ARC WHITE-BOX ESTIMATION CHALLENGE 2026

Auditable Compute Reductions in a Factored Cumulant Estimator

Phase 2 technical write-up: implementation, measured gains, failed experiments, and limits

Ömer Kiraz · omer_kiraz · Submission **#337188**

3.124e-9	50 / 50	0.573%
Reported public score	Public networks successful	Public score reduction vs #336974

Abstract

White-box estimation is a joint accuracy-and-compute problem: a numerically better prediction can lose if obtaining it costs too much. This report documents a narrowly scoped improvement to an existing, openly released cumulant estimator. The submitted approach inherits a factored third-cumulant chain, fourth-cumulant regeneration, pruning, and a learned correction from credited upstream work. Our final intervention changes the maximum recursion depth of eligible Strassen products and regenerates their static scratch-buffer and view-preparation plan; it leaves the estimator's mathematical update rules, ranks, pruning schedules, and learned parameters unchanged.

In the participant-provided AIcrowd public results, submission #337188 achieved an adjusted score of **3.124×10^{-9}**, against **3.142×10^{-9}** for #336974, an approximately **0.573% reduction** at the displayed precision. Both completed all 50 public networks without failures. Final-layer MSE was essentially unchanged: 1.852×10^{-8} versus 1.853×10^{-8}.

The archived local comparison is more revealing than the headline alone. Across four additional public networks, the ungated score proxy improved by 0.299% while raw MSE increased by 0.271%, analytical FLOPs decreased by 0.568%, and mean wall time increased by 11.606%. Four exploratory alternatives failed their development-screen criterion. We report these trade-offs, the exact source delta, a reproducible numerical audit, and the limits of inference from a small, adaptively selected public experiment.

Contribution in one sentence

An auditable example of a small compute-efficiency gain on top of a strong inherited estimator, together with evidence that representation, arithmetic order, allocation plans, and the evaluator must be assessed jointly.

Prepared: 30 October 2026. **Participant:** omer_kiraz. **Local build:** V129_MM6_WARM. This is a retrospective technical report; no private re-evaluation result or prize ranking is asserted.

1. What this report establishes

The strongest result is deliberately specific. A small, source-audited implementation change improved the reported public adjusted score while preserving successful evaluation on all 50 public MLPs. The final patch is understandable without reproducing the entire history of the underlying cumulant method. Its value is a clear account of what changed, how much it bought, and what it cost elsewhere.

Claim	Evidence	Scope
Public adjusted score fell 0.573%	3.142e-9 → 3.124e-9; both 50/50 valid	Participant-supplied official summaries
Additional-panel proxy improved 0.299%	4/4 paired public networks	Small local native panel; no official gates
Statistical engine was held fixed	17 top-level definitions unchanged; exact AST delta checked	Source audit, not numerical equivalence
Improvement has costs	MSE +0.271%; wall time +11.606%; higher peak RSS	Additional four networks
Alternative pilots failed the screen	Four increased the development proxy	One development network per pilot

Three kinds of evidence

Official-result evidence consists of the submission listing and detailed AIcrowd dashboard values supplied by the participant in the development conversation. These are transcribed with their original displayed precision in the supplement. They are not a newly downloaded server-side result export. The organiser can verify the submissions directly by ID.

Archived experimental evidence consists of the V129 source package, reference source, comparison JSON, individual native-run receipts, the selection protocol, environment record, and release checks. The report recomputes the score ratios from those receipts. The local native harness does not enforce every official qualification gate.

Fresh static verification performed for this report compares the archived sources, checks the estimator bytes inside both upload archives, rebuilds V129 from its reference and static plan, and recomputes the derived tables. This is a numerical and provenance audit, not a fresh performance benchmark.

How to read the evidence

Sections 2-5 explain the task, inherited estimator, ownership, and final patch. Sections 6-11 present the experimental protocol, official trajectory, local comparison, and failed pilots. Sections 12-15 discuss lessons, limitations, reproduction, and LLM assistance. The appendices provide exact hashes, raw per-network measurements, and a claim-to-file map. The accompanying evidence archive makes those checks possible without relying on prose.

Boundary of the claim

This report does not introduce a new cumulant closure, claim ownership of the upstream corrector, establish a private-set gain, or demonstrate statistical significance for the public-score difference. Its contribution is the measured implementation change and the reproducible analysis around it.

2. The objective and its consequences

Phase 2 supplies the weights of a randomly initialized, bias-free ReLU MLP. The target is the expected post-activation value of every neuron under standard Gaussian input. The challenge shape is width n = 1024 and depth L = 16; the estimator returns a 16 × 1024 array. The relevant reference means are produced by high-budget Monte Carlo. In column-vector notation, the estimation problem is [R1, R2]:

$$x_0 \sim N(0, I), \quad z_\ell = W_\ell x_{\ell-1}, \quad x_\ell = \max(z_\ell, 0)$$
$$\mu_\ell = E[x_\ell], \quad \hat{\mu}_\ell = \text{estimator}(W_1, \ldots, W_\ell)$$
$$E_m = (1/n) \Sigma_i (\hat{\mu}_{m,L,i} - \mu_{m,L,i})^2$$

For a valid run, the per-network score combines final-layer MSE E_m and analytical FLOPs F_m. The budget is $B = 2^{41} = 2{,}199{,}023{,}255{,}552$. The public score averages the per-network products [R1]:

$$s_m = E_m \cdot \max(0.1, F_m/B)$$
$$S = (1/M) \Sigma_m s_m$$

This ordering matters. In general, the mean of the products is not the product of mean MSE and mean utilization. All local score comparisons here calculate each network's product first and then average. Consequently, the official average score and average MSE do not identify the exact average FLOP count of #337188; that quantity is left unreported.

