Compact Cumulant Propagation with Learned State Corrections

Matt Liston

ARC White-Box Estimation Challenge, Phase 2
Submission #337641
October 23, 2026

Abstract

We estimate the mean activations of width-1024, depth-16 random ReLU networks by propagating means, covariance and compact higher-order information, without sampling inputs at inference. The method builds on ARC's cumulant-propagation framework. Its principal design choices are a signed-pair representation of newly generated third-order information, selective compression of older factors, and fourth-order corrections carried through directions already present in those pairs. A learned recurrent model adjusts intermediate estimates. Its per-layer readouts are subsequently refitted on trajectories captured from the preceding estimator; those updates can change later trajectories.

Submission #337641 recorded a public adjusted score of **1.4006490** $\times 10^{-9}$, public final-layer MSE of **1.289** $\times 10^{-8}$, and mean compute utilization of **10.866%**. The archived grading round completed 100 networks with no failures. These are hosted records, not a new evaluation or the result of the separately designated private re-evaluation.

The research contribution we examine is how to preserve useful non-Gaussian information at affordable cost. Historical controlled experiments support several particular representations and state updates. They also show why better intermediate moment estimates, larger tensor models and more accurate local regressions need not improve the complete estimator. Learning is useful both as a correction mechanism and as a way to locate missing structure; we have not mechanistically explained all of its gains.

Submitted source · Reproduction · Research graph · Source-linked anatomy

1 Task, scope and relation to ARC's method

Using the submitted source's weight orientation and zero-based layer numbering, the target is

$$x \sim \mathcal{N}(0, I_{1024}), \qquad h_{-1} = x, \qquad h_l = \max(W_l^T h_{l-1}, 0), \quad l = 0, \ldots, 15. \tag{1}$$

The estimator returns all 16×1024 means $\widehat{m}_{l,i} \approx \mathbb{E}[h_{l,i}]$. Only the final layer enters the official error metric. For network s, let E_s be its final-layer MSE against the organizer's Monte Carlo reference, F_s its metered computation, and $B = 2^{41}$ the budget. The public score is

$$\frac{1}{N} \sum_{s=1}^{N} E_s \max\left(0.1, \frac{F_s}{B}\right). \tag{2}$$

Thus raw MSE and computation must be reported separately. Below 10% utilization, further compute reduction alone does not improve the multiplier. Phase 2 also imposes execution limits: 120 seconds per prediction, 0.4 seconds of residual control time, and five seconds for import/setup. Challenge; archived rules.

This writeup describes **one Phase 2 submission, #337641**. Earlier Phase 2 experiments explain its design; they are not represented as ablations of the final file. ARC's judging guidance emphasizes understandable mechanistic contributions, evidence for their improvement, and transparent LLM use. We separate structural changes, fitted corrections and arithmetic optimization accordingly.

Wu et al. [1] propagate cumulants through linear maps and nonlinearities using Hermite expansions. Their augmented variant retains additional information; factorization changes how the retained tensors are represented. The basic $K = 3$ method already includes a fourth-order trace; fourth-order information itself is therefore not our addition. The factorized basic $K = 3$ algorithm is a structural reference for Table 1, not a matched benchmark for our final estimator.

Table 1: Relationship to ARC's analytical methods and the submitted modifications.

Component	Inherited machinery	Modification in this submission
Third-order state	ARC transports factored tensors and treats repeated indices separately.	Signed cubic pairs represent new contributions, with residual/trace terms and fitted changes to their directions.
Compression	ARC's basic factorization preserves the output of its specified approximate algorithm.	A shared rank-384 basis introduces an additional lossy approximation for older factors; recent factors remain explicit.
Fourth-order effects	The basic method retains trace information; augmented variants retain more.	Covariance-shaped and pair-carried approximations feed selected fourth-order effects into the recurrence.
Learned updates	Analytical propagation provides the state and baseline moment estimates.	Fitted updates change means, covariance and new third-order contributions; per-layer readouts are refitted on corrected parent trajectories.

These are changes to a practical estimator, not claims of global novelty for factorization, closure or fourth-order information. Our lossy and fitted updates do not inherit ARC's theoretical guarantees. The source audit also records omissions in the organizer's factored augmented implementation relative to its unfactored counterpart. Historical experiments below support selected

modifications against their contemporary parents; they do not allocate the final score among components or establish an overall performance multiple over ARC.

