients (a, b, c) such that the block's contribution is $\sum_c [a_c x_c(x) y_c(x) y_c + b_c x_c(x) x_c(x) y_c + c_c x_c^3]$ plus a low-rank remainder (z, vv). Blocks are transported through later layers by projecting their vectors through W (one matrix product per block per layer) and scaling by the layer gains. Young blocks (ages 1-4) are kept exact; at age 5 a block is compressed by a weighted SVD of $[x \sqrt{w_x}, y \sqrt{w_y}]$ to rank 256 (224 before layer 8), with a trace/diagonal repair term. Two read-outs feed the closure: the diagonal td (all kept units; mean and variance corrections) and the mixed slices $T(j,j,k)$ for rows with $|a_j| < 2.5$ (covariance update). Without the history the estimator reduces to Gaussian moment propagation (~330x less accurate); compressing it costs at most ~4% of the raw error.

3.3 Covariance closure

The post-activation covariance uses Gaussian pair terms (Hermite expansion in the correlation up to second order), a third-cumulant correlation term $0.5 T(j,j,k) g_{2j} p_k + \text{transpose}$, fused Gaussian-star closures (mixed and cubic), a fourth-order pair term K22 built from the radial scalar, and alpha-dependent closure factors (quadratic in clipped a). The deep tail keeps only its variance.

3.4 Correction heads and exact terminal refits

Three small learned heads correct what the closure misses. (i) A mean-response head adds to each unit's post-activation mean a term quadratic in 20 standardised local features (a, v, td, the radial term, layer gains, ...) plus a 20-32-32-1 tanh network. (ii) Shadow and tree heads modulate the fourth-order contributions, linear in the features and in the response network's hidden layer. (iii) A linear diagonal-transport (DT) head rescales the transported cumulant diagonal by $(1 + \text{pdf} * \text{beta})$. Two structural facts make these heads refittable cheaply and exactly. First, the head's source enters the output through a linear recursion $q_l = (q_{l-1} W_l) g_l + s_l$, so its 264 linear parameters admit a ridge Gauss-Newton fit from per-layer captures. Second, the returned final layer is exactly linear in the terminal head and modulation constants (checked by finite differences), so the last two layers carry their own copies of these constants, fitted in closed form; layer 14 enters through one transport step.

3.5 Calibration on held-out networks

Physical closure constants (strengths of the Edgeworth, star and K22 terms; alpha-dependent closure factors; compression and readout thresholds) are calibrated by Gauss-Newton on finite-difference Jacobians of the final-layer error over held-out networks. Fitting uses consumed public networks (432 FIT networks from earlier shards; 48 TEST networks never used for fitting); every step is pre-registered with an explicit acceptance gate and judged by paired per-network differences, because the per-network paired standard deviation (0.7-2.8%) makes 10-network evaluations unreliable. Each calibration transferred to the official board: constants -0.93%, closure strengths -0.81%, alpha factors -0.23%, head refits -0.40/-0.24/-0.09%, separate last-two-layer constants -0.20%.

4. Computing under a metered budget

The final estimator costs about $2.6e11$ metered FLOPs (FLOP factor 0.117-0.122 of 2^{41}), dominated by the covariance product $W^T C W$, the transport of third-cumulant blocks and the SVD compressions. Large products use a metered Strassen-Winograd recursion whose depth (4-6) is chosen by a dimension-only cost model; operands are padded and chunked to respect the server's 100 MiB array limit.

The binding constraint late in the competition was not FLOPs but residual time. The grader runs the estimator as a client of a flopscope server; every array operation, slice and broadcast view is a round trip, and residual time grows by about 4-5 us per round trip. We rebuilt the client-server setup locally, counted round trips per source line, and learned three rules. (1) Size-threshold every FLOP-saving rewrite: output assembly into preallocated buffers, packed operand buffers and in-place chunk outputs save FLOPs in proportion to array size but cost a fixed number of trips per call, so applying them only above a dimension threshold keeps most of the saving (35% of assembly calls carried 89% of its saving). (2) Skip unused

