

Kerdock/MUB Spherical Cubature for White-Box Activation Estimation

A static, network-independent 5-design plus tracked Strassen–Winograd propagation, under the Phase 1 effective-compute budget

Alcrowd Phase 1 submission ID: 320802

ARC White-Box Estimation Challenge 2026 | Participant **skye_nygaard** | Graded successfully, 50/50 public MLPs, 0 failures
Submitted 2026-07-29 13:03 | Write-up prepared 2026-08-03 | Submission page:
aicrowd.com/challenges/arc-white-box-estimation-challenge-2026/submissions/320802

Graded metric (public split, 50 MLPs)	Value
Adjusted final-layer score	1.55e-7
Final-layer MSE	2.416e-7
Ratio vs. Monte Carlo reference (6.47e-7)	4.2x
Best / worst MLP	3.62e-8 (dylan-meyer) / 5.11e-7 (angela-walker)
IQR, p25–p75	9.81e-8 – 1.94e-7
All-layers MSE, mean	0.7437 (see §3.7)
Mean effective compute	1.75e11 of a 2.72e11 budget (64.19%)
Failures	0 of 50

Scope of the headline number. These are public-split figures. The 50-MLP private split is sealed until Phase 2 close, so the final rank for this submission is not yet determined. Everything in this document that is not marked as an official grader result is a local measurement on the exposed development cohort, which is a *different set of networks* from the graded cohort. The two are never chained.

1. What was submitted, in one paragraph

The estimator replaces Monte Carlo sampling of the input distribution with a **deterministic algebraic cubature rule**. Because the challenge input distribution is an isotropic Gaussian, the expectation defining each post-ReLU mean is an integral over the sphere in dimension 256. That integral is evaluated on a fixed 66,048-point spherical 5-design built from the Kerdock code and the associated maximal real mutually unbiased bases (MUBs). The design is chosen once, frozen as a shipped asset, and reused unchanged for every network — it never looks at the weights. Two structural properties of the Kerdock construction then make the rule affordable: the design matrix factors through a Walsh–Hadamard transform, so applying it to the first weight matrix costs eight butterfly stages instead of an explicit 66,048 x 256 by 256 x 256 product; and the remaining propagation is charged through a tracked depth-5 Strassen–Winograd kernel that costs $7^5 = 16,807$ products where the conventional recursion charges $32^3 = 32,768$. The accuracy comes from the design; the affordability comes from the structure of the same algebraic object.

2. Why this is the interesting part

The measurable claim of this submission is not that a large deterministic design beats sampling — that is expected. It is that **the cost side moved without the accuracy side moving at all**. On one fixed 100-network development cohort, holding the statistical design, the node set, the radius, and the returned rows exactly constant, and changing only the arithmetic used to propagate the design:

Arm (identical 66,048-point design)	Raw final MSE	Effective compute	Budget	Adjusted score
Dense propagation	2.2826e-7	2.689e11	98.86%	2.2565646e-7
Tracked Winograd propagation	2.2819e-7	1.748e11	64.27%	1.4641716e-7

Raw MSE is unchanged to three significant figures. The adjusted score improves **1.5412x**, and every bit of that came from the charged-cost term. On ten held-out selection networks the two arms agree to a maximum final-mean drift of $7.551\text{e-}6$ (dense raw $1.715489347\text{e-}7$, Winograd raw $1.715362009\text{e-}7$), confirming the change is cost-only rather than a different estimator. This is the mechanism I would want ARC to look at: under an effective-compute score, an exact restructuring of the arithmetic is worth as much as a better statistical rule, and the Kerdock design happens to admit both.

3. Method

3.1 Objective

Given the weights of a width-256, depth-32 ReLU MLP, predict the per-neuron mean post-ReLU activation under the input distribution. The graded quantity is the adjusted final-layer score: final-layer MSE scaled by a multiplier derived from consumed compute, where effective compute is tracked FLOPs + $1\text{e}11 \times \text{residual wall seconds}$ against a $272\text{e}9$ budget. The grader's hard wall-clock cap is 60 s per network; the 30-second per-network figure used below is a self-imposed development limit, chosen because the development machine runs faster than the grader.

3.2 Layer 0 is closed form, not estimated

Under an isotropic Gaussian input the pre-activation of unit j in layer 0 is a centered Gaussian with standard deviation $\|w_j\|$, so its post-ReLU mean is the half-normal mean $\|w_j\| / \sqrt{2\pi}$ exactly. No cubature is spent there.