Our observation that older third-cumulant factors concentrate in lower-dimensional subspaces was independently confirmed by Team 504aldo [2] in their public release on 10 September 2026 (UTC). Our proposal to compress older third-cumulant factors in a shared basis cited their public implementation as a reference. We implemented that experiment within our existing signed-cubic state, using squared-coefficient Gram weights and our own residual and support bookkeeping. Subsequent work changed the join schedule and basis construction.

2 The submitted estimator

ReLU changes the shape of an activation distribution: means and covariance alone generally do not determine the next layer's mean. We retain selected third- and fourth-order cumulants to describe departures from a Gaussian distribution. A **closure** approximates the next layer's statistics from this finite summary. The design problem is to retain useful information while avoiding full higher-order tensors.

One intermediate-layer update. The following stages summarize the active loop:

1. **Transport the retained information.** Transform means and covariance through the weights. At scheduled layers, compress older factors in the preceding layer's coordinates before transporting them; keep recent factors explicit.
2. **Estimate the ReLU moments.** Read third-order information and the retained fourth-order corrections from those factors. Apply the scalar and pair approximations to obtain new means, covariance and higher-order targets.
3. **Apply learned adjustments.** Correct covariance and means, with a corresponding covariance-centering adjustment. These corrected values enter subsequent propagation.
4. **Prepare the next layer's summary.** Construct new third-order contributions, gate and shift inherited factors, and update residuals and pair coefficients. A new contribution is called a **birth**; factors introduced at the same layer form a **cohort**.

The terminal layer only computes the scalar quantities needed for its means and applies its learned readout. The formulas below describe the retained model; masks, calibrated strengths and finite precision introduce further approximations documented in the anatomy.

2.1 State and linear transport

The initial state is $m = 0$, $C = I$, with zero higher-cumulant and learned state. Table 2 distinguishes the statistical state from the learned model's internal features:

The intended linear transformations are $\mu = W^T m$ and $C_{\text{pre}} = W^T C W$. Factors are transported through W^T. Actual execution includes selected-unit restrictions and mixed precision, so these equations describe the mathematical operation before those approximations. The covariance is a matrix state; it is not globally replaced by a rank-384 matrix.

Let $z = W^T h$ be the preactivation vector, with mean μ, covariance C_{pre} and centered value $\widetilde{z} = z - \mu$. Its scalar third and fourth cumulants are

$$k_{3,i} = \mathbb{E}[\widetilde{z}_i^3], \qquad k_{4,i} = \mathbb{E}[\widetilde{z}_i^4] - 3v_i^2, \qquad v_i = C_{\text{pre},ii}. \tag{3}$$

Table 2: Propagated state between ReLU nonlinearities.

State	Meaning
m, C	Mean vector and covariance matrix.
Signed cubic factors	Compact contributions to the third cumulant. Recent factors remain explicit.
Q, F and a small Gram core	Shared representation of older cubic factors.
A residual matrix and trace vector b	Repeated-index third-order information carried separately from the factors.
A fourth-order scale and local pair coefficients	Covariance-shaped fourth-order information, local kurtosis and norm-coupling corrections.
A 64-dimensional state per neuron	Learned features carrying information from earlier calculations; these are not individually identified cumulants.

The subtraction in k_4 removes the Gaussian fourth moment. For factor columns P_r and signed coefficients λ_r, the modeled third-order contribution is

$$T^{\text{factor}}_{ijk} = \sum_r \lambda_r P_{ir} P_{jr} P_{kr}, \qquad K^{\text{factor}}_{21,ij} = \sum_r \lambda_r P^2_{ir} P_{jr}. \tag{4}$$

Write $K_{21,ij} = \kappa(z_i, z_i, z_j)$ for the physical third-cumulant slice. Its modeled contribution adds the residual and the trace term $v_i b_j + 2C_{\text{pre},ij} b_i$ to the factors, with a separate diagonal update. The runtime array named K_{21} also contains folded fourth-order corrections off the diagonal. We denote that **effective read** by $\widetilde{K}_{21}$ below; it need not equal a physical third cumulant. State and assembly.