The local trade-off rule

When both candidates are above the 10% multiplier floor on the same network, the score ratio is exactly the MSE ratio times the FLOP ratio. If a change reduces cost by a fraction δ, its MSE may rise by less than δ/(1-δ) and still improve the score. This is a direct consequence of the scoring equation, not a new statistical estimator. It explains why an MSE-only selection rule would reject our final local candidate.

Below the multiplier floor, further compute savings no longer improve the score. The networks in this local experiment remain above the floor. The useful target was therefore a lower product of error and compute, with validity preserved, rather than minimum MSE in isolation.

Validity is a separate requirement

Phase 2 uses analytical FLOPs for the multiplier and separate timing and memory constraints. The starter kit documents a 0.4-second residual cap, a 120-second predict cap, a 5-second setup cap, and an 8 GB solution-process memory limit [R1, R2]. Our native receipts describe an ungated diagnostic. They must not be interpreted as proof of satisfying those official limits; the reported public grading supplies the separate 50/50 validity observation.

3. The inherited estimator

The reference in our archive is attributed to Yan Su / logxio's *fourth-cumulant* implementation and preserves its MIT notices and third-party acknowledgements. Its submitted path is a factored K=3 cumulant estimator with fourth-cumulant regeneration and a learned correction. The repository also discusses separate full K=4 research; that research is not the method we implemented or validated here. We describe the archived estimator rather than substituting the repository's later claims for our own evidence [R3, R4].

Stage	Inherited role	Status in V129
Mean and covariance	Propagate first/second moments and nonlinear updates	Unchanged
Factored third cumulant	Transport source factors and evaluate needed cumulant slices	Unchanged
Fourth-order regeneration	Approximate fourth-cumulant information without a full dense K=4 state	Unchanged
Compression and pruning	Shared/nested bases, age gates, selected active rows/columns	Unchanged
Learned correction	Use local and transported features with fixed learned weights	Unchanged
Arithmetic and storage	Strassen products, pooled buffers, cached views	Depth caps and static plan changed

Why a Gaussian baseline is not the whole estimator

If a scalar pre-activation Z were exactly Gaussian with mean μ and standard deviation σ, then $E[\max(Z,0)] = \sigma\varphi(\mu/\sigma) + \mu\Phi(\mu/\sigma)$, where φ and Φ are the standard normal density and distribution functions. This is an exact Gaussian identity, but the distribution after repeated nonlinear layers is not generally Gaussian. The inherited cumulant machinery approximates corrections to this baseline rather than asserting Gaussianity at every layer.

For an affine step without bias, means and covariances transform as $\mu' = W\mu$ and $C' = WCW^{T}$; the third cumulant transforms with one W factor on each index. Explicit high-order tensors would be expensive at width 1024. The inherited source-factor representation, shared bases, slice contractions, and final-layer-only readout avoid materializing much of that state. These mechanisms explain both the accuracy of the baseline and the concentration of work in matrix products.

The retained configuration includes an old-source shared-basis rank of 352, a nested older-source rank of 224, feedback rank 4, and residual rank 8. These values are read from the archived source, not selected in the final patch. The learned correction carries information across layers through the network weights and uses stored learned parameters. We neither trained that corrector nor altered its parameters in V129.

4. Attribution and the exact scope of our work

An improvement to an open estimator is only interpretable if its ownership boundary is explicit. The large accuracy advantage of the submitted engine comes from existing work. The evidence supports a small additional gain from our configuration and preparation changes; it does not support assigning the entire final score or the historical submission trajectory to a new method of ours.

Component	Credit / origin	Our treatment
Factored K=3 chain and shared bases	504aldo, V29; inherited through logxio	Reused; not rederived as our contribution
Dead-ReLU row pruning	koushik_rudra, V34	Preserved
Selected arithmetic/view cost blocks	acco-cyber V37; upstream notices retained	Preserved underlying implementation
Regeneration and fitted correction in the reference	Yan Su / logxio, with credited upstream methods including Puffi SSC	Reused; no corrector retraining
V129 execution configuration	This project, with Codex assistance	Five globals plus regenerated static plan
Selection and evidence package	This project	Paired receipts, rejected pilots, exact rebuild, report audit

What changed in V129

The final candidate retains the reference estimator's function and class definitions. A structural Python AST comparison finds all 17 top-level function/class definitions unchanged. After removing the static _WPLAN assignment and the five appended global overrides, the complete reference and candidate ASTs are identical. The experiment therefore holds the statistical engine fixed while changing which eligible arithmetic kernels it executes and which scratch objects it prepares.

Our work comprises candidate design and selection, the depth configuration, regeneration of the static preparation plan, paired evaluation, rejection of unfavorable alternatives, packaging, and the present audit. Strassen multiplication itself, pooled scratch management, warm replay, cumulant propagation, and learned correction all predate this intervention. The novelty claim is limited to the measured combination in this submitted configuration and the lessons documented by the experiments.

Data and licensing

The local bundle contains 16 cached public network seed/target records from the Phase 2 public dataset. Five were used in the reported comparison. The inherited corrector's data provenance is recorded in the preserved upstream THIRD_PARTY.md, including public challenge data and a public Keenan Pepper corpus. That record is attribution, not an independently repeated training audit. No private evaluation data was used by the local harness. Original software notices remain in both upload archives; challenge data retain their original terms.

