ARC SIMPLE in FlopScope: a cumulant-propagation implementation

Franklyn Rosario, MBA-ITM | AIZOYA

ARC White-Box Estimation Challenge 2026, Phase 2 | Submission ID: #337649

October 25, 2026

Abstract

We describe a FlopScope implementation of ARC SIMPLE that estimates expected activations in random ReLU networks by propagating means, covariance, structured third-order sources and a radial fourth-order state. The implementation retains full repeated-index residuals and transports the radial metric through linear layers. On eight known public networks, the implemented SIMPLE configuration reduced mean compute-adjusted error by 91.22% relative to our BASE configuration while using 3.29% more FLOPs. The selected submission's recorded public grading completed all 50 networks, with adjusted score 2.681e-8 and final-layer MSE 3.564e-8. The practical result is a compact realization of an inherited method at about 75% of the compute budget. We explain its state, layer update, development choices and observed limits.

1. Problem and development

Given a random multilayer perceptron's weights, the task is to estimate expected post-ReLU activations under the challenge input distribution. The recorded Phase 2 evaluation uses width n = 1024, depth 16 and budget B = 2^41 FLOPs. The estimator returns a mean vector for every layer, including the final post-ReLU output. [1, 3]

R007, R008 and the other R-numbers below are our internal candidate or study IDs. The recorded development progressed from R007 BASE, which propagates structured third-order information without a fourth-order state, to R008 SIMPLE, which adds ARC's radial fourth-order closure. The comparison asks whether retaining that inherited component improves the implemented chain enough to justify its cost. Both versions retain full third-order residuals. This makes BASE-to-SIMPLE the relevant combined configuration comparison, rather than an ablation of residual storage. R044 and R051 subsequently revalidated the frozen implementations under the updated accounting environment. [3, 4]

Resource failures constrained development. Historical R008 repeats exceeded the 0.4-second residual cap on three of four fixtures. A separate candidate, R020, retained a hard runtime STOP after a recorded 0.625258-second residual; it is historical context, not part of the selected estimator. Later R008 successes and public grading added positive evidence without removing those failures. The engineering lesson was to distinguish accuracy, charged computation and runtime reliability: success on one axis did not certify the others. The final notice repair corrected attribution and license text while preserving R008's numerical body. [3-5]

2. State and linear propagation

Each layer's state consists of mean μ, covariance C, a list of third-order sources, and radial fourth-order core c with metric M. Initialization is μ = 0, C = I, an empty source list, c = 0 and M = I. Weights W have input-by-output orientation. Linear propagation is:

$\mu \leftarrow W^T\mu$; $C \leftarrow W^TCW$; $M \leftarrow W^TMW$.

Each source contains matrices P, A, Z and vectors s, e, h, corresponding to code fields p/a/z/s/e/w2. Columns index birth coordinates and rows current coordinates. P transports direct source legs; A transports covariance-hub legs; Z transports the full repeated-index residual repair. The vectors s, e, h retain birth coefficients. Linear propagation multiplies each of P, A, Z by W^T. [3]

At birth, P = I. Let u be the ReLU gain, v its second Wick coefficient, and C_off the incoming covariance with diagonal removed. The other birth quantities are:

$A = \mathrm{diag}(u)\, C_{off}$; $h = v$;
$s = K_{3,\mathrm{diag,new}} - K_{3,\mathrm{diag,old}}$;
$e = 2g_1 - 2\mu' \odot u$;
$R = K_{21,\mathrm{new}} - K_{21,\mathrm{old}} - A^2\,\mathrm{diag}(v) - \mathrm{diag}(e)\, C_{off}\, \mathrm{diag}(u)$;
$Z = 3R^T$.

Here ⊙ and matrix squares denote elementwise products; μ′ is the outgoing mean and g_1 the Gaussian ReLU mean. The “old” terms are incoming third-order slices multiplied by the corresponding ReLU gains. K_{21} is the repeated-index third-cumulant slice. Z retains the residual after transported history, covariance-hub and separable terms are accounted for. This defines the fields operationally, without treating the parametrization as a new cumulant method. [3]

The helper reconstructs required third-order slices from these sources. For one source, using columnwise vector broadcasting, define:

$Q = P \odot s + 3A \odot e + Z$;
$L_a = 2(A \odot P) \odot h + P^2 \odot e$;
$L_p = A^2 \odot h + P^2 \odot s/3 + 2(Q \odot P)/3$;
$T = \Sigma_{\mathrm{sources}} [L_aA^T + L_pP^T + P^2Z^T/3]$.