2.2 ReLU closure

Write $v_i = C_{\text{pre},ii}$, $\sigma_i = \sqrt{v_i}$ and $\alpha_i = \mu_i/\sigma_i$. With standard-normal density ϕ and distribution function Φ, the Gaussian ReLU mean is $m^G_i = \sigma_i\phi(\alpha_i) + \mu_i\Phi(\alpha_i)$. Our scalar approximation adds third- and fourth-cumulant responses:

$$\widehat{m}_i = m^G_i - \frac{k_{3,i}}{6\sigma^2_i}\alpha_i\phi(\alpha_i) + \frac{k_{4,i}}{24\sigma^3_i}(\alpha^2_i - 1)\phi(\alpha_i). \tag{5}$$

The scalar routine similarly estimates raw moments through order four, then centers them. The pair routine computes covariance and the third- and fourth-order slices needed for subsequent updates. Its Gaussian component uses a Hermite expansion in correlation: the general pair map retains powers through order three, and the first-layer specialization through order six. Diagonal quantities come from the scalar formulas. Variance floors handle near-degenerate inputs. These are truncated approximations, not exact formulas for arbitrary non-Gaussian joint distributions. Scalar and pair maps.

The baseline fourth-order approximation follows covariance geometry:

$$K_{ijkl} \approx \frac{r}{3}\left(C_{\text{pre},ij}C_{\text{pre},kl} + C_{\text{pre},ik}C_{\text{pre},jl} + C_{\text{pre},il}C_{\text{pre},jk}\right). \tag{6}$$

Here $K_{ijkl} = \kappa(z_i, z_j, z_k, z_l)$, so $k_{4,i} \approx r v^2_i$. The repeated-index slice is $K_{22,ij} = \kappa(z_i, z_i, z_j, z_j)$. After the nonlinearity the scale is estimated from its postactivation counterpart:

$$r_{\text{new}} = \frac{3\sum_{ij} K^{\text{post}}_{22,ij}}{(\operatorname{tr} C^{\text{post}})^2 + 2\sum_{ij}(C^{\text{post}}_{ij})^2}. \tag{7}$$

Layer-band calibration and the local corrections below modify this baseline. Its advantage is that correlations determine the shape of fourth-order dependence without storing a fourth-order tensor.

2.3 New third-order contributions and learned directions

After each ReLU we separate inherited third-order information from newly created information. A new contribution associated with unit j uses two shifted copies of a common direction u with opposite signs:

$$a\left[(u+te_j)^{\otimes 3}-(u+se_j)^{\otimes 3}\right]. \tag{8}$$

The pure $u^{\otimes 3}$ term cancels algebraically, leaving terms containing the unit direction e_j. This represents coordinate-associated cubic structure using two columns rather than a dense tensor. The residual and diagonal updates retain information that the factor construction does not match. The cancellation identity is exact; the complete fitted, clipped and truncated closure is approximate.

The common birth direction is also adjusted using analytical functions of C_{pre} and the effective read $\widetilde{K}_{21}$. Let $g_i = \Phi(\alpha_i)$, $d_i = \phi(\alpha_i)/\sigma_i$ and $q_i = \alpha_i/\sigma_i$. Three retained candidate directions have the forms

$$\begin{aligned} D^{(1)}_{ij} &= \tfrac{1}{2} d_i \widetilde{K}_{21,ij}, \\ D^{(2)}_{ij} &= -\tfrac{1}{2} g_i q_j \widetilde{K}_{21,ji}, \\ D^{(3)}_{ij} &= -\tfrac{1}{2} d_i q_j (C_{\text{pre},ij})^2. \end{aligned} \tag{9}$$

Their diagonals are zeroed; learned coefficients, projection adjustments and masks determine the applied action, with corresponding residual repair. The historical experiment fitted four directions; a later removal experiment left **three active directions** in the release. Active birth update; removal record; graph.

This is a restricted use of learning: select coefficients for specified state changes while retaining the mathematical state representation. It is more interpretable than an unrestricted output correction, but the fitted coefficients are not a derived closure law.

The Gaussian gate g in these candidate directions is distinct from the gate used to carry inherited third-order information. The latter includes skewness, kurtosis and a second-order skewness term. The birth density also receives a skewness correction, and the trace vector has a correction tied to changes in the fourth-order scale. These updates alter what survives the nonlinearity before the next linear transport. Corrected transport gate; active birth and trace updates.

2.4 Low-rank compression of older factors

A low-rank matrix approximation represents many columns through a smaller set of shared directions. For older cubic factors we use

$$P \approx QF, \qquad Q \in \mathbb{R}^{1024\times 384}, \qquad W^T P \approx (W^T Q)F. \tag{10}$$

The shared dimension is **384**; it is neither the number of neurons nor the number of retained cubic terms. All joined signed columns remain represented by coefficients. Recent contributions remain explicit.