5. The final change: arithmetic and preparation

The final configuration raises the maximum recursion depth for eligible general and covariance products from five to six. It also permits that depth on the first prediction call. Shared-basis operations retain their prior cap, and the designated slice-hub path retains its prior maximum. The complete appended configuration is:

```
STRASSEN_LEVELS = 6
STRASSEN_FIRST  = 6
CPRE_LEV        = 6
STRASSEN_HUB    = 5
STRASSEN_SB     = 5
```

These values are caps, not a requirement to recurse six times on every product. The unchanged level-selection code checks divisibility of all three matrix dimensions and the minimum leaf size, falling back to a shallower level if necessary. The designated _hub2 path uses min(STRASSEN_HUB, self._s_hub) + 1, which remains capped at six. It does not become a seven-level path.

Why one more level can help

A Strassen decomposition replaces eight half-sized block multiplications with seven and adds linear-combination work. At the leading multiplication term, one additional level has a factor 7/8. The total estimator gain is much smaller: only eligible products change, additions and data movement remain, and many products already have restricted dimensions. A 12.5% reduction in a selected leading term is therefore not a prediction of a 12.5% end-to-end saving.

Why the preparation plan changes with it

The existing warm-replay mechanism allocates flat scratch buffers and constructs cached views. Deeper recursion changes the required buffer capacities and view paths. We replaced the static _WPLAN literal to match that operation structure. The plan stores shape and access metadata, not network weights, reference targets, or predictions. Replay is shape-gated and retains the inherited time-bounded behavior; any unprepared views are created lazily.

Static-plan property	Reference	V129
View/preparation events	7,207	7,390
Flat-buffer specifications	30	32
Nominal float32 flat capacity	5,861,575,664 bytes	6,291,834,864 bytes
Shape for replay	1024 × 16	1024 × 16

The plan's nominal flat capacity is not a measurement of resident memory, and views are not independent dense arrays. The recorded development check found predictions bit-identical between MM6 with and without the updated plan on index 4. This supports separating preparation from numerical arithmetic on that case; it is not an all-network bitwise-equivalence proof. The final source still changes floating-point operation order relative to the original reference.

6. Experimental design and selection

A small paired experiment with a declared development screen

The saved protocol selects local public index 4 for development. A candidate advances only if its per-network MSE × max(0.1,F/B) decreases on that network. The four additional indices are 5, 6, 8, and 12. The protocol records that this list was fixed before their results; no independent public timestamp or external preregistration is supplied. We therefore describe a recorded development/validation split, not an externally certified preregistered study.

Item	Recorded choice
Development case	Local public index 4
Additional panel	Local public indices 5, 6, 8, 12
Selection objective	Per-network MSE × max(0.1, FLOPs / 2^{41})
Reference	Archived source associated with #336974
Candidate	V129_MM6_WARM
Isolation	Fresh process per network and method; one numerical thread
Data	Same public weights and cached target means for each pair
Qualification scope	Native diagnostic; official residual/memory gates not enforced

The reference and candidate run on identical reconstructed weights and identical cached targets. Each network/method pair is evaluated in a fresh process with numerical thread counts restricted to one. This controls process history and library thread settings for the paired arithmetic comparison. It does not reproduce the official runner's full lifecycle, where state may persist across multiple networks.

Why the two samples remain separate

Index 4 was used to inspect alternatives and select the final configuration. Its result is consequently optimistic as an estimate of future performance. The four additional public networks are summarized separately. Including development gives a larger apparent improvement, 0.423%, than the additional-network mean, 0.299%. Reporting only the five-network average would obscure that selection effect.

The public indices identify records in public_rows.json, not positions or aliases in the 50-network Alcrowd dashboard. The local network names are timothy-wagner, sara-gill, james-pearson, timothy-davis, and james-reese. These data are publicly available, and independence from every upstream fitting choice or corrector training set has not been established.

No performance rerun disguised as an audit

For this write-up, the preserved logs were recalculated and the source was audited. We did not regenerate Monte Carlo truth, retrain the inherited corrector, or run a new 50-network benchmark. Package validation and the small generated-network smoke result are reported as archived checks. This distinction keeps the report's preparation date separate from the experiment and submission dates.

7. Official public result

The participant supplied a detailed dashboard for the best submission and a submission list tying the new result to #337188. The immediately preceding result is #336974. Both are reported as graded successfully. The detailed values below have more precision than the rounded listing, which displays 3.12e-9 and 3.14e-9 respectively [E1].

Reported metric	#336974	#337188
Adjusted score	3.142e-9	3.124e-9
Final-layer mean MSE	1.853e-8	1.852e-8
All-layer mean MSE	8.54e-9	8.54e-9
Public networks graded	50 / 50	50 / 50
Failures	0	0
Submission date in listing	18 October 2026	19 October 2026

The displayed adjusted-score improvement is (3.142 - 3.124)/3.142 = 0.5729%. The final-layer MSE values differ by approximately 0.054% at their displayed precision, and the all-layer mean is unchanged at the displayed precision. These summaries support a small gain in the budget-adjusted objective with essentially stable raw error. They do not establish that every network improved.

