

ARC White-Box Estimation Challenge 2026, Phase 2: algorithmic-contribution write-up

mainguyenanhvu (AICrowd)

2026-10-28

Participant: mainguyenanhvu (AICrowd). **Paired submission:** #337552, graded in Phase 2 at 2.941084842405988e-09 (primary score, lower is better). Host report: final-layer MSE before the cost multiplier 1.921e-08, mean_score_multiplier 0.15312537944644647, 100 of 100 networks completed, n_failed_mlps 0, residual median 0.327 s and maximum 0.333 s (cap 0.400 s), predict wall mean 89.6 s and maximum 94.2 s (cap 120 s), no residual, wall or budget exhaustion. Graded 2026-10-21 (ledger READBACK 2026-10-21T16:40:02Z). On the public board read 2026-10-21T19:35:53Z the account ranks 39 of 307 at the displayed 2.9e-09; that row references #337504, which displays the same rounded value. Sources for every number: §I.7 and Appendix S.

Version. Refreshed 2026-10-28, after the Phase-2 submission deadline (2026-10-21T23:59Z). It supersedes the 2026-10-17 text (WRITEUP_phase2-algorithmic-contribution.v20261017.md, which described #335249 and its successors), kept unchanged. **Part I** (§I.0–§I.7) is new and describes the paired #337552. **Part II** (§0–§10, the references and Appendix A) is the 2026-10-17 text about our earlier, independently engineered lineage, re-audited claim by claim for this version; its measurements and negative results stand, but it does not describe the paired file.

In one paragraph. #337552 is the public e30_a estimator of team LordProcciuto (MIT; itself derived from team 504aldo’s V29, MIT) with four changes. Run as published, e30_a graded 3.1163015334141984e-09 for us (#337281), the same displayed score as its authors’ #335981. (i) We added our own learned corrector, a recurrent message-passing network over the engine’s per-neuron state that is carried layer to layer through the billed weight products (our code and training). (ii) We switched on e30_a’s own source-column retirement threshold (X_DEAD_TAU_B, default off) at 2.5, a one-line setting of public code. (iii) We ported, verbatim and with its author’s coefficients, Tristan Miano’s fitted covariance-defect dictionary (MIT; his ALG054 algorithm). (iv) We wrote an op-batched evaluation of that dictionary. It cut the logged operations of the stacked file from 24,118 to 22,321, which is what let the stack pass the host’s 400 ms residual cap: the unbatched stack (#337520) lost two networks to the host’s residual and time limits and graded 0.035849. Against e30_a as published, the paired file scores $\times 0.9438$: final-layer MSE $\times 0.9348$ (2.055e-08 $\rightarrow$ 1.921e-08) at a multiplier $\times 1.0096$ (0.15167 $\rightarrow$ 0.15313). We also report the measurement that made us switch lineage (§I.3): our own chain held a 5.6% lower final-layer MSE than e30_a but billed 29% more FLOPs, so on this score it lost by $\times 1.22$.

Reading guide. §I.0 states the claims and the score history of the paired lineage. §I.1 separates the reference method from our changes, line by line, and states the licences. §I.2 describes each change with its local and live measurement. §I.3 is the parity measurement against our own chain. §I.4 lists the negative results of this lineage. §I.5 gives the gap to the frontier, §I.6 reproducibility, §I.7 the sources of Part I. Part II follows.

Conventions. Every graded number is a read of the AICrowd grader: the submission ledger (.claude/loop/submissions.log, timestamped READBACK rows, full precision) or the host report (sources/host-report- $\langle id \rangle$.json, results.aggregates.public, four significant figures). “Multiplier” is the host’s mean_score_multiplier (the mean of $\max(0.1, C_m/B)$ over the 50 public networks). “Local paired” means our 6-network client-server gate (MLPs 500, 505, 508, 512, 515, 520) against a fixed control file; it is a local number, not a grade. In Part I, “plan@0afe76f :N” is line N of .claude/plans/PLAN_arc-whitebox-master.md as committed at 0afe76f3 (2026-10-22). Part II’s

citation convention is stated at its start.

Part I — the paired submission #337552

§I.0 What is claimed, and the score history

Nominated rows. The plan designates PICK-1 #337552 and PICK-2 #337504 for nomination (plan@0afe76f :23822). #337552 is the single submission paired with this document.