outputs exactly: reading out compressed-history slices only for the rows that use them, and reusing SVD factors, saved 1.5% and $\sim 1\%$ of FLOPs with no extra trips and predictions identical up to float32 reassociation. (3) New large server-side arrays cost residual time beyond their trip count: a restricted covariance product that barely changed the trip count failed the official gate, and a stack of deeper recursion and wider packing (+4,763 trips) cost about twice the per-trip estimate. With every trip-costly rewrite enabled and the gate ignored, FLOPs fall by only $\sim 3\%$, so the remaining score headroom lies in accuracy, not arithmetic. A shape-only replay of the 193 top-level Strassen calls per network (a mock array type that counts round trips and metered FLOPs, exact against measured deltas) shows that the four recursion/packing thresholds trace a near-linear frontier of about 2,000-2,200 round trips per 1% of FLOPs: under a fixed residual cap the recursion parameters offer no free saving. (Figure 2.)

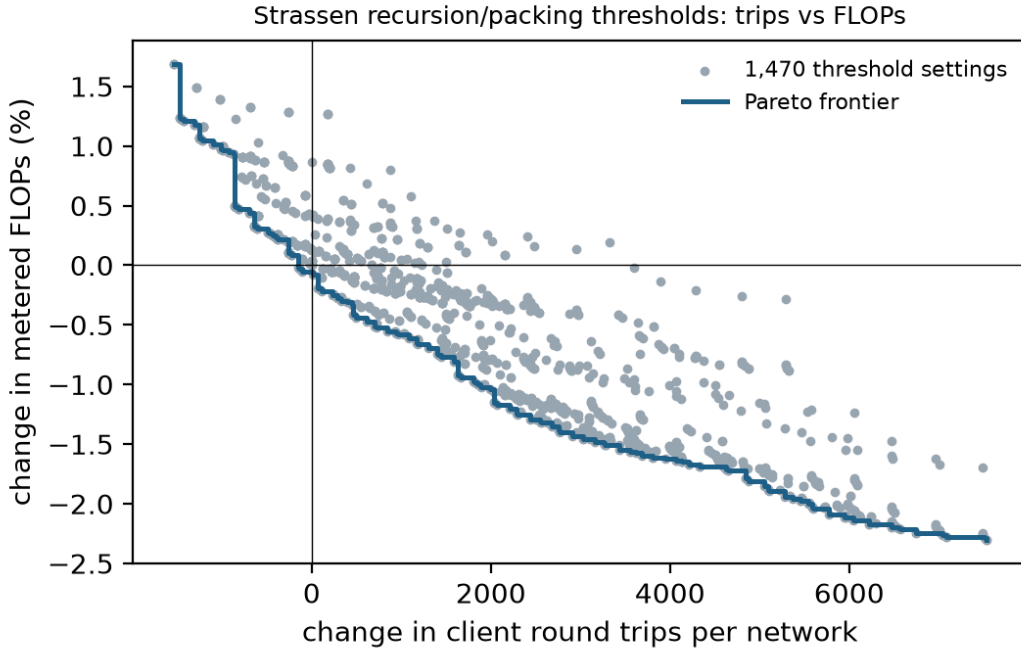


Figure 2. Round trips vs metered FLOPs for the Strassen recursion and packing thresholds (shape-only replay): a near-linear frontier of about 2,000-2,200 round trips per 1% of FLOPs.

The way out was to make room in the residual budget instead. Most non-Strassen round trips came from Python loops over history blocks (about 25-30 small array operations per block, ~ 10 blocks per layer). Executing those operations once per layer on the concatenated block columns, with segment sums for per-block reductions, removed 3,100 round trips at float32-reassociation parity; two further rounds of exact rewrites (batched closure weights, matrix-form trace contractions, fused channel states, in-place diagonal fills, gathered rather than recomputed quantities) removed another 4,622. Officially the median residual fell from 0.3345 s to 0.3107 s, and the freed budget bought the deeper recursion that had previously failed the gate: the final-eligible score moved from $1.9665e-9$ to $1.9194e-9$ (-2.4%) with raw MSE unchanged. Two measurements matter for anyone working under such a cap: the official cost was ~ 4 us per removed round trip, and vectorisation that creates new large per-layer arrays gives part of the saving back (the first vectorised build had 549 fewer round trips than its predecessor but a 3 ms higher median).