Distribution reported for #337188

Public diagnostic	Reported value
Best / worst adjusted network score	2.50e-9 / 3.96e-9
25th / 75th percentile	2.88e-9 / 3.32e-9
Layer-error growth L0 → L15	26.94×
Steepest reported layer	L3, 1.65×
Dashboard comparison to Monte Carlo	362×

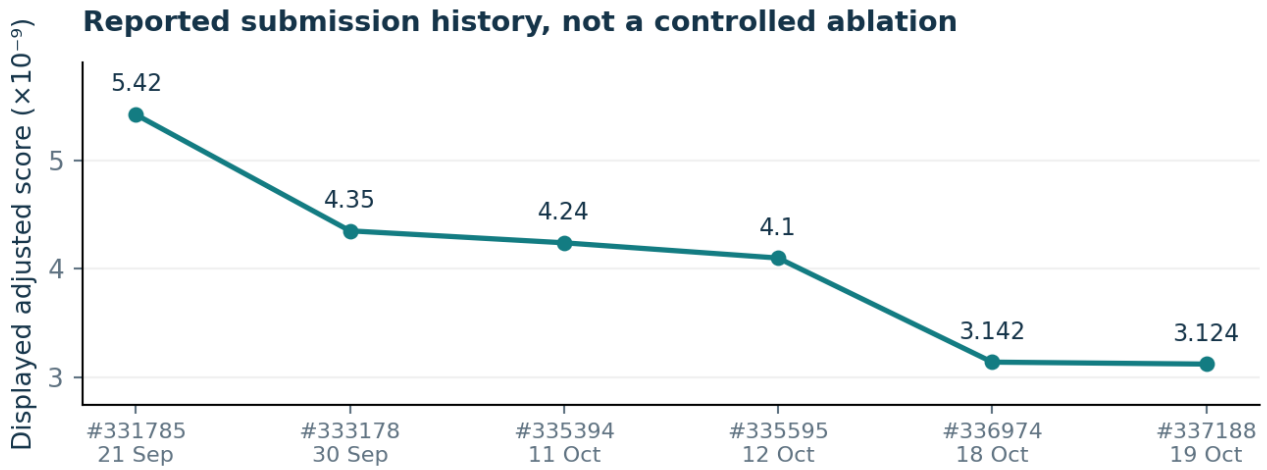
The dashboard identifies the best and worst public networks as jason-davies and jerry-kerr. These are within-submission network diagnostics, not leaderboard positions. The reported 362× comparison against Monte Carlo is a dashboard score comparison; it is not a measured wall-clock acceleration and is not attributable to our final patch alone.

What cannot be recovered from the summary

A full paired per-network official export is not available in this evidence package. We therefore do not calculate a public-network confidence interval, a paired win count, or a significance test. The new candidate's exact mean compute and runtime are also not supplied in the pasted summary. We do not infer either from ratios of aggregate metrics.

8. How the best score was reached

The submission history gives useful context for the final engineering decision. The table and chart reproduce selected participant-reported milestones. They show a sustained reduction in the displayed adjusted score, with a larger transition into the strong inherited baseline and a smaller final improvement. This is a historical trajectory, not a controlled ablation of each earlier version.



ID	Submitted	Adjusted score	Final MSE
#331785	21 Sep 2026	5.42e-9	2.13e-8
#333178	30 Sep 2026	4.35e-9	2.19e-8
#335394	11 Oct 2026	4.24e-9	2.19e-8
#335595	12 Oct 2026	4.1e-9	2.14e-8
#336974	18 Oct 2026	3.142e-9	1.853e-8
#337188	19 Oct 2026	3.124e-9	1.852e-8

Between #331785 and #337188 the displayed adjusted score falls by approximately 42.36%. That number describes the submitted solutions as a whole. It combines method changes, inherited components, implementation choices, and potentially evaluator-version effects; it is not an estimate of the contribution of the V129 patch. Source-and-log evidence sufficient for controlled attribution is retained for the final reference/candidate comparison, not for every milestone.

The decisive final choice

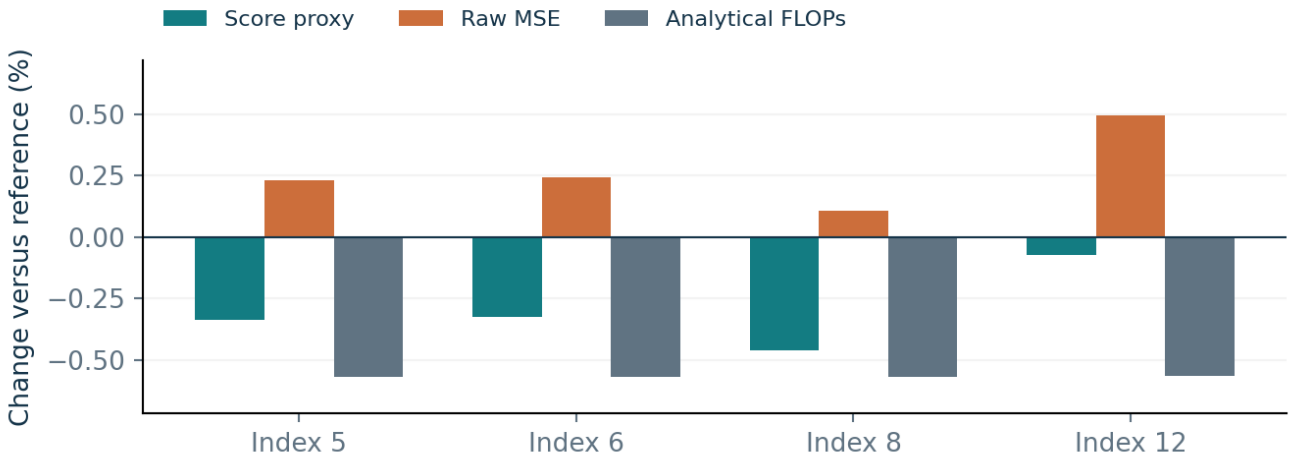
Once #336974 supplied a successful, competitive reference, the most defensible next step was to preserve its statistical machinery and search for a small cost reduction. The local alternatives that changed compression or correction behavior did not pass the development screen. The selected configuration changed the matrix-product execution plan instead and then survived the four additional public checks. The later participant-reported official score improved in the same direction.