Score history of the paired lineage. All rows are graded; the MSE and multiplier columns are host-report reads (no host report was captured for #337480 and #337487).

Graded id	What changed (detail in §I.2)	Graded score	Final-layer MSE	Mult.	Ledger READBACK
#337281	e30_a as published, notice header added	3.116301533414128055e-08 09	0.15167		2026-10- 20T06:28:03Z
#337347	+ all-layer per-neuron corrector (MP, trained on 714 networks)	3.079949710728236012e-08 09	0.15311		2026-10- 20T15:02:06Z
#337391	recurrent- message corrector (gmp, 714 networks) in place of MP	3.044289332182434974e-08 09	0.15425		2026-10- 20T19:58:02Z
#337394	the same corrector trained on 300 networks	3.054106599118918080e-08 09	0.15425		2026-10- 20T20:30:09Z
#337478	gmp + source- column retirement X_DEAD_TAU_B = 2.5 (“b25”)	3.019056711418491079e-08 09	0.15257		2026-10- 21T06:26:04Z
#337480	gmp + retirement at 2.7 (“b27”)	3.0330662247215323e- 09	—		2026-10- 21T06:46:04Z
#337487	b25 with importance- ranked column choice (“w25”)	3.020693196522859e- 09	—		2026-10- 21T07:52:03Z
#337489	gmp retrained 12 epochs, width chosen on a held-out proxy (“gmplean”), no retirement	3.033900358519600924e-08 09	0.15372		2026-10- 21T08:30:03Z

Graded id	What changed (detail in §I.2)	Graded score	Final-layer MSE	Mult.	Ledger READBACK
#337500	gmplean + b25	3.008937268503012980e-08	0.15203		2026-10-21T10:14:03Z
#337504	gmp + b25 + covariance-defect dictionary, unbatched (PICK-2)	2.946413152164796021e-08	0.15342		2026-10-21T10:46:05Z
#337520	gmplean + b25 + dictionary, unbatched, 24,118 ops	0.035848942238476025	networks failed)	—	2026-10-21T13:46:03Z
#337538	gmplean + b25 + dictionary on layers 10-14 only	2.974297451808716953e-08	0.15231		2026-10-21T15:24:03Z
#337546	gmp + b25 + dictionary on layers 10-14 only	2.983989601539886052e-08	0.15285		2026-10-21T15:56:03Z
#337552	gmplean + b25 + op-batched dictionary from layer 2, 22,321 ops (paired)	2.941084842405988021e-08	0.15313		2026-10-21T16:40:02Z

From #337281 to #337552 the whole gain is accuracy: final-layer MSE $\times 0.9348$, multiplier $\times 1.0096$, score $\times 0.9438$. All-layer MSE moved from $7.986e-09$ to $7.515e-09$ (host reports).

What is claimed as ours.

1. The recurrent-message corrector (gmp): its design, our training code (research-phase2/qw530/train_gmp.py, whose docstring reads “Our own code”), its training on the organiser’s public split, and its evaluation inside the estimator as billed flopscope operations (§I.2 C1).
2. The op-batched evaluation of the covariance-defect dictionary (research-phase2/qw595/batch_cov.py, shipped at estimator.py:4240–4340), and the measurement that it, not the dictionary, decides whether the stack survives the residual cap (§I.2 C4, §I.4 row 59).
3. The selection measurements: which public knob setting, which dictionary depth and which corrector size transfer to the grader, each sent as a pre-registered graded row (§I.0 table, §I.4).
4. The parity measurement against our own earlier chain (§I.3).
5. Everything in Part II.

What is not ours. The estimator engine is e30_a, unchanged in 3,995 of its 3,997 lines (§I.1). The retirement mechanism behind X_DEAD_TAU_B is e30_a’s; we chose only its value. The covariance-defect dictionary, its 32 fields and its fitted coefficients are Tristan Miano’s.

§I.1 Reference method versus our changes, and licences

The organiser’s forum ruling of 2026-10-05 (topic 18240, staff reply, recorded at plan@0afe76f :17415 and :21903) allows a participant’s public code to be used as a base when its licence permits it, provided the licence notice is kept in the shipped files, the write-up states what was started from and what changed, and the participant’s own changes are substantial. We read the prize eligibility of a mainly derived entry as being at the organisers’ discretion (plan@0afe76f :21903). This section is that statement.