At three scheduled joins, the basis is built from a weighted Gram matrix using squared signed-cube coefficients as nonnegative weights. A deterministic Nyström sketch selects high-weight columns to construct the basis; the **complete** joined block is then projected into it. A small Gram core summarizes older contributions for later joins. The savings come from transporting a common basis once, while retaining each cubic term's coefficients. Construction and schedules; graph.

The joins occur at layers 7, 10 and 13, absorbing birth cohorts 0–2, 3–5 and 6–8. The Nyström construction keeps 1,536, 1,024 and 1,024 high-weight columns at the three joins before the complete block is projected; these counts were set by paired measurements on 32 held networks (column counts).

The compression metric is a practical surrogate. It does not certify preservation of signed cancellation, tensor queries or final expected activations. Randomized approximation results such as Halko, Martinsson and Tropp [7] provide background for the research comparisons; they are not guarantees for this deterministic, weighted construction.

2.5 Fourth-order information carried by existing pairs

The same two directions that represent a third-order contribution can support selected fourth-order calculations. This is the central representation choice: additional coefficients and element-wise operations reuse directions whose transport is already paid for. The following constructions retain particular components of the fourth cumulant; their strengths and age schedules are calibrated.

Fourth-order column shifts. A retained fourth-order contribution influences newly generated third-order information at the next ReLU. The column-shift correction applies this influence as a first-order change to recent transported columns, with matching trace and diagonal adjustments in the residual. This avoids the extra large tensor contractions required by earlier proposals. The archived derivation and source establish the intended approximation; the historical whole-estimator comparisons below measure its usefulness. Derivation and measurements; implementation.

Local fourth-order variation. From a pair $p_\pm = u + t_\pm e_j$, reconstruct its two directions as

$$Y = \frac{t_+ p_- - t_- p_+}{t_+ - t_-}, \qquad H = \frac{p_+ - p_-}{t_+ - t_-}. \tag{11}$$

This reconstruction is algebraic. Interpreting the directions as transported activation structure remains approximate after gates, learned direction changes and shifts. The pair's modeled contribution to the scalar fourth cumulant is

$$4S_1 Y_i^3 H_i + 6S_2 Y_i^2 H_i^2 + 4S_3 Y_i H_i^3 + S_4 H_i^4. \tag{12}$$

The coefficients come from the contribution's ReLU moments. Summing selected recent pairs supplies neuron-specific fourth-order variation beyond the single covariance-shaped scale. A fitted multiple of v_i^2 is removed to reduce overlap with that existing component. Coefficient derivation; scalar calculation; graph.

Norm coupling. The statistic of interest is the fourth-cumulant contraction

$$N_{ij} = \sum_k \kappa(z_i, z_j, z_k, z_k) = \mathrm{Cov}(\tilde{z}_i \tilde{z}_j, \|\tilde{z}\|^2) - 2(C_{\mathrm{pre}}^2)_{ij}. \tag{13}$$

Here C_{pre}^2 denotes the matrix product $C_{\mathrm{pre}} C_{\mathrm{pre}}$. The statistic measures how a pair's centered product varies with the total squared centered norm, beyond the dependence implied by Gaussian

covariance. The identity defines the target; the update used to approximate it is a separate modeling choice. A pair contribution can be stored as

$$B_j = [Y\ H] M_j [Y\ H]^T, \qquad M_j \in \mathbb{R}^{2\times 2} \text{ symmetric.} \tag{14}$$

For example, the three entries of M_j weight YY^T, $YH^T + HY^T$ and HH^T. Transporting the existing pair therefore also transports this modeled matrix contribution. This replaced a representation that needed separate transported vectors. The implemented approximation removes fitted covariance/mean components and uses the remainder to modify the effective third-order read.; pair construction and evidence; graph.

An effective read, not a new cumulant identity. Define $K_{31,ij} = \kappa(z_i, z_i, z_i, z_j)$. The local quartic also supplies a modeled contribution to this mixed fourth-order slice. After subtracting contractions already represented by norm coupling, call its remaining component $L^{(4)}_{ij}$. The reference first-order fold is

$$\Delta\widetilde{K}_{21,ij} = -\frac{\mu_i}{3v_i} L^{(4)}_{ij}. \tag{15}$$

The release combines this local contribution with the norm-coupling fold, using fitted strengths, masks and subtraction terms. It restores the cubic diagonal separately. These prescriptions approximate fourth-order responses through a calculation already used for third-order information; they do not identify the physical K_{21} with a fourth-order statistic. The subtracted component and the local remainder are called second- and fourth-chaos components in the archived derivation. Local mixed-slice derivation; effective read.