The resulting story is one of narrowing scope as the baseline improves. Large historical progress established a useful engine; the final contribution extracted a small remaining compute benefit while making the accompanying accuracy, memory, and runtime costs explicit.

9. Paired local accuracy and compute

All values in this section are native, ungated diagnostics with flopscope 0.12.3 and whestbench 0.16.2. Lower score-proxy change is better. The development row is shown separately from the four additional public-network rows [E2].

Index / role	MSE change	FLOP change	Proxy change
4 / dev	-0.273%	-0.760%	-1.030%
5 / additional	+0.233%	-0.569%	-0.338%
6 / additional	+0.244%	-0.569%	-0.326%
8 / additional	+0.107%	-0.569%	-0.463%
12 / additional	+0.495%	-0.567%	-0.075%



Every additional-network proxy improves, but every additional-network raw MSE rises slightly. This is not contradictory: the FLOP reduction is larger than the accuracy loss in the scoring product. Index 12 is the weakest local case, with only a 0.075% proxy improvement; it illustrates how little margin remains when a small rounding-sensitive error increase nearly consumes the compute saving.

Averaging the per-network products yields 3.206154512e-9 for the reference and 3.196569137e-9 for V129 on the additional four networks. The ratio of means is 0.9970103, or a 0.2990% improvement. The mean-MSE ratio is 1.0027073 and the mean-FLOP ratio is 0.9943165. Multiplying the latter two rounded aggregate ratios is not the method used to compute the score.

Across all five networks, including the development case, the mean proxy falls from 3.090731145e-9 to 3.077645425e-9, a 0.4234% improvement. This is reported for completeness, not as the preferred validation estimate. With four additional public networks, no strong population-level or private-generalization claim is justified.

10. Runtime, memory, and numerical behavior

A lower analytical operation count does not necessarily mean faster execution. An extra recursion level creates more small products, combinations, intermediate buffers, and control-flow work. This is especially relevant when large BLAS operations are replaced by a larger number of small operations. The local measurements show the trade-off directly.

Additional-panel metric	Reference	V129	Change
Mean raw MSE	1.91637e-08	1.92155e-08	+0.271%
Mean analytical FLOPs	3.67915e+11	3.65824e+11	-0.568%
Mean wall time	52.80975 s	58.93895 s	+11.606%
Max process RSS	5,177,828 KiB	5,604,168 KiB	+8.234%
Max native residual	0.62399 s	0.71341 s	+14.330%

Interpretation of the time increase

The mean wall time on the four additional networks increases from 52.81 to 58.94 seconds. Because Phase 2 does not price residual time into the adjusted score, this increase does not directly cancel the analytical FLOP saving. It does reduce operational margin. A hypothetical deployment concerned with latency could prefer the reference even when the contest objective prefers V129.

The native residual measurements exceed the documented official 0.4-second cap for both reference and candidate. This harness did not enforce that cap, so its results are not local SDK qualification results. The reported official public success of both submissions is separate evidence. Differences in execution environment, process lifecycle, and instrumentation prevent treating a local timing number as a universal validity prediction; their individual causal contributions were not isolated.

Floating-point order is part of the experiment

The unchanged source-level mathematical rules do not imply identical floating-point outputs. Strassen reorganizes additions, subtractions, and multiplications. In float32, cancellation and rounding can change propagated moments and the features consumed by the learned corrector. The observed MSE changes are compatible with this mechanism, but we did not measure a layer-by-layer numerical sensitivity decomposition and do not claim a unique causal explanation for each neuron's change.

The separate recorded plan-only check is narrower: on the development network, updating the scratch/view plan without another arithmetic change left MM6 predictions bit-identical. That suggests the plan update itself was numerically neutral on that case. It does not make the complete reference-to-V129 transition numerically exact.

Engineering consequence

Optimize analytical cost, numerical behavior, latency, and peak memory together. A single scalar score is sufficient for ranking valid submissions, but insufficient for describing the implementation.

11. Four experiments that did not earn their cost

Four archived exploratory pilots were screened on development index 4 and rejected because their score proxy increased. They were not advanced to the four-network additional panel. These are local negative results, not universal impossibility claims; the evidence archive retains their receipts, while complete standalone source snapshots for every rejected pilot are not retained [E2, E3].

Development pilot	Raw MSE	FLOPs	Proxy change
Old-source rank 288	1.7016e-08	3.6499e+11	+7.427%
Columns minus 64	1.7860e-08	3.7723e+11	+16.537%
Two-corrector ensemble	1.5721e-08	3.7378e+11	+1.641%
k31 residual term	1.5839e-08	3.6896e+11	+1.082%

11.1 A narrower old-source basis

Reducing the old-source rank from 352 to 288 lowered cost but increased error enough to worsen the proxy by 7.43%. The result argues against the particular compression setting on this baseline: a relatively small remaining computational saving can discard information that the downstream mean estimate still needs. It does not prove that all alternative basis constructions or rank schedules fail.