5. Experimental protocol and results

Protocol. All development decisions were made on consumed public networks: FIT (432 networks from shards 28-51 and 55-57) for fitting, TEST (48 networks, shards 52-54) for paired acceptance tests, and a 1e8-sample development network for diagnostics. Each experiment had a plan frozen before outcomes, with an explicit gate (typically a paired mean gain with $t > 3$ on TEST and an official-runtime budget).

Official trajectory (exact scores, score = MSE x FLOP factor; Figure 1):

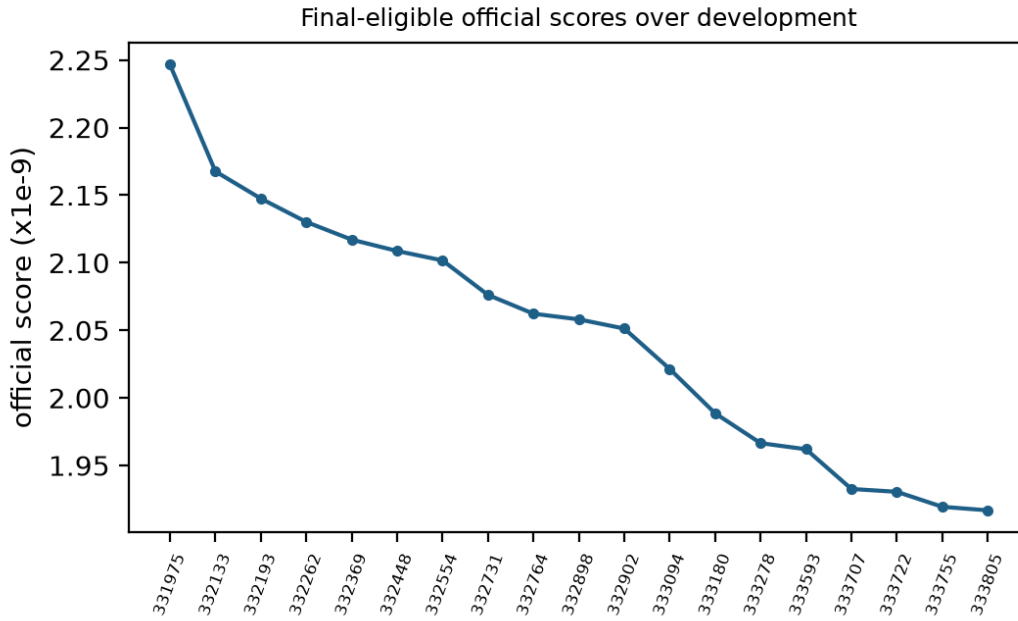


Figure 1. Official scores of runtime-verified candidates over development. The last point (333805, 1.9168e-9) is a variant with a 7-20 ms higher residual-time baseline than 333755; we name 333755 here because it is the candidate we audited and chose as our primary final.

Submission	Change	Score
331975	baseline cumulant-history pipeline	2.2466e-9
332133	compression at age 5, rank 256	2.1676e-9
332193 / 332262 / 332369	Gauss-Newton constants / closure strengths / alpha factors	2.1475 / 2.1302 / 2.1171e-9
332448 / 332554	mean-head refits (linearised, larger data)	2.1087 / 2.1017e-9
332731 / 332764	size-thresholded assembly / packing	2.0762 / 2.0623e-9
332898 / 332902	last-two-layer constants / in-place chunk outputs	2.0581 / 2.0513e-9
333094	readout on readout rows only (exact)	2.0214e-9
333180	SVD-factor reuse + diagonal-transport mix	1.9886e-9
333278	buffer-reuse band (leading verified candidate)	1.9665e-9
333516	deeper recursion stack (residual max 0.397 s; not final-eligible)	1.9339e-9
333593	CR24 two-hop weight-graph terminal head (+0.18% held-out)	1.9620e-9
333707	vectorised per-history-block loops (-3,100 trips) + selective deeper recursion	1.9326e-9
333722	further exact trip cuts (-2,546)	1.9305e-9
333736	depth-6 rule relaxed (X=250000)	1.9222e-9
333755	further exact trip cuts (-2,076), fewer large server arrays (residual max 0.3345 s)	1.9194e-9