The diagonal of T gives D_3, the marginal third cumulant; its off-diagonal part gives D_{21}. This is the source-slice calculation performed by the helper for the stored representation. [3]

3. ReLU closure and radial fourth order

For incoming variance $\sigma^2 = \mathrm{diag}(C)$, define $\alpha = \mu/\sigma$, φ = normal PDF(α), and Φ = normal CDF(α). The code evaluates ReLU coefficients and the Gaussian first moment as:

$u = \Phi$; $v = \varphi/\sigma$; $w_3 = -\alpha\varphi/\sigma^2$;
$w_4 = (\alpha^2 - 1)\varphi/\sigma^3$; $g_1 = \mu\Phi + \sigma\varphi$.

It also evaluates Gaussian truncated moments through degree four and mixed partial-moment slices. The retained radial state supplies:

$K_{4,iiii} = c\, M_{ii}^2$;
$K_{4,iijj} = c(M_{ii}M_{jj} + 2M_{ij}^2)/3$, for $i \neq j$.

The mean update makes the added state's role explicit:

$\mu' = g_1 + D_3 \odot w_3/6 + K_4,\mathrm{diag} \odot w_4/24$.

The implementation constructs marginal and mixed partial moments, converts them to ordinary cumulants, symmetrizes outgoing covariance and clamps its diagonal to nonnegative values. It projects the outgoing fourth cumulant to a scalar core and resets the metric:

$c' = 3[\Sigma_i K_{4,iiii,out} + \Sigma_{i\neq j} K_{4,iijj,out}] / [n(n + 2)]$; $M' = I$.

Metric transport is exact for the retained representation, while the nonlinear closure retains a projected fourth-order state. The code applies the full closure after every weight matrix, including the final layer. Existing P, A, Z receive rowwise gain u before the new source is appended. The numerical path uses analytic moments and source transport, with no fitted correction table or sampling loop. It accepts but ignores the budget argument: the work pattern does not adapt to unused budget. [3]

4. Evaluation and findings

Historical local comparison

R044 and R051 used known public Mini rows 0-7, in two separately frozen batches. Each row ran R007 then R008 in fresh official SubprocessRunner workers, without retries or candidate changes. The external evaluator used installed scoring expressions and the budget-adjustment helper. Local adjusted error was final-layer MSE × max(0.1, FLOPs/B). The environment was Python 3.10.21, NumPy 2.2.6, FlopScope 0.12.3 and WhestBench 0.16.2. [4]

- Mean adjusted error: BASE 3.115305358718265e-7; SIMPLE 2.7358292669885972e-8.
- SIMPLE had lower adjusted error on all eight pairs. The ratio of arithmetic means corresponds to a 91.22% reduction.
- FLOPs per network: BASE 1,601,270,342,623; SIMPLE 1,653,900,429,294, a 3.29% increase.

The implemented SIMPLE configuration substantially improved on the implemented BASE control on this public cohort. The added cost was modest relative to the measured error reduction. These experiments preceded the notice repair; static byte and AST comparisons bridge their numerical implementation to the selected repaired package. [3, 4]

Recorded public grader outcome

The recovered grading photograph displays 50/50 public networks scored, 50 completed, zero failed, adjusted score 2.681e-8, final-layer MSE 3.564e-8 and all-layers mean MSE 1.43e-8. A separate selection photograph visibly identifies #337649 as GRADED and SELECTED. These are rounded displayed results, distinct from the eight-case local comparison. [5]

A billing consistency check is informative: public adjusted score / public final MSE ≈ 0.752245, while local SIMPLE FLOPs / 2^41 ≈ 0.752107. The close values are consistent with similar charged compute. The public ratio is an MSE-weighted adjustment multiplier, and the displayed scores are rounded: this is an inference about consistency, not authentication of server-side code or an independent per-case billing measurement. The local count establishes 75.2107% budget use. [4, 5]

5. Attribution, limitations and lessons

ARC's factorized cumulant propagation supplies the K3 method, Wick/ReLU expansion, partial-moment conversion, SIMPLE radial K4, metric transport and radial projection. We follow ARC commit `93d091a4c26c042bfffa28f2e76a81bc0aba94bb`. Shared-helper source-leg regrouping is informed by 504aldo's whest-p2-cumulant-k3 at `1a4083f89184d6b1cc55d2e55a5a561d2ae50a8d`. AIZOYA's work is the concrete FlopScope/WhestBench port, explicit source storage, variance policy, packaging and evaluation controls. The archive retains the ARC, 504aldo and AIcrowd MIT notices. [1-3]