3.3 The design: a static Kerdock/MUB spherical 5-design

For the remaining layers the estimator propagates a fixed cloud of directions rather than random samples. The cloud is a spherical 5-design in dimension 256 — a node set on which the average of every polynomial of degree at most 5 equals its exact spherical average. It is assembled from 128 Kerdock/maximal-real-MUB bases plus the coordinate basis, each direction carried together with its antipode:

```

128 bases x 256 directions x 2 antipodes = 65,536
1 basis x 256 directions x 2 antipodes =    512
total = 66,048 rows

```

No basis is dropped and no row is subsampled. Antipodal closure kills every odd-degree term for free, so a 5-design costs the same as a 4-design here. The radius is set to the mean Gaussian radius in dimension 256, computed from a log-gamma ratio, which matches the first moment of the Gaussian shell that the design is standing in for. The whole object is frozen in `kerdock_mub5_seed3.npz` at rotation seed 3, selected once on development IDs 0–49 and then held fixed; IDs 50–99 were kept as a frozen holdout during that selection.

The design is **network-independent**. It is not fitted, tuned per network, or adapted during prediction. The estimator is deterministic and does not read the budget argument or any grader seed; repeat predictions are bit-identical.

3.4 Applying the design through a Walsh–Hadamard transform

The Kerdock bases are generated by quadratic-phase (chirp) vectors acted on by a Walsh–Hadamard transform. That structure is the reason the rule is affordable at all: instead of forming the $66,048 \times 256$ design matrix and multiplying it by W_0 , the estimator multiplies the rotated weight matrix pointwise by the chirp table and runs eight butterfly stages over the basis axis. This is the single place where the algebra of the Kerdock code, rather than the geometry of the design, does the work.

3.5 Propagation: tracked depth-5 Strassen–Winograd

Hidden layers 1–30 propagate the full 66,048-row activation through each 256×256 weight matrix, with a ReLU after each. These are the dominant cost. They are evaluated by an exact depth-5 Strassen–Winograd recursion: five levels of the 7-multiplication

scheme give $7^5 = 16,807$ leaf products where the conventional blocking charges $32^3 = 32,768$, a **1.9497x** reduction in charged multiplies. The implementation carries the first three levels as tensor axes and keeps the deepest two levels' sixteen decoded quadrants as a small Python tree that is assembled exactly once, which removes seven large intermediate block copies per hidden layer. The result is the exact product, not an approximation, and every operation goes through the tracked array namespace — there is no untracked arithmetic, no direct NumPy import, no data-dependent indexing.

3.6 Final layer: chunked, immediately reduced

The last propagation is evaluated in 2,048-row chunks, each chunk reduced straight into a float64 accumulator, so the full 66,048 x 256 final activation is never materialized in float64. The propagated row set, the kernel, and every matrix product are identical to the unchunked form; only evaluation order changes.

3.7 What the estimator does not do

Layers 1 through 30 are returned as zeros. Only layer 0 and layer 31 carry estimates. This is a deliberate choice against the scored objective, which is the adjusted *final-layer* score, and it is the entire reason the all-layers MSE sits near 0.74 while the final-layer MSE is 2.4e-7. Read as a full activation-profile estimator this submission is poor; read as a final-layer estimator it is what it claims to be. The cubature machinery would extend to the intermediate layers — the design is propagated through all of them anyway — but writing those rows was never scored and was not spent on.

The estimator also checks that width is 256 and depth is 32 and returns zeros otherwise. It is a specialized entry for this benchmark, not a general tool.

4. How it was developed

The route to this submission went through three families, each abandoned on measurement rather than on taste. Randomized quasi-Monte Carlo blends came first and plateaued near 3.46e-7 adjusted on the development cohort. Replacing the randomized rule with the static Kerdock/MUB 5-design was the first mechanistic step and bought 1.5336x. At that point the estimator was consuming 98.86% of the FLOP budget, so accuracy improvements were nearly worthless: any extra work would be taxed straight back through the compute multiplier. The second mechanistic step was therefore not statistical at all — it was recovering budget headroom through exact restructured arithmetic, which bought a further 1.5412x at unchanged raw error and left roughly a third of the budget unused.

Development trajectory (exposed Mini-100 cohort)	Adjusted score	Gain
Two-nearfull RQMC blend	3.460699e-7	—
Kerdock/MUB 5-design, dense propagation	2.256565e-7	1.5336x
Same design, tracked Winograd propagation	1.464172e-7	1.5412x

The graded submission #320802 is the Winograd arm as it stood on 2026-07-29. The 1.464172e-7 row above is a *later* build measured on the development cohort on 2026-07-31; it differs from the graded archive by 33,816,576 tracked operations in the final reduction and was never submitted. It appears here as design evidence only.