The reference method. e30_a is LordProcciuto’s best Phase-2 submission (#335981, displayed 3.1163e-9), published with an MIT licence carrying both the 504aldo and the LordProcciuto copyright notices (research-phase2/qw462/src/LICENSE; README “Provenance”). Its README describes it as “a modified version of 504aldo’s public V29 estimator” whose changes are “calibration knobs and cost trims on top of V29”. The engine is therefore the factored $K=3$ cumulant-propagation chain described in Part II §2.1 (504aldo V17–V29, building on ARC’s reference [3]), with LordProcciuto’s cost trims, dead-neuron retirement and a refitted 87-feature mean-correction table (the shipped file’s header comment, estimator.py:34–35). The published estimator.py has sha256 6ddd97a96c30397683815c530bfb422f0a49ffdea49e4f7827149ed0a14729b0 (README, and recomputed from qw462/src/estimator.py).

What we changed, measured by diff. Against the published file (3,997 lines), the paired file (4,347 lines) adds 350 lines and changes 2. Nothing is deleted.

Lines in the paired estimator.py	Content	Origin
1–28	MIT notice (504aldo and LordProcciuto), as a comment	licence text of the reference
349–350	import sys, byte-order constant used by the corrector’s weight decoder	ours
645–770	recurrent-message corrector: feature list, trained weights as base64 data, forward pass in billed ops	ours (C1)
990 (changed)	X_DEAD_TAU_B default "0" → "2.5"	setting of e30_a’s knob (C2)
1002	X_FCOV switch for the dictionary (default on)	ours
2456, 3425–3435, 3673–3674, 3815–3819; 3378 (changed)	capture of per-layer engine state for the corrector, and the call that applies it	ours (C1)
3668, 3688–3691	hook that adds the dictionary’s covariance delta to C and its diagonal	ours (C3 hook)
4174–4178	attribution comment for the dictionary	ours
4179–4239	covariance-defect dictionary: coefficient tables, fused specification, per-name kernels	Tristan Miano, verbatim (C3)
4240–4340	op-batched evaluator of the same dictionary	ours (C4)

Licences, and two defects in the graded file. The file carries the full MIT licence text with the 504aldo and LordProcciuto notices (estimator.py:1–27) and the line “Copyright (c) 2026 Tristan Miano and contributors” with the licence name at the dictionary block (:4175–4176). Two defects stay in the graded file, which we leave as submitted. First, the header line 3, written for the as-published #337281, says “the code below this header is byte-for-byte the published file”; for #337552

that is false, and the table above is the correct account. Second, the MIT permission text for Miano’s notice is not repeated at the dictionary block. Appendix L reproduces it in full from his repository’s LICENSE (research-phase2/qw582/pub/LICENSE). Any released copy of the code will carry all three notices with the full text.

Code from repositories without a licence. None. The gmp corrector shares its idea family with the learned local-error corrector of Part II §2.3 C1 (idea: Hisernberg V41 [12]); its code, features and training are ours.

§1.2 The changes, each with its local and live measurement

“Local paired” ratios are geometric means over the 6-network gate against the stated control file. Three of the six gate networks (500, 505, 520) lie inside the corrector’s training range (public ids 70–799 in qw530/feats; qw540/run.sh:5 states 200–799), so the local ratios of corrector changes are optimistic (qw540/run.sh:5); the live ratio is the measurement.

C1. Recurrent-message corrector (gmp; ours). A learned per-neuron correction of each layer’s mean, $c_l = \phi_l \cdot (c_{l-1} W_l) + g_l$, where g_l is a small network over the engine’s per-neuron statistics plus learned message channels $m1_l = W_l^T P_{l-1}$ and $m2_l = (W_l \circ W_l)^T P_{l-1}$, so information moves across neurons through the network’s own weights (qw530/train_gmp.py:1–4). The correction is carried layer to layer through billed weight products; every operation is a metered floscope call (estimator.py:645–647). Training used the public full split.