11.2 More aggressive column trimming

The columns-minus-64 pilot made both raw error and billed compute worse, increasing the proxy by 16.54%. Fewer nominal columns did not imply a cheaper measured execution. A plausible structural explanation is interaction with divisibility and recursion depth in the unchanged Strassen selector. The receipt establishes the unfavorable outcome; a full per-kernel counterfactual cost decomposition was not preserved, so the mechanism remains an interpretation.

11.3 An ensemble of two correctors

The two-corrector ensemble increased compute and slightly worsened MSE, producing a 1.64% proxy regression. Averaging outputs is useful only if the errors, state evolution, and added cost cooperate. This single pilot did not establish enough complementary error to pay for a second corrector. It gives no evidence that every ensemble weighting or independently trained model would behave the same way.

11.4 A residual fourth-cumulant slice term

The k31 residual pilot added a correction associated with the (3,1) fourth-cumulant slice. Its MSE increase dominated a very small compute increase, worsening the proxy by 1.08%. A theoretically motivated missing term can interact unfavorably with an already fitted closure and learned corrector. The experiment therefore supports testing a correction end to end, including existing compensations, before judging it from a local formula alone.

12. Lessons from the complete decision

Lesson 1: Preserve the objective when optimizing the implementation

The final local candidate improved the scored objective despite a slight MSE regression. This is a concrete example of why accuracy-only and cost-only screens are both incomplete. The per-network product should be calculated from the actual meter receipt, followed by the suite average. A theoretically cheaper construction or a lower raw error does not establish a better submission.

Lesson 2: Representation and arithmetic choices interact

Compression changes dimensions; dimensions determine eligible recursion levels; recursion changes arithmetic and scratch requirements. The failed trimming pilot and the successful depth/plan configuration make it unsafe to optimize these pieces as independent knobs. The appropriate unit of evaluation is the complete estimator on a fixed network, including representation construction and all downstream work.

Lesson 3: Prepare storage for the operation structure actually used

A deeper arithmetic schedule requires a matching scratch and view plan. Reusing an older preparation plan can force lazy reconstruction or buffer growth during prediction. In this project, the selected candidate includes the regenerated plan rather than just the depth overrides. The saved bitwise development check separates the plan's role from the numerical effect of changing arithmetic order, within the scope of that one check.

Lesson 4: A strong learned baseline changes the value of added theory

A closure correction may approximate a missing physical or statistical effect while also disturbing errors that another component has learned to compensate. Our residual-slice and ensemble pilots did not help this inherited engine. The practical lesson is to evaluate changes with the intended corrector and state transport, and to avoid presenting a failed local augmentation as evidence against an entire mathematical idea.

Lesson 5: Validity deserves its own evidence trail

Shape checks, finite outputs, package checks, runtime gates, and accuracy are different tests. A generated small-network smoke check catches interface failures but says little about 1024-wide resource use. A native full-network run can diagnose score trade-offs but not enforce every official cap. The official public result is therefore recorded separately rather than treated as interchangeable with local validation.

Lesson 6: A small, traceable gain can teach more than an inflated claim

The final change is modest. Its useful content is that the source delta is exact, the unfavorable metrics are included, the rejected directions are documented, and the reported public result is consistent with the local selection direction. This provides a reusable evaluation pattern for future cumulant, compressed-state, or learned-correction estimators without claiming that the same settings will transfer unchanged.

13. Limitations and threats to validity

Small, adaptively selected public sample

The final local comparison includes one development network and four additional public networks. Several candidates were examined on the development case, so its improvement is subject to selection bias. The four additional successes are encouraging but too few for a strong population claim. We do not report p-values, bootstrap intervals, or a private-set win probability from this sample.

Upstream fitting and data overlap

The inherited chain contains fitted tables and a learned corrector. This project did not retrain them and did not independently audit their entire training history. The public-network panel must not be described as wholly unseen by every upstream component. Fresh private networks would be the appropriate test of generalization; their results are not part of the evidence available here.

Rounded official summaries and incomplete artifact binding

The official metrics were supplied by the participant and are rounded dashboard values. A server-side artifact digest and full per-network export were not independently retrieved for this report. The local V129 package is associated with #337188 in the development record, and its bytes are verified internally. The organiser's stored submission is the authority for independently confirming that association and the public metrics.

Hardware and lifecycle scope

The local runner starts a new process for every network/method pair. Official suite execution can reuse process state. Timing, memory residency, and warm-plan completion may consequently differ. The preserved environment records package versions and thread settings but is not a complete hardware image. Reproducing analytical FLOPs and source identity is stronger than reproducing identical wall time on another machine.

Limited ablations and numerical attribution

There is no complete factorial study separating every depth cap, buffer size, preparation event, and learned feature. The final chosen combination is measured; the isolated contribution of each setting is not. A recorded plan-only bitwise check exists on one development network, but its full prediction arrays are not in the current bundle. No float64 reference study quantifies rounding amplification through the full chain.

Distribution and task scope

The calibrated path targets the challenge's 1024 × 16, bias-free random ReLU networks. The source includes a generic off-shape path and an archived 128 × 16 finite-output smoke check, but neither establishes accuracy on other widths, depths, activations, biases, or trained networks. No result here supports a general neural-network verification or rare-event-estimation claim.