5. Cost accounting, and the mistake it prevented

Quantity	Value
Tracked FLOPs per network	170,906,815,488
FLOP budget	272,000,000,000
Mean effective compute (graded)	~1.745e11 (64.19% utilization)
Official row-0 wall time	24.28 s against a self-imposed 30 s limit (grader cap 60 s)
Official row-0 charged residual	0.0547 s

Quantity	Value
Peak RSS, production arm	2,172 MiB

Because effective compute is tracked FLOPs *plus* a large multiple of residual wall time, tracked FLOPs alone is the wrong selection criterion. This is worth stating plainly because it cost me a promotion decision. A streaming variant of the kernel (internally ‘A43’) saved exactly the 524,123,904 tracked operations it was projected to save and cut peak memory by 82.74%, with output bit-identical to the shipped arm. It was still **rejected**: it paid for those operations with roughly 2.2 s of extra charged residual wall time per network, about +222e9 of effective compute against a 5.241 ms break-even margin, and it pushed wall time past the self-imposed 30-second limit. The tracked-FLOP projection was arithmetically correct and the decision it implied was wrong.

Transferable warning. On this benchmark, any candidate must be priced on complete measured subprocess cost, not on tracked FLOPs. The two criteria disagree, and they disagree by more than the margins competitors are fighting over. Residual wall time is also hardware-dependent: all local timings here come from a single macOS arm64 machine with BLAS pinned, and the one available calibration point suggests the official grader ran about 11% slower than this machine.

6. Ablations

6.1 Basis count: the full design is the operating point

Four frozen packages take a literal prefix of the original basis order, with no statistical correction, to test whether a cheaper partial design wins back more in compute than it loses in accuracy.

Bases	Rows	Raw MSE	Adjusted ratio vs. 129	Local wall	Peak RSS
129	66,048	1.75875e-7	1.0000	24.77 s	407.4 MiB
96	49,152	3.21805e-7	1.3743	18.72 s	335.8 MiB
64	32,768	4.86473e-7	1.4096	11.80 s	319.7 MiB
32	16,384	1.04775e-6	1.5579	6.21 s	303.3 MiB

Every partial arm is worse on adjusted score. MSE scales as roughly $k^{-1.21}$ to $k^{-1.24}$ in the basis count; an exponent above 1 is exactly the condition for the adjusted curve to favor the complete design, since accuracy is being lost faster than compute is being recovered. A partial design would only make sense as a cheap host for some other correction, and at 96 bases the measured hurdle for such a correction was 1.2841x raw gain at +5e9 compute, against a 1.0670x projection from archived data. Nothing available cleared even the optimistic bar, so 129 ships standalone.

Hedge: this curve is an archived exposed-split projection, not an official four-arm measurement. The frozen test weights behind it were stored in float16, which the original report warned could matter near 96-base parity — and the measured 96-base arm did come in worse than the projection predicted.

6.2 Final-layer reduction form

Two reduction forms were measured head to head on the same networks. They produce **bit-identical** output — maximum absolute difference exactly 0.000e+00 — so this is purely a cost choice:

Form	Tracked ops	Residual	Peak RSS
sum(activation.astype(float64), axis=0)	170,908,912,640	54.5 ± 3.3 ms	3.883 GiB
sum(activation, axis=0, dtype=float64)	170,875,096,064	56.0 ± 4.1 ms	3.880 GiB

The fused form was selected on the deterministic term: 33,816,576 fewer tracked operations every run. The 1.5 ms residual difference pointing the other way sits inside the 3–4 ms per-network standard deviation and is not a reproducible signal. This change postdates the graded archive.

6.3 Streaming arithmetic variants (rejected)

Two full-depth streaming rewrites, A42 and A43, were built and measured. Both are locally bit-identical to each other and drift from the production arm by at most $3.418\text{e-}12$ RMS. A43 reduced peak memory by 82.74% and tracked operations by 524,123,904. Neither was promoted, for the wall-time reason in section 5. A43 is retained as a memory-reduction module, not as a scoring candidate.

7. Negative results worth other people's time

These are reported because they are cheap for someone else to repeat badly, not because they are proofs. Each closes a *tested implementation family*, not a mathematical class.

- **Shared-reference Taylor evaluation of a heteroscedastic mixture state.** Expanding component covariance maps around the pooled-within covariance shrank covariance offsets substantially (at $K=64$, layer 29: 0.574 to 0.357) while approximation error did not improve ($4.00\text{e-}3$ to $5.41\text{e-}3$). The error is mean-offset dominated, and the component-mean offsets are structural: increasing K works precisely by separating component means, so one shared reference gets worse as the representation gets more expressive.
- **Low-rank Hermite / direct-diagonal extraction.** Truncating the component covariance to rank r gave relative errors $2.16\text{e-}1$, $5.44\text{e-}2$, $6.73\text{e-}3$, $7.86\text{e-}4$ at $r = 4, 16, 64, 128$. The accuracy gate is about $1.5\text{e-}3$, so only rank 128 passes; the affordable rank under the cost budget was about 4.4. At rank 128, $2n^2r$ is approximately n^3 — the low-rank route has become the dense route. This closes the tested construction, not every possible algorithm for that diagonal.
- **Layer-31 independent anchor corrections.** The propagated centre sits near 0.65% error where roughly 0.45% is break-even after compute cost, leaving about 0.3% of usable headroom. No candidate cleared the downstream-weighted replacement threshold on untouched networks.
- **Partial-MUB and companion-basis hosted controls.** Measured hurdles were far above the projected ones at every basis count tested ($1.2841x$ vs $1.0670x$ at 96 bases).
- **Learned sign/scale models and handcrafted weight features.** No demonstrated complete-score value after actual compute cost was charged.