- *Predecessor, MP (all-layer corrector without learned channels).* Local paired score $\times 0.9829$ against e30_a as published; live #337347 / #337281 = 0.9883 (ledger READBACK-LESSON 337347, 2026-10-20T15:31:48Z).
- *gmp, 714 training networks.* Local paired score $\times 0.9612$ against e30_a as published (MSE $\times 0.9449$, multiplier $\times 1.0172$; plan@0afe76f:23071). Live #337391 / #337347 = 0.9884 and #337391 / #337281 = 0.9769 (ledger READBACK-LESSON 337391, 2026-10-20T20:02:13Z).
- *Training-set size.* 300 networks (#337394) graded $\times 1.0032$ of the 714-network corrector.
- *gmplean (12-epoch retrain, width chosen on a held-out proxy).* Against e30_a as published, local paired score $\times 0.9553$ and MSE $\times 0.9424$ (qw563/out/verdict.txt); live score $\times 0.9736$ and final-layer MSE $\times 0.9606$ (#337489 against #337281). Against the gmp corrector, live score $\times 0.99659$ (#337489 / #337391), with final-layer MSE unchanged at $1.974e-08$.

C2. Source-column retirement at $X_DEAD_TAU_B = 2.5$ (setting of e30_a’s knob). e30_a retires “dead” neurons, whose standardised pre-activation mean lies below $-\tau$ ($\tau = 2.85$); its knob $X_DEAD_TAU_B$ gives the columns of the born sources their own threshold τ_B ($0 < \tau_B < \tau$; default 0, off) (qw462/src/estimator.py:771–773, :830–834). On the gmp file, $\tau_B = 2.5$ gave local paired score $\times 0.9949$ (MSE $\times 1.0069$, multiplier $\times 0.9881$) and live #337478 / #337391 = 0.99171 (ledger READBACK-LESSON 337478, 2026-10-21T06:56:11Z). $\tau_B = 2.7$ gave local $\times 0.9966$ and live $\times 0.99631$. Stacked on gmplean, live #337500 / #337489 = 0.99177: the two legs are independent.

C3. Covariance-defect dictionary (Tristan Miano, MIT; ported verbatim). At each layer it adds to the propagated covariance C and its diagonal a fitted linear combination of 32 fields built from marginal, mixed and quadratic third-cumulant terms, with coefficients in three depth bins (li // 5), fitted by its author on 32 public training networks (estimator.py:4174–4178). The block was copied from his file submissions/phase2_alg058_v001/estimator.py; his write-up ALG054 ported it to e30_a (fitted earlier in his T170; his #336778, 3.0332910471e-09; qw582/pub/ALG054-V000.md). We copied it by a script that asserts the block’s boundaries (qw582/patch.py; the same extraction in qw583/patch.py). On the gmp + b25 file it gave local paired score $\times 0.9648$ against gmp (MSE $\times 0.9716$, multiplier $\times 0.9930$; qw582/out/verdict.txt), i.e. $\times 0.9698$ against b25, and live #337504 / #337478 = 0.97594, about 80% of that local log-gain (ledger READBACK-LESSON 337504). This is the largest single step of the lineage, and it is not ours.

C4. Op-batched evaluation of the dictionary (ours). The per-name kernels issue many small floscope calls per layer. On the host, residual time grows with the number of logged operations (Part II §1, constraint 1), so the unbatched stack failed: #337520 (gmplean + b25 + dictionary, 24,118

operations on the census network) lost two networks: one crossed the residual cap at 0.401 s and one failed with the host’s time-exhaustion flag at 107.2 s of predict wall. It graded 0.035849 (host report, per_mlp indices 37 and 42; ledger READBACK-LESSON 337520). The batched evaluator stacks the dictionary’s index, coefficient and Hermite-basis tensors at setup and evaluates each depth bin with a few batched calls (estimator.py:4240–4340; its comment says “~114 fewer client ops per layer”). It is the same mathematics in a different float32 summation order, not a bit-identical rewrite: on the two census networks the batched all-layer arm differs from the unbatched one by a relative 4.755e-03 in mse_final (qw595/out/verdict.txt). Applying it from layer 2 on (arm B2) gives 22,321 operations against 24,118, a predicted residual median of 0.312 s with predicted loss probability 0.000 (sources/resid-23ea4c0215e46496.verdict), and a local paired score $\times 0.9592$ against gmp (MSE $\times 0.9678$, multiplier $\times 0.9911$). A final repair removed a numpy import from setup code, with setup tensors checked identical (qw596/edit.py; qw596/out/verdict.txt “IDENT OK”).