From 331975 to 333278, raw final-layer MSE moved from 1.633e-8 to 1.575e-8 (-3.5%) and the FLOP factor from 0.138 to 0.125 (-9%); about 30% of the improvement is accuracy (calibration and heads) and 70% arithmetic (exact rewrites). Our final is chosen among runtime-verified candidates (residual max $\leq$ 0.36 s across official runs): the leading final is 333755 (1.9194e-9; residual median 0.311 s, max 0.335 s;

two independent audits, a 384-network sweep with every network improved and a 48-network paired test at +2.13%); the private result on fresh networks is pending at the time of writing.

5.1 Runtime variance and the choice of final submissions

Scores are deterministic: repeated official runs of identical code return identical scores. The per-network residual time is not. Across nine official runs of 333755 the slowest network took 0.334-0.348 s in seven runs and 0.361 s in one. In the remaining run the whole grading batch slowed (90th percentile up by ~13 ms, four networks above 0.36 s) and the slowest network reached 0.391 s, 9 ms below the 0.4 s cap at which a network scores as all-zero predictions. Because the private re-evaluation keeps the better of two designated submissions, we pair the lowest-residual high-scoring candidate with a variant traded for runtime margin rather than with the best public score, whose margin is the thinnest. The margin variant switches off one depth-extension rule. Over 384 held-out networks, paired, it changes raw MSE by -0.011% (s.e. 0.008%) at +1.94% FLOPs, so its score is 1.9% worse. In exchange, its slowest-network residual time over five official runs was 0.301-0.322 s, about 30-40 ms below 333755's.

6. Where the error comes from, and what limits this family

Error birth. A linearised decomposition of the final error into contributions born at each layer attributes 81% to layers 10-15; sampling noise in the truth is ~3.5% of the final MSE. The closure is not the problem. Given the true pre-activation mean, variance, third and fourth cumulants per unit, the Edgeworth ReLU closure reproduces the true post-activation means to $\sim 1e-10$. The covariance closure form is likewise exact to ~3% of the local covariance error when fed its own inputs exactly; that local error splits into covariance error carried from earlier layers (~55%), error in the third-cumulant slices (~22%) and an unexplained remainder (~25%). The slice error is not caused by compression, birth window or readout window (all variants give identical slice accuracy).

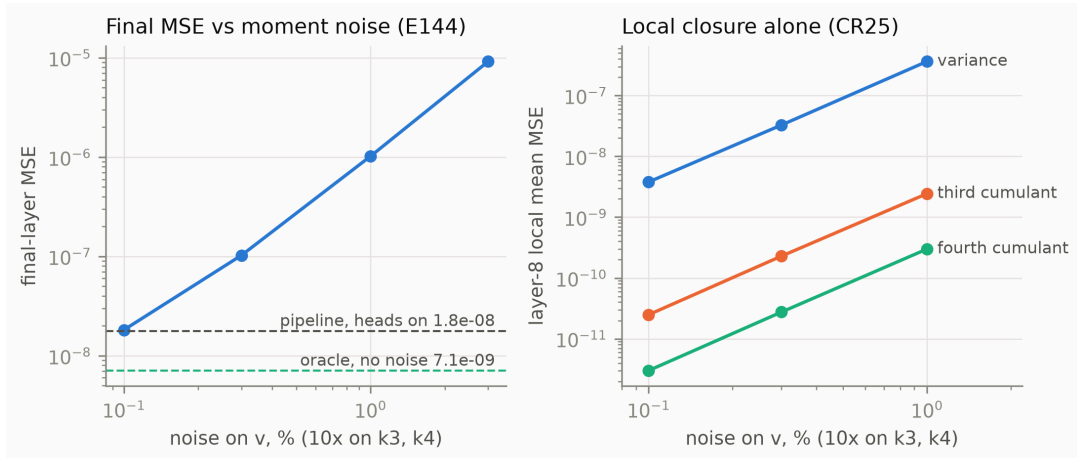


Figure 3. Left: final-layer MSE when the true moments injected at layers 8-15 carry multiplicative noise (eps on the variance, 10 eps on k3 and k4); 0.1% noise already erases the oracle gain. Right: the same noise in the layer-8 local closure alone, by component; the variance term dominates.