The honest summary of the failed program is not that a compact joint state does not exist. The local evidence suggests it does — the representation reaches error about $1.78\text{e-}3$ at $K=1536$. The failure is that every tested way of *evaluating* that state loses either accuracy or cost.

8. Reproducing the submitted estimator

ARC already holds the tarball uploaded for submission ID 320802; that upload is the object of record. The matching source is published at:

```
github.com/SkyeNygaard/arc-whitebox
  arc_whitebox/submissions/production_baseline_320802/ (estimator, kernel, asset, archive)
  whestbench/phase1_320802.json (machine-readable binding record)
```

Artifact	SHA-256
submission.tar.gz	77be0e8865b2aeee6c6c16314cac4d38496efefed6b2b758f75bc3033bb6b7bc
estimator.py	f1e32ce44fe43b53eba3f70f9cf6383da588ec1bbb3d82c047edbc916a98d8df
fast_matmul.py	fb1b93cb625b66ce5f26220ea3b6b685dbb9887d50f8756cfa9426577d45085
kerdock_mub5_seed3.npz	58eac1b69707b204d00f6d50cf4e1996b1fcd566154ec93a7ecb5668c1acbfad
dev cohort official_phase1_mini	5b00938b6bd809fe80acef08772c5654edf467863225ca9e304b76c779ecf433

```
python scripts/check_competition_release.py
whest validate-package arc_whitebox/submissions/production_baseline_320802/submission.tar.gz
```

Environment pinned to the machine the measurements were taken on: Python 3.12.13, whetbench 0.13.0, flopscope 0.9.1, numpy 2.4.6, macOS arm64, 12 logical cores, BLAS pinned. The archive validates with the public CLI and its manifest hashes match its members.

Provenance boundary, stated because it is a real limitation. The published archive is a frozen re-package of the production estimator, cut on 2026-07-30 one day after the upload. I identify it as the estimator behind #320802 by its *recorded operating point* — the instrumentation bundle records the then-current position as adjusted $1.55e-7$, raw MSE $2.416e-7$, effective compute $1.745e11$, compute multiplier 0.64154, which matches the graded page field for field and matches no other local package — and not by a byte comparison against the uploaded tarball. Separately, an earlier archive in this lineage shipped a manifest whose declared estimator.py hash disagreed with the bytes it actually contained; that defect is recorded in the repository rather than quietly fixed. The archive published here is internally consistent.

9. Limitations

- Public-split numbers only. The 50-MLP private split is sealed, so the final rank of #320802 is undetermined.
- Layers 1–30 are zeros. This is a final-layer estimator, not a full activation-profile estimator (§3.7).
- Specialized to width 256, depth 32; returns zeros otherwise.
- Residual wall time is hardware-dependent, and all local timings come from one machine. The single available calibration point suggests the official grader is about 11% slower.
- The basis-count ablation is an archived exposed-split projection, not an official measurement, and its frozen test weights were float16.
- The negative results close tested implementation families, not mathematical classes. I do not claim an information-theoretic lower bound anywhere in this document.
- Separate external-review materials in this project discuss near-optimality of the Kerdock design within a static, network-independent cubature class at a fixed node budget. Those claims are conditional on a computer-assisted interval certificate that has not been independently reconstructed or human-reviewed, and they are deliberately not used to support anything in this write-up.

10. Use of language models

Stated plainly, per the challenge guidance. Multiple LLM agents were used heavily throughout this project: for ideation, for implementation of the estimator and its kernel, for the experiment scaffolding and instrumentation bundles, for proof attempts in the adjacent theoretical work, for code review, and for drafting this write-up. The research direction, the promotion and rejection decisions, and the evidence labels are mine, but the code is substantially model-written.

What that means for the reader: **LLM assistance is not independent verification of anything here.** The claims I have personally checked are the archive hashes, the whet validate-package results, the graded figures read directly from the AICrowd submission page, and the operating-point match that binds the archive to the submission ID. The mechanistic account in section 3 is my reading of code I did not write line by line; it is consistent with the measured FLOP counts and the frozen manifests, but I flag it as a reading rather than a proof. Where numbers in sections 6 and 7 come from archived agent-run experiments that I have not re-executed end to end, they are labeled as archived or projected in place.

Submission ID **320802** | ARC White-Box Estimation Challenge 2026, Phase 1 | skye_nygaard | 2026-08-03