Live. #337552 graded 2.941084842405988e-09: $\times 0.96610$ against gmp (#337391), $\times 0.97745$ against its parent gmplean + b25 (#337500), and $\times 0.99819$ against the unbatched dictionary on the gmp file (#337504). Host residual median 0.327 s, maximum 0.333 s, no network lost.

§I.3 Parity measurement: our chain against the public e30_a

Before switching lineage we measured our own best chain, clA (#336874: the r352 lineage of Part II §0.1 plus a parameter-free (3,1) closure term at layers 8–14; plan@0afe76f :21023), against e30_a run as published. Host reports, 50 public networks:

	Graded score	Final-layer MSE	Multiplier	Logged operations (census)
clA #336874 (ours)	3.808302189364595e1.94e-0809		0.19628	24,810
e30_a as published #337281	3.11630153341419842.055e-0809		0.15167	19,857
ratio clA / e30_a	$\times 1.222$	$\times 0.944$	$\times 1.294$	$\times 1.249$

The census counts are local (research-phase2/qw393/out/census_A.json for clA, MLP 500; READBACK-LESSON qw462-e30a-parity, 2026-10-19T23:23:39Z, for e30_a). The local 50-network paired gate read the same picture before any slot was spent: e30_a / clA score 0.806784, MSE $\times 1.0446$, C/B 0.14953 against 0.19489 (same ledger row). Our chain carried better content per network, but it paid 29% more FLOPs for it, and on MSE $\times \max(0.1, C/B)$ the cost leg dominated. Our planning record reads the public engine’s advantage as its cost structure (dead-neuron removal, a fixed-basis old level, regenerated κ_4 ; plan@0afe76f :21908), which our cost engineering in Part II had not reached. We therefore moved the remaining slots to e30_a and spent our own work on the accuracy leg (C1) and on the operation budget (C4).

Correction. Our planning note for this section quoted e30_a’s MSE as 1.975e-08 and the bill ratio as $\times 1.27$ (plan@0afe76f :21917); the host reports give 2.055e-08 and $\times 1.294$, used above.

§I.4 Negative results of the e30_a lineage

Numbering continues Part II §8.

#	What was tried	Measured number	Reading
59	Dictionary stacked unbatched at 24,118 operations (#337520)	graded 0.035849; 2 networks failed (one at residual 0.401 s, one time-exhausted at 107.2 s wall)	the residual cap is an operation cap; the local resgate printed PASS but predicted loss probability 0.174 (qw583/out/resgate.out; ledger READBACK-LESSON 337520)
60	Dictionary restricted to layers 10-14 to save operations	#337538 $\times$ 1.0095 and #337546 $\times$ 1.0128 against #337504	depth truncation loses most of the dictionary's gain; batching, not truncation, fixed the budget
61	Importance-ranked choice of retired columns (w25)	#337487 $\times$ 1.0005 against b25 #337478	the count of retired columns matters, not which ones
62	Stronger retirement on gmp, τ_B 2.2 and 1.9 (local)	paired score 1.0015 (MSE $\times$ 1.0247, C/B $\times$ 0.9774) and $\times$ 1.0926	MSE rises faster than the bill falls below τ_B = 2.5 (plan@0afe76f:23349)
63	Seed fills on e30_a and on gmp	#337353 / #337281 = 0.999996; #337405 / #337391 = 0.999998	the seed axis is flat on this engine
64	Retrain of our earlier terminal corrector on e30_a's final layer (e30a-lt)	offline $R_{\text{new}} / R_{\text{ship}}$ = 0.9985 against a 0.98 bar	KILL before any slot (ledger READBACK-LESSON qw463-e30a-lt, 2026-10-20T02:08:14Z)
65	Port of our graded (3,1) closure term (clA) into e30_a	census 20,474 operations against a 19,900 limit, C/B 0.15398 against 0.1496	the term costs more bill than the engine affords (ledger 2026-10-20T01:29:36Z)
66	Corrector refit judged on the gate networks (gmplean)	against e30_a: local MSE $\times$ 0.9424, live $\times$ 0.9606; against gmp, live MSE unchanged	three of six gate networks are in the training range; a corrector's local gain is not a live measurement

Instrument finding: an 8 GB memory gate that measured its own clamp. Our in-process memory gate ran the floscope server and the estimator in one process under one 8,192 MB address-space limit, and every e30_a variant read a peak of 8,185–8,191 MB and failed. Run through the client-server path the host uses, the estimator's own peak was 1,576 MB (ledger READBACK-LESSON qw504/qw503, 2026-10-20T04:45:40Z). The gate was rebuilt on the client path before any e30_a file was sent.