2.6 Learned state corrections and per-layer readouts

Learning adjusts the statistical summary during propagation. A recurrent model reads moments, gate probabilities, covariance summaries and weight statistics, together with information from preceding layers. It predicts a mean adjustment δ and carries learned features forward. A separate pair model corrects covariance at layers 8–14; fitted direction coefficients change how new third-order information is represented. The network being estimated is fixed. Training changes the estimator's correction models, whose coefficients are embedded for inference.

The mean adjustment is scaled to the preactivation standard deviation. Layer 0 receives no learned mean correction. At intermediate layers, δ is clipped to keep $m + \delta$ nonnegative. Holding the raw second moment fixed while changing the mean requires

$$D = m\delta^T + \delta m^T + \delta\delta^T, \qquad C \leftarrow C - D. \tag{16}$$

The release adds calibrated layer-dependent compensation, so its full update does not exactly preserve that raw moment. Corrected means and covariance enter subsequent propagation. This is why improving a local fit need not improve the final answer: the correction also changes the inputs to later approximations. Active update.

Table 3 summarizes successive fits, rather than one final train/test partition. The stages below establish specific targets and fitting procedures; a complete mapping from every training run to every embedded coefficient in #337641 has not been reconstructed. Later joint refits changed several modules, so earlier splits should not be read as final-weight provenance.

The training-record account gives the objectives, selection rules, later joint fitting and remaining provenance limits.

Table 3: Documented training stages and their scope.

Learned component	Target and documented fitting stage
Mean correction	A recurrent mean-model refit used Monte Carlo mean targets on 160 reused development networks, with 40 for validation. It minimized final-layer MSE plus a smaller intermediate-layer loss, with the other modules frozen. Later joint fits also updated the encoder.
Birth directions	Coefficients of prescribed directions were trained through downstream mean error. The global fit used 160/40 networks; a conditional-head pilot used 20/4 with encoder and global coefficients frozen. These describe historical fits, not a fresh test of final coefficients.
Covariance correction	The final width-eight compression fit matched a frozen teacher's covariance corrections on four training networks, with three validation networks unused for selection. The checkpoint was selected by lowest recorded training loss. This fit used teacher outputs, not network-truth targets; it does not establish the teacher's full training history.
Per-layer readouts	The final fit predicted mean residuals using 90 training networks and two Monte Carlo target streams. An earlier 72/18 split evaluated feature choices. The encoder, birth and covariance arrays were fixed during this readout fit.

The **final revision of the per-layer readouts, at layers 1–15**, was fitted on captures from **90 training networks along the preceding estimator's own corrected trajectories**. It predicts remaining mean errors from frozen learned features using constrained, regularized linear fits. The recurrent encoder is unchanged by this export. Updating intermediate readouts can change subsequent states, so this is a one-step refit on frozen parent captures, not a guarantee that the final estimator visits exactly those states. Fit and export; graph.

The fitted design has 961 columns per layer: the 64-wide state, a constant, both 128-wide hidden activations, their elementwise squares, each hidden activation multiplied by the gate, and the product of the two hidden activations. The archived fit uses ridge 0.03 and damping 0.5, with constraints and residualization against the existing readout directions (fit code; measurements). The exported update is scaled by a strength constant baked into the float32 coefficients: 1.35 in #337641 and 1.5 in #337619, which is the only difference between the two designated files apart from the removal of inert code from #337641 (strength ladder).

The anatomy records model dimensions, feature families, auxiliary rank reductions and exported fit settings. Those implementation details specify the released model; they do not by themselves explain the learned gains mechanistically.

2.7 Arithmetic and execution

Covariance transport and factor products dominate computation. We use Strassen [3] and Winograd [4] recursion, Karstadt–Schwartz [5] alternative bases and grouped products based on Smirnov's bilinear scheme [6], together with symmetry and product reuse. These have established mathematical antecedents. Smirnov's rank counts bilinear products in an exact multiplication scheme; the rank-384 basis above instead makes a lossy approximation to factor geometry.

The implementation also omits selected weak-response rows and uses mixed precision. These change the numerical estimator and require empirical checks. All per-network numerical work is issued through the metered array interface. The final layer avoids a full covariance, and fixed setup and routing choices control runtime. The anatomy preserves the thresholds, precision choices and source-order details needed to identify the active computation.

3 How research directions were chosen

We used three questions to decide whether to develop an idea: **Is useful information missing? Can it be represented and read within the budget? Does the complete corrected recurrence improve?** A truth-supplied diagnostic could justify an experiment, but could not establish a deployable gain. A cheap representation could justify a complete test, but not a claim that its omitted information was harmless.