Oracle ladder. Injecting true variance, k3 and k4 at layers 8-15 (heads off) cuts the final error to 0.40x the full pipeline; from layer 11 alone, 0.575x. But the gain disappears with 0.1% noise on the variance (1% on the cumulants) and turns into a 5.8x loss at 0.3%. The state errors at kink units (variance 0.06-0.13%, k3 ~4%, k4 20-29%) are made of uncorrelated components that local features predict with held-out $R^2 \leq 3\%$. Co-adaptation. Heads, closure strengths and output factor sit at a joint optimum: improving the physics alone (e.g. +18% heads-off from recalibration) loses accuracy once the heads are on, and joint refits do not recover it. Implication. A better estimator in this family needs late-layer variances consistent to ~0.05% and cumulants to ~0.5%; the limiting mechanism is the coupled propagation of covariance and third cumulant, not any single closure formula.

6.1 What the remaining error is not (October measurements)

A sequence of oracle and fidelity tests, each with a bar fixed before the run, narrows the residual further.

- Floating point is not the limit: running the whole estimator in float64 changes raw MSE by a factor 1.0001 (0.998-1.001 over six networks) at twice the FLOPs.
- The covariance closure form is exact to $\sim 3\%$ of the local covariance error; the third-cumulant birth form is exact to 0.04% given its inputs. The residual post-covariance error is 1-2% of the non-Gaussian pair energy and is near-orthogonal to every second-order descriptor tried: Gaussian-mixture/regime splits explain at most 1.6% of it (correlation ≤ 0.14), and a carried gradient-covariance state $\Theta = E[\text{grad } z \text{ grad } z^T]$ explains at most 0.5% (higher chaos orders are already inside C: only 19-25% of late pre-activation variance is chaos order one).
- Injecting exact third/fourth-order information at layers 8-14 would cut the final error to 0.43-0.45x, but (i) late windows keep little of it (layers 12-14: 0.74-0.76x), (ii) single-layer fixes do not propagate (the third-cumulant diagonal error is re-born at every layer), (iii) a carrier must reach correlation $\geq \sim 0.93$ with the true correction to deliver a rank-changing gain, while the exact source computable from the model's own state reaches only 0.27-0.55 (third cumulants) and 0.66-0.72 (K31), because 69-80% of the birth error sits in the inputs.
- Operating-point changes do not help: refitting all constants and heads at 0.9x or 1.1x the FLOPs recovers 0-0.8%, and a perfect per-network choice among 14 configurations gains $\leq 1\%$.

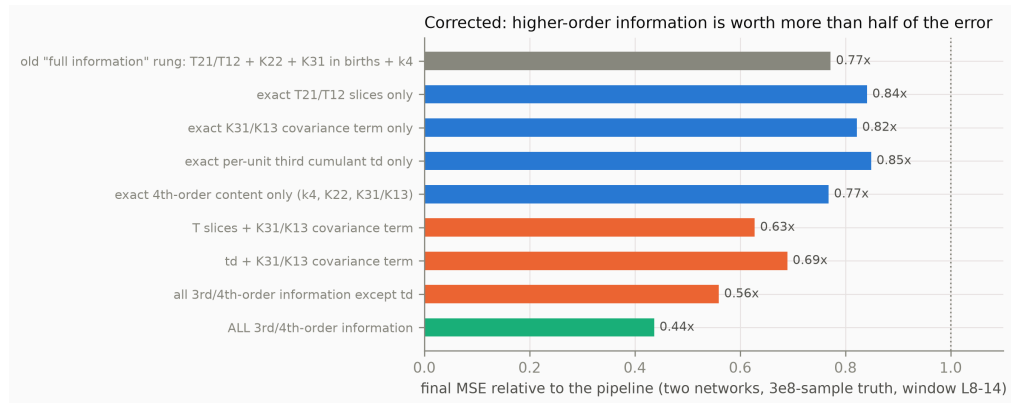


Figure 4. In-pipeline oracle ladder (two fresh networks, 3e8-sample Monte Carlo truth, exact information injected at layers 8-14 so the model itself computes the scored final layer). Exact third- and fourth-order information together is worth 0.44x of the final error; no supplier built from the model's own state reaches it within budget.