Transfer by family. The propagation-side change (C3) kept about 80% of its local log-gain live; the output-side correctors kept about 70% (MP) and about 60% after a refit (gmplean, score against e30_a). Cost-side changes (C2) transferred at or above their local value.

§I.5 The frontier

On the 2026-10-21T19:35:53Z board, rank 1 (marius_binner) shows $1.3\text{e-}09$ with a secondary (final-layer MSE) of $1.1\text{e-}08$; we show $2.9\text{e-}09$ and $1.92\text{e-}08$ (sources/leaderboard-1429-20261021T193553Z.json). On these rounded values the gap is $\times 2.2$ in score and $\times 1.7$ in final-layer MSE, so most of it is accuracy. Part II §7 decomposes the earlier gap with host telemetry; the same method applies here.

§I.6 Reproducibility of #337552

Environment. The host graded #337552 under whetstbench 0.16.1, flopscope 0.12.3 (client and server) and Python 3.10.21 (host report runtime_environment). Local gates ran the client-server path with the vendored flopscope 0.12.3 server (research-phase2/qw502/vendor/) under the conda env compl0p2.

Code of record. research-phase2/qw596/ship_e30a_gmplean_b25_fcovbatch2/estimator.py, one file, 4,347 lines, sha256 98193588312b3117e76a478f2ealb51435380086f8a2329f6843bfd1c772cc26 (recomputed 2026-10-28), registered as candidate c10211546561460 (.claude/loop/state/candidates/arc.tsv)

Regeneration chain. Each step is a script that edits its parent with asserted unique matches: qw540 (e30_a + gmp, 714 networks) → qw562 (X_DEAD_TAU_B = 2.5) → qw563 (gmplean corrector) → qw580 (gmplean + b25) → qw583 (+ dictionary, via qw583/patch.py, the extraction of qw582/patch.py) → qw595/patch.py (batched evaluator, arm B2) → qw596/edit.py (setup without numpy). The final gates, in order: setup-tensor identity, operation census (22,321 on MLP 505), 6-network paired gate, the official scorer's smoke run, wall gate, residual gate and the client-path memory receipt (peak about 1,577 MB) (qw596/out/verdict.txt, receipts.out, resgate.out, smoke.out).

Availability. As in Part II §9: the cited paths will be released with the solution code if this entry is prize-eligible, and are available to the organisers on request. Training data is the organiser's public split only.

§I.7 Sources for Part I

Kind	Path	Used for
Host reports	sources/host-report-{337281,337347,337391,337394,337478,337489,337500,337504,337520,337552}.json	MSE, multiplier, residual, wall, smoke
Ledger	research-phase2/qw211/hostreport.py table on 2026-10-28	
	.claude/loop/submissions.log (READBACK and READBACK-LESSON rows at the timestamps given)	full-precision grades, local paired ratios
Plan	.claude/plans/PLAN_arc-whitebox-master.md at commit 0afe76f3	nomination, ruling record, pre-registrations
Reference code	research-phase2/qw462/src/(e30_a estimator.py, README.md, LICENSE, WRITEUP.md)	§I.1
Miano capture	research-phase2/qw582/pub/(alg058_estimator.py, ALG054-V000.md, LICENSE, README.md)	§I.1, §I.2 C3, Appendix L
Build and gates	research-phase2/{qw530,qw540,qw562,qw563,qw580,qw582,qw583,qw595,qw596}/(run.sh, out/verdict.txt)	§I.2, §I.6

Kind	Path	Used for
Board	<code>sources/leaderboard-1429-20261021T193553Z.json</code>	rank 39 of 307, \$1.5

Appendix L — licence text for the covariance-defect dictionary

Reproduced from `research-phase2/qw582/pub/LICENSE` (Tristan Miano, `arc-whitebox-estimation-2026`):

MIT License

Copyright (c) 2026 Tristan Miano and contributors

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

The `e30_a` licence (504aldo and LordProcciuto) is reproduced in the paired file's lines 1–27 and in `research-phase2/qw462/src/LICENSE`; the 504aldo text alone is in Part II §2.1.