The comparisons in Table 4 show how those questions changed the design. Each row uses its contemporary Phase 2 parent and evaluation protocol. **The percentages are not additive and are not leave-one-component-out measurements of #337641.** "Native" denotes the metered implementation, as distinct from a research simulator; local panels are not official hosted scores.

Table 4: Selected Phase 2 research decisions and archived measurements.

Decision and evidence	Networks and measured change	Interpretation and boundary
Covariance-shaped fourth-order scale. Record.	**Networks:** 2 reused development networks. **MSE:** 4.0824 / 4.3204 to 3.2346 / 3.2068, all $\times 10^{-8}$.	An analytical change helped before adding learning. This is not a population estimate or a comparison with every ARC variant.
Learned third-order construction. Native comparison.	**Networks:** 40 development-validation networks; 38 wins. **MSE:** −5.19%. **Adjusted:** −4.89%.	Four fitted directions, with response heads frozen. Later pruning leaves three active directions in the release.
Joint compression of older factors. Paired comparison.	**Networks:** 14 with independent reference draws. **MSE:** +2.04%. **Compute:** −10.51%. **Adjusted ratio:** 0.91228.	A useful accuracy–cost tradeoff, but not a pristine network holdout. This candidate failed cross-backend parity; later implementations were required.
Fourth-order column shifts. Experiment.	**Networks:** 24 held out; 24 wins. **MSE:** −3.54%. Separate native confirmation: −3.83%.	Reusing existing columns avoided new large tensor contractions; extra-carrier proposals had failed cost break-even.
Local fourth-order variation. Experiment.	**Networks:** 24 held out; 18 wins. **MSE:** −1.41% on top of column shifts. **Compute:** +0.115%.	Cheap local detail helped. Broader age extensions required separate selection.
Norm coupling on existing pairs. Native comparison.	**Networks:** 10 fresh. **MSE:** −7.93%. **Compute:** +0.876%. **MSE × compute:** approximately −7.16%.	The representation made this modeled contribution affordable. The product is a local tradeoff summary, not a hosted adjusted-score measurement.
Compressed learned contractions. Paired comparisons.	**Networks:** fresh panels of 16 and 8. **Adjusted:** −0.577% and −0.560%, respectively.	Preserving the constant component allowed a small approximation loss to be exchanged for cheaper correction.
Refitted per-layer readouts. Complete-stack comparisons.	**Networks:** Linux panel of 32, Mac panel of 32, fresh panel of 12. **Adjusted:** −0.572%, −0.538%, −0.384%.	These comparisons include runtime settings as well as the readout update, not an isolated contribution to the final score.

Two negative directions are particularly informative about low rank. Replacing 6,144 signed cubic atoms with 384 orthogonal cubes preserved their shared linear span but left **95.4%** of squared query signal unexplained in the specified diagnostic. The experiment was motivated by

Kolda's orthogonal decomposition [9], whose structural assumptions were not presumed to hold here. A subsequent 128-triple nonorthogonal fit, informed by Kolda's optimization approach [10], also failed its prescribed fidelity gate. A low-dimensional column space therefore did not make these particular cubic representations adequate. Neither failure is a tensor-rank lower bound. Orthogonal test; nonorthogonal test.

A two-block rank-384 Krylov construction, motivated by Musco and Musco [8], worsened pooled adjusted error by **4.16%** on two reused development networks after its actual construction cost was included. More accurate generic matrix approximation is not automatically the right objective for a signed cumulant representation. Separately, a correlation-resummed pair calculation using Price's theorem [11] worsened pooled adjusted error by **9.69%** on two development networks; its precision and quadrature also differed, so it did not isolate correlation truncation alone. Krylov experiment; pair-map experiment.

The estimator-development graph connects these choices to their surrounding research. Its edges record documented dependencies and development milestones, not a proved shortest path or an assertion that every adjacent node is an immediate source-file parent. The graph opens in a browser; each work folder contains browsable code and evidence on GitHub. The self-contained `graph/index.html` also works offline.

4 What learning tells us about missing mechanisms

Learning can help focus mechanistic research. A correction that generalizes across networks shows that available features predict some remaining error. It does not uniquely identify that error as an omitted physical term: the correction may approximate missing cumulant interactions, compensate for truncation, or exploit biases created by other approximations.