The BASE-to-SIMPLE gain contains inherited mechanisms. It neither isolates a new AIZOYA scoring mechanism nor establishes originality from low textual overlap. A configured upstream comparison must align weight orientation, final activation, metric options, dtype and variance handling. Tiny-case correspondence supports the method-family mapping; broad equivalence to unchanged upstream defaults remains unproved. [3, 5]

The local networks were already known public examples. Private performance, final ranking and prize eligibility remain unknown. The public aggregate was read from original images rather than recomputed from an official per-network export. Historical numerical drift between local runs has no isolated cause. [4, 5]

Incoming nonpositive minimum variance raises an error. Small positive variance is not floored as in the pinned ARC coefficient implementation. Symmetry and a nonnegative diagonal do not guarantee positive-semidefinite covariance or finite outputs; the prediction path lacks comprehensive finiteness validation. Remote validation was blocked by IPC bind denial and handshake timeout, completing zero of three planned predictions. Bounded local same-input persistence succeeded, but a worker return code of −9 after close left graceful shutdown and its cause unresolved. These observations support bounded operation rather than a general lifecycle guarantee. [3-5]

The practical lesson is that an explicit FlopScope realization of inherited SIMPLE mechanics can reproduce a large BASE-to-SIMPLE improvement within the recorded FLOP budget. A second lesson concerns unused budget: the implementation uses about three quarters of it but has no policy for allocating the remainder. That observation describes the selected code, rather than a tested improvement or replacement.

6. AI assistance

AI assistance was material to code and writing. Franklyn Rosario confirmed that AI assistants generated or substantially revised the submitted R007/R008 code. OpenAI Codex generated and revised this technical write-up under his direction, including evidence recovery, static comparisons and document production. Repository records identify Codex engineering and OpenAI-assistant review; the notice-repair report also records an advisory Claude attachment review. Exact model attribution for individual numerical-code passages and contribution percentages are unavailable. Validation remains bounded as described above; comprehensive mathematical equivalence and general reliability remain outside it. [3, 5]

Appendix A. Artifact and reproducibility record

The receipt names r008-notice-repaired.tar.gz. Its recovered local archive is 6,696 bytes, SHA-256 `eae86acf715ad44d1a75b94573bf0d43b896d4fdf28bbd04810bb0501a2b4597`. All eight manifest-listed files match; the manifest is the ninth member. Entry point estimator.Estimator subclasses R008 without overriding behavior. Both repaired numerical module bodies match their originals byte-for-byte after leading docstrings; the corresponding ASTs also agree. [3]

Historical local data were pinned to dataset commit `aa99830fdc09fad15407b10e8e3459d3e18bba0a`, with software versions listed in Section 4. Code, evidence, source-field mapping, notices and the claim table are retained in the project reproducibility package and can be provided on request. This PDF assumes no access to a public AIZOYA repository. The receipt/image chain supports association with the selected entry; exact server-side tarball identity and per-network result export remain unauthenticated. No estimator execution or candidate change was performed for this revision. [3-5]

References and evidence availability

[1] Alignment Research Center, mlp_cumulant_propagation, pinned implementation of factorized cumulant propagation and SIMPLE. https://github.com/alignment-research-center/mlp_cumulant_propagation/tree/`93d091a4c26c042bfffa28f2e76a81bc0aba94bb`

[2] 504aldo, whest-p2-cumulant-k3, source-leg regrouping lineage. https://github.com/504aldo/whest-p2-cumulant-k3/tree/`1a4083f89184d6b1cc55d2e55a5a561d2ae50a8d`

[3] AIZOYA frozen R008 package and R058 notice-repair record, October 21, 2026. Includes numerical modules, entrypoint, manifest, attribution, full notices, static parity record and recorded validation. Retained project evidence; available on request.

[4] AIZOYA R044/R051 matched public evaluation reports, October 18; R055 attribution audit, October 20, 2026. Includes per-row prediction records, scoring protocol, recorded summaries and pinned environment/data identities. Retained project evidence; available on request.

[5] AIZOYA receipt/public-result record for #337649; original grading and selection photographs, October 21; advisory attachment review and October 25 owner AI-assistance confirmation. Photographs support displayed public results and selection; reviews are advisory. Retained project evidence; available on request.