- With Monte Carlo truth at every depth (layers 1-15, one typical network), injecting exact third- and fourth-order content at layers 1-14 gives 0.35x. The gain is depth-flat, matching the per-layer profile of the leading public scores. Uniform fidelity of about 0.59 at all depths would be needed for 0.80x. Measured per part, the model's own third-order slices are near-exact (T21 and diagonal relative error 0.05-0.11). The other third-order slice (T12) degrades with depth only because rows of strongly active units are not computed, and exact T12 alone does not help (1.03x). Fourth-order content is essentially absent at every depth (correlation -0.17 to 0.26).
- The small T21/diagonal residual is nevertheless worth 0.63x if removed, and it is closure error, not error in the inputs. It is present after a single layer with no history, it is orthogonal to the model's estimate (so no constant refit captures it), and no knob costing under 5% FLOPs moves it by more than a tenth. Its source is the Gaussian-closure order of the birth formulas. Early-layer fixes wash out: exact content at layers 1-7 alone gives only 0.90x. Three more candidate sources were each removed exactly, and each moved the residual by under 1.5%:
 - the higher-order tail of the exact bivariate-Gaussian series;
 - the diagonal fourth cumulant;

- the cross fourth cumulant K31 fed into the birth formulas. Removing all history truncation (5.9x FLOPs) gains only 6%. What remains is the three-distinct-unit terms, whose cost grows as n^4 . Together these bound what any refinement of a cumulant-propagation state can achieve and indicate that further progress needs a different estimator family.

7. Negative results

Each with its decisive number. Monte Carlo with control variates: $\geq 100\times$ off (degree one carries only $\sim 19\%$ of the per-sample variance at depth 16; $\geq 99.9\%$ would be needed). Learned residuals on a Gaussian baseline: baseline $330\times$ worse than our pipeline. Two-hop weight-graph features: $+0.19\%$. Adaptive per-network calibration: a real per-network optimum exists ($+1.9\%$ on held-out units) but is unpredictable from weight or state features. Full nonlinear head retrains: the hidden layer feeds three heads and large steps break the linearisation. Gaussian-prior physics rebuild: $2.6\times$ worse heads-off, $1.27\times$ after recalibration. Calibrating 18 closure constants to the state error directly: at most -0.9% of the variance-error RMS, and worse final error. Deeper recursion and restricted products: official runtime failures (332423 wall-time timeouts; 332586 and 332604 (57/100) residual gate; 333107 residual gate on 13/100 networks, max 0.415 s). FLOP floor: all trip-costly rewrites together save $\sim 3\%$.

8. Discussion and future work

Moment closure with a compressed cumulant history beats sampling by hundreds of times in compute on this problem, and its remaining error is well localised: late layers, kink units, and the covariance/third-cumulant loop. The next step is not a better closure formula but a more consistent joint propagation of C and T near the kink - for instance carrying per-unit corrections of the third-cumulant birth computed from exact low-dimensional integrals, or a second-order (pair) cumulant history for kink units. On the systems side, metered client-server execution changes what "cheap" means: round trips and server-side allocations, not arithmetic, set the cost of fine-grained code.

9. Reproducibility, compliance and open-source release

All numerics run through flopscope; no hidden data, private seeds or grader internals were accessed, and no accounting was manipulated. Development used only the published networks. Sources, frozen plans, receipts and independent audits are kept in version control, and every candidate is identified by SHA-256. The exact source of #333755 (SHA-256 b1f91d17...473ceb, all constants embedded) is released under the MIT licence at <https://github.com/Hero988/arc-whitebox-estimation-2026-phase2>.

10. Use of AI tools

As the rules allow, most of the implementation, experiments and drafting were carried out by AI coding agents (Anthropic's Claude and OpenAI's Codex) working under my direction, with pre-registered gates and independent agent audits of every candidate. I take responsibility for the accuracy and completeness of this write-up. All numbers come from committed receipts of official grader runs or of local runs on published networks.