Our birth-direction experiment is a concrete example of using learning to investigate a mechanism. Truth-supplied joint-third diagnostics suggested local opportunity. We then changed the creation of third-order state, held the response heads and factor count fixed, fitted coefficients of explicitly specified directions, and evaluated the complete recurrence. The protocol included zero initialization and null-action/restoration checks. This supports the usefulness of that restricted action; it does not identify every coefficient's independent causal contribution.

The counterexamples matter just as much. A centered-feedback fit reduced held one-step MSE by **20.95%**, yet improved two complete recursive runs by only **1.72% and 1.69%**. A later learned joint third/fourth-moment correction improved intermediate diagnostics but worsened native final MSE by **49.56%** on two reused networks. A fourth-order-strength adjustment that helped the preceding estimator stopped helping when combined with the refitted per-layer readouts. These results are consistent with coupled and compensating errors; they do not prove a unique explanation for each failure. Feedback experiment; joint-moment experiment; readout interaction.

One diagnostic suggests a specific next target. On a single October 2 development network, replacing covariance diagonals with 32-million-sample estimates improved final MSE by **1.06%**; replacing the whole covariance improved it by **13.84%**, while worsening all-layer MSE by **16.9%**. This points toward off-diagonal structure and objective-dependent error propagation in that parent. It is not a universal ceiling, a noise-free intervention or a result for the final release. Covariance intervention.

Proposed next experiments, not completed results:

1. Freeze #337641 and predefine paired network splits. Repeat single-family and joint interventions on covariance, third-order residuals and fourth-order terms, with independent reference estimates to assess sampling noise.
2. Fit small diagnostic predictors to nested, explicitly defined feature families. Test whether each adds useful information in the complete recurrence, rather than interpreting feature importance or one-step fit alone.
3. Replace successful learned actions with compact analytical formulas that respect neuron-permutation symmetry, appropriate scaling and the Gaussian limit. Compare the original correction model frozen against a version refitted for the changed trajectory.
4. Test compression directions by their downstream influence on expected activations, alongside their Gram energy. A small discarded direction may matter after further weights and nonlinearities; a large direction may mostly cancel against another signed term.
5. Require retained gains in final MSE at measured cost, inspect errors at every layer, and then test other widths, depths and input distributions before claiming broader applicability.

The historical fourth-order and pair-form experiments support the usefulness of several explicit corrections on the tested estimators. They do not yet account for the full learned correction. Turning that remaining predictive advantage into a smaller, transferable statistical update is an open research objective.

5 Results, limitations and reproduction

The archived hosted record identifies the exact submitted file and reports:

Table 5: Archived hosted results for submission #337641.

Quantity	Submission #337641
Official public adjusted score	$1.4006490071107744 \times 10^{-9}$
Public final-layer MSE	1.289×10^{-8}
Public all-layer MSE	5.165×10^{-9}
Mean compute utilization	0.1086588753
Graded / failed networks	100 / 0
Prediction wall time, mean / maximum	98.678 / 109.745 seconds
Residual time, mean / maximum	0.336 / 0.363 seconds

The grading round covered 50 public and 50 private networks; the reported public score is not the separately designated private re-evaluation result. It is the average of per-network adjusted scores, not the product of the aggregate MSE and aggregate utilization.

We have not established a matched end-to-end performance multiple over the organizer's paper, nor a complete final-source ablation allocating the result among mechanisms. Historical gains use different parents, learned coefficients and panels. Some local studies reuse development networks or subtract estimated Monte Carlo noise; their protocols delimit the claim. The release is specialized to the challenge geometry, and success here does not establish performance on trained networks, different inputs or rare-event probabilities. Several constants, age schedules and learned functions are calibrated rather than derived.

For exact source reconstruction and execution, start with estimator/REPRODUCE.md. Run the following from the **arc-whitebox-estimation-challenge repository root**, using the environment installed as described there:

```
_build/reproduce-venv/bin/python -B tools/reproduce_estimator.py
_build/reproduce-venv/bin/python -B tools/run_estimator.py \
  --submission 337641 --estimator _build/reproduced/337641/estimator.py
```

The first command reconstructs both archived release sources and checks their hashes. The second runs #337641's seeded inference, not a hosted score reproduction. The guide separately provides public-target evaluation and explains timing differences. The submitted #337641 source SHA-256 is `5a4c15405c4462703aeead1bb6ccbc9b98ccf6ba2b5974ce23713303c87d594d`.