Originality and award scope

This is an implementation and evaluation contribution built on substantial prior work. It is not a new full K=4 algorithm or a newly trained correction model. Whether that contribution and a late write-up satisfy prize criteria is for the organisers to decide. The report does not assert eligibility, selection, acceptance, or a final ranking.

14. Reproduction and audit procedure

The supplement preserves the original WHEST_V129_CODE_AND_RESULTS.zip as well as the two upload archives. Extract the original code/results ZIP to obtain the WHEST_V129 directory. The archived environment is Python 3.13.16, NumPy 2.3.5, SciPy 1.17.0, flopscope 0.12.3, whestbench 0.16.2, and threadpoolctl 3.6.0. Numerical library versions should be pinned before comparing receipts.

A. Audit without a full benchmark

```
python audit_evidence.py
# checks source hashes, package contents, AST delta,
# raw receipts, derived ratios, and manuscript tables
```

This report's audit verifies that the packaged source matches the archived candidate, that the reference package matches the archived reference, and that the candidate changes only the static plan and five globals. It also verifies that the saved comparison ratios equal recomputation from the individual measurements. These checks require only ordinary Python and do not spend the challenge compute budget.

B. Rebuild the exact candidate

```
cd WHEST_V129
python build_candidate.py
# expected: Exact candidate rebuilt; SHA256 verified.
```

The builder checks the reference hash, appends the settings, replaces the _WPLAN literal, and checks the candidate hash. The reconstruction was executed successfully during report preparation. This verifies exact local bytes, rather than merely a semantically similar reimplementation.

C. Repeat the paired native diagnostic

```
python -m pip install -r requirements.txt
python benchmark.py --mode native --indices 4 5 6 8 12 \
  --methods REFERENCE V129
```

The harness writes per-run JSON and prediction arrays to a timestamped my_results directory. Do not pool the development row into the additional-network headline. Compare per-network products, retain wall and residual times, and inspect failures. This command intentionally remains a native diagnostic; use the official starter-kit subprocess workflow for a separate qualification check [R1, R2].

D. Check the upload artifact

```
whest validate-package WHEST_V129_MM6_WARM.tar.gz
```

The archived release check reports successful package validation and a finite 128 × 16 smoke output costing 2,269,423,640 FLOPs. Both archives retain LICENSE, README.md, THIRD_PARTY.md, estimator.py, and manifest.json. The older package documents were written before the new official result and correctly contain an unknown candidate score; this report adds the later participant-supplied outcome without rewriting those historical records.

15. LLM assistance and author responsibility

Human and model roles

Ömer Kiraz directed the project goal, requested improvements, supplied the AIcrowd results, and made the participant-side submission decisions. OpenAI's ChatGPT/Codex assisted with inspecting public material, implementing the final configuration and preparation changes, writing the local evaluation harness, running the recorded experiments, checking source identity, packaging the candidate, and drafting this retrospective report.

The arithmetic, source comparisons, and tables in this write-up are grounded in preserved files or explicitly identified participant-provided dashboard values. Text generation is not treated as an experimental measurement. The present audit recomputes the derived numbers from the archived receipts. It does not convert a model-generated interpretation into an independently established mechanism.

No use of Anthropic's Claude is asserted for this local contribution because such use is not established by the retained record. Upstream authors may disclose their own tools in their respective reports; those disclosures are not adopted as claims about this participant's work. Exact model versions and a complete prompt-by-prompt transcript are not part of the released evidence package.

Attribution in the submitted implementation

The underlying estimator, corrector parameters, and substantial numerical machinery remain the credited upstream authors' work. Their notices are preserved. The source delta attributed to this project is the V129 configuration and static plan replacement, supported by the AST comparison. The experimental selection and analysis reported here concern that delta and the archived rejected pilots.

Research value and the next decisive experiment

A useful follow-up would evaluate both frozen archives on the same fresh network suite, export the complete per-network meter and error records, and report uncertainty with the network as the unit of resampling. It should separately measure first-call and steady-state behavior, preserve prediction arrays for a numerical-order study, and record the exact hardware and runner settings. That would test the generality of the small gain without retuning on the evaluation suite.

A second follow-up could explore shape-aware recursion policies using an explicit development set and frozen additional tests, jointly constraining MSE, FLOPs, latency, and memory. These are proposed experiments, not completed results. The available evidence supports the present configuration on its measured local panel and the participant-reported official public outcome.

Concluding contribution

The best documented result is a 3.124e-9 public adjusted score with 50/50 successful networks. The additional contribution is a small compute reduction with visible costs, an exact source-level boundary, four documented negative experiments, and a reproducible account of how the candidate was selected.

Appendix A. Source identity and artifact contents

Source hashes are the most stable identifiers of the locally audited estimators. The archived provenance file records the upstream repository and commit below. During preparation of this report, the repository was accessible but its recorded commit URL did not resolve through the GitHub API. The bundled source bytes and hashes therefore provide the reproducibility anchor; no assumption is made that the repository's current main branch is identical.

Reference estimator SHA-256

```
253796045ba6bdf2bdb0e33bc4b67721
f8afb081d75e106b439bd89aebcea669
```

V129 estimator SHA-256

```
2dc355f80f31f7bf9496ad49ccab4220
5b36184ac21878ccef6058610ed52104
```

Archived upstream provenance

```
Repository: https://github.com/logxio/fourth-cumulant
Recorded commit:
7c545065851f5ce54227a0ecb0a059ea33ca58d1
```

Check	Outcome
Reference upload source hash	Matches reference/estimator.py
Candidate upload source hash	Matches candidate/estimator.py
All 17 function/class definitions	Unchanged
Whole-module AST after removing documented delta	Identical
Exact builder replay	Candidate SHA-256 reproduced
Paired score-ratio recalculation	Matches comparison.json
New full-network performance run for this report	Not performed

The static-plan audit counts 7,207 reference events and 7,390 candidate events, with 30 and 32 flat-buffer specifications respectively. The nominal float32 capacities are calculated from integer element counts, not inferred from measured RSS. The plan fields are n, L, flat, and ev; they describe shape, capacities, and view-construction paths.

The original evidence ZIP includes reference and candidate estimator.py files, licenses, benchmark.py, build_candidate.py, requirements.txt, public_rows.json, comparison.json, experiment_protocol.json, environment.json, provenance.json, release_checks.json, settings.txt, warm_plan.literal, and individual logs. The new supplement adds this manuscript, report-generation code, audit code, transcribed official metrics, derived tables, and figures.

Appendix B. Raw paired measurements

Values below are copied from the archived run receipts; score proxies and percentage changes are independently recomputed. The index refers to the local cached public-record order. FLOPs are analytical counts. MSE is final-layer raw error. Memory is process peak RSS in KiB, which is distinct from the plan's nominal capacity.

Index	Reference MSE	V129 MSE	Reference FLOPs	V129 FLOPs
4	1.567389e-08	1.563116e-08	368,849,981,807	366,047,786,483
5	2.301364e-08	2.306717e-08	367,530,971,525	365,439,911,813
6	1.715752e-08	1.719945e-08	367,667,988,599	365,576,928,887
8	1.764442e-08	1.766325e-08	367,530,971,525	365,439,911,813
12	1.883904e-08	1.893228e-08	368,928,951,021	366,837,891,309

Index	Wall ref / new (s)	Residual ref / new (s)	RSS ref / new (KiB)
4	54.09826 / 57.62909	0.64641 / 0.64494	5,362,428 / 5,788,940
5	52.87191 / 57.88288	0.61737 / 0.63593	5,102,900 / 5,526,116
6	52.81559 / 58.15386	0.61694 / 0.62815	5,027,180 / 5,436,592
8	52.68970 / 58.07471	0.62331 / 0.63561	5,100,640 / 5,527,872
12	52.86181 / 61.64434	0.62399 / 0.71341	5,177,828 / 5,604,168

Index 4 is the development network. Its reference receipt contains an additional note about arithmetic-ledger instrumentation, so the main timing comparison uses only indices 5, 6, 8, and 12. The maximum candidate RSS across all five runs is 5,788,940 KiB, approximately 5.52 GiB. Across the additional four, the maximum is 5,604,168 KiB, approximately 5.34 GiB.

Recorded raw pilot receipts are also preserved. The rejection screen compares each pilot against the same development-network reference. Only the final candidate has the additional four-network paired validation in this archive; the pilot table must not be interpreted as a five-method validation tournament.

Appendix C. Claim-to-evidence map and references

Claim / check	Primary evidence in supplement
Official scores and 50/50 validity	official_results_transcribed.json [E1]
Paired local gains and costs	comparison.json; logs/validation_{reference,candidate}_*.json [E2]
Rejected pilots	logs/r288_4.json, c64_4.json, ensemble_4.json, k31_4.json [E3]
Recorded selection protocol	experiment_protocol.json [E4]
Exact patch and rebuilding	reference/estimator.py; candidate/estimator.py; settings.txt; warm_plan.literal; build_candidate.py [E5]
Versions and archived release checks	environment.json; requirements.txt; release_checks.json [E6]
Fresh static/numeric audit	audit_evidence.py; derived_audit.json; audit_result.json [E7]

References and evidence identifiers

[R1] AIcrowd. Whest starter kit: README and Scoring Model. Accessed 30 October 2026. Official specification for the objective and validity gates. Source link.

[R2] AIcrowd. Whest starter kit: Competition Rounds. Accessed 30 October 2026. Phase 2 shape, budget, limits, and differences from Phase 1. Source link.

[R3] Yan Su / logxio. fourth-cumulant. Archived estimator, LICENSE, and THIRD_PARTY.md retained in E5. Current repository content may differ from the archived source. Source link.

[R4] 504aldo. whest-p2-cumulant-k3, V29; koushik_rudra V34 and acco-cyber V37. Credits and notices reproduced in the archived THIRD_PARTY.md and LICENSE. Source link.

[E1] Participant-supplied AIcrowd submission listing and detailed dashboard excerpts for #336974 and #337188. Transcribed in official_results_transcribed.json; no server-side export.

[E2-E6] WHEST_V129_CODE_AND_RESULTS.zip. Preserved local reference, candidate, comparison, protocol, logs, environment, and release checks. Historical candidate-official-score fields remain null.

[E7] This report's audit and rendering sources, derived measurements, artifact hashes, and source-delta verification. Generated from E1-E6 on 30 October 2026.

Reporting note. Dashboard-derived values retain their displayed precision. Local derived values may show more digits to support exact recalculation. More printed digits do not imply greater population-level certainty. The report and supplement were prepared on 30 October 2026; the experiments and official results are historical records, not a claim that new submissions remain open.