Inference coefficients are embedded. Repeating training additionally needs captures, targets, fitting dependencies and the original environment; some large assets are not distributed in Git. Data availability states those limits. The copied research programs preserve their original path assumptions, so their presence should not be mistaken for a turnkey retraining package. This writeup was prepared by source inspection and review of archived measurements; no new estimator benchmark was run for it.

6 Assistance and attribution

Matt Liston directed the research with extensive assistance from Anthropic Claude and OpenAI Codex agents. Agents proposed methods, wrote code, ran experiments and drafted notes. OpenAI Codex drafted this writeup from source inspection and archived evidence. No LLM or external service is called during estimator inference.

Source inspection establishes the implemented operations; archived experiments support the reported effects. We have not validated every mechanistic explanation or reconstructed every training stage. The archived disclosure gives contemporary context, superseded where necessary by the final-source description and completed hosted record above.

References

[1] **Wilson Wu, Victor Lecomte, Michael Winer, George Robinson, Jacob Hilton and Paul Christiano.** *Estimating the expected output of wide random MLPs more efficiently than sampling.* arXiv:2605.05179v2, 2026. Paper; organizer implementation. Foundation for cumulant propagation, Hermite closure and structured higher-order representations.

[2] **Team 504aldo.** *whest-p2-cumulant-k3: Phase 2 public implementation.* Reviewed commit `18c17e2d7a9aeacd399cfc2c6b571e4e16dbfb45`, 2026. Pinned repository; public release, 10 September 2026 UTC. Public implementation consulted and cited in our proposal to compress older third-cumulant factors in a shared basis. This is a participant implementation, distinct from ARC's code.

[3] **Volker Strassen.** "Gaussian elimination is not optimal." *Numerische Mathematik* 13, 354–356, 1969. doi:10.1007/BF02165411. Recursive matrix-multiplication foundation.

[4] **Shmuel Winograd.** "On multiplication of 2 × 2 matrices." *Linear Algebra and its Applications* 4(4), 381–388, 1971. doi:10.1016/0024-3795(71)90009-7. Lower-addition multiplication variant.

[5] **Elaye Karstadt and Oded Schwartz.** "Matrix Multiplication, a Little Faster." *Proceedings of SPAA*, 101–110, 2017, doi:10.1145/3087556.3087579; expanded version, *Journal of the ACM* 67(1), 1–31, 2020, doi:10.1145/3364504. Alternative-basis arithmetic adopted in the Phase 2 kernel experiment.

[6] **Alexey Vladimirovich Smirnov.** "The bilinear complexity and practical algorithms for matrix multiplication." *Computational Mathematics and Mathematical Physics* 53, 1781–1795, 2013.

doi:10.1134/S0965542513120129. Coefficients were obtained from **Alexandre Sedoglavic's Fast Matrix Multiplication Catalogue**, Université de Lille/CRISTAL, $3 \times 3 \times 6$, 40-product entry; local source audit. Our contribution concerns its integration and execution, not the underlying bilinear identity.

[7] **Nathan Halko, Per-Gunnar Martinsson and Joel A. Tropp.** "Finding Structure with Randomness: Probabilistic Algorithms for Constructing Approximate Matrix Decompositions." *SIAM Review* 53(2), 217–288, 2011. arXiv:0909.4061. Sketching background explicitly considered in the Phase 2 comparisons; its randomized guarantees are not claimed for our joins.

[8] **Cameron Musco and Christopher Musco.** "Randomized Block Krylov Methods for Stronger and Faster Approximate Singular Value Decomposition." *Advances in Neural Information Processing Systems* 28, 1396–1404, 2015. arXiv:1504.05477. Motivated the tested two-block alternative, which was rejected.

[9] **Tamara G. Kolda.** *Symmetric Orthogonal Tensor Decomposition is Trivial.* arXiv:1503.01375, 2015. Paper. Motivated the orthogonal-cube diagnostic, not a structural assumption about our tensor.

[10] **Tamara G. Kolda.** "Numerical Optimization for Symmetric Tensor Decomposition." *Mathematical Programming* 151, 225–248, 2015. doi:10.1007/s10107-015-0895-0; arXiv:1410.4536. Background for the tested nonorthogonal fitting alternative.

[11] **Felix Voigtlaender.** "A General Version of Price's Theorem: A Tool for Bounding the Expectation of Nonlinear Functions of Gaussian Random Vectors." *Journal of Theoretical Probability* 34, 1474–1485, 2021. doi:10.1007/s10959-020-01017-w; arXiv:1710.03576. Gaussian differentiation used in the correlation-resummed pair-map experiment.