Part II — the earlier lineage (2026-10-17 text, re-audited)

This part is the 2026-10-17 text. In it, “the paired submission” or “paired artifact” means #335249, the submission paired with that earlier text; the submission paired with this document is #337552 (Part I). “Plan :N” is line N of the plan file as committed at `efd07351` (2026-10-11) for $N \geq 32,000$, at `ecd7d960` (2026-10-17) for $22,000 \leq N < 32,000$, and at `68f45b85` (2026-10-13) for $N < 22,000$; “plan §3.xx” is a section of the 2026-10-11 plan. Corrections found by the 2026-10-28 audit are applied in place.

§0 The earlier lineage: #335249 and its record to 2026-10-17

Update 2026-10-17: current record lineage (r352). The method record of this lineage is #335699 (old-tier rank 352 on the paired core): graded `3.855002e-09`, host `final_layer_mse` `1.964e-08` x `mean_score_multiplier` `0.1963` (plan :21693; `sources/host-report-335699.json`; `READBACk` plan :21566). Its same-seed parent #335575 graded `3.915495e-09` (plan :21566). The paired-submission text below describes the earlier #335249 artifact and is kept as the history of the lineage; the id paired with this document is #337552 (Part I).

§0.1 Old-tier rank cut (measured)

- **r352 vs r384.** Local 50-MLP paired ratio 0.98694 against the same-seed s1 base (0.934813 vs 0.947184; plan :21473). Live ratio 0.98455 (#335699 vs #335575, plan :21566); live/local transfer 0.9976 (plan :21595).
- **r336 (local, not submitted).** 50-MLP paired ratio vs c1 0.942239 (seed 1) against r352's 0.934813, i.e. +0.79% worse (plan :21877, :21867); the old-tier rank axis stops at r352.
- **r320 (local, not submitted).** Per-MLP ratio to r384: MLP 508 1.0417, MLP 512 1.0721 (plan :21634); MSE $\times 1.035 / \times 1.093$ at a bill of $\times 0.916$ (plan :21381). On the 16 MLPs common to all tables, $S(320)/S(384) = 0.940162$ against $S(352)/S(384) = 0.931498$ (plan :21526, qw334). r352 is the better cut.
- **Where the loss sits (local, not submitted).** An exact-diagonal (FITC) oracle on the rank cut recovers 0.009 (MLP 508) and -0.027 (MLP 512) of the loss (plan :21634, qw333), so the loss is in off-diagonal basis quality, not the diagonal.
- **Sketch variance (local, not submitted).** Averaging 4 sketch seeds gives mean ratio m4/m1 0.9919 (QW335_VERDICT KILL mean_ratio=0.9919, job log .claude/loop/state/jobs/arc/logs/arc-20261013T051651Z-b7216f.log line 1067): seed variance of the sketch is under 1% of MSE.
- **Seed transfer.** Fill #335728 (seed 9, r352) graded 3.876106e-09, +0.55% vs #335699; the local s9-s1 delta was +0.52% (plan :21695; sources/host-report-335728.json).
- **Best-by-noise grade (2026-10-17).** Fill #336629 graded 3.843810e-09 (ledger READBACK 2026-10-17T03:46Z), -0.29% against the record #335699/#335921 at 3.855002e-09. Same lineage and equal multiplier, so this is seed noise inside the $\pm 0.65\%$ band of the same-lineage seed fills, not a method change. The method claim stays based on #335699/#335921 (bit-identical 3.855001616e-09).

§0.2 Negative results added 2026-10-17 (measured, none shipped)

- **qw374 ablation ladder.** No removal pays: ratios against the control for nofeed 1.0747 and noconf 1.0904; oldlam 0.99845 (n_ops 24782, C/B unchanged); v25off 1.0000; notrim census 26141 ops (above 24900); v17off saves 2.3% of F at an MSE cost beyond +1%. No component is overpriced at C/B 0.196 (plan :22079).
- **qw375c variance collapse.** An early-layer order-3 cut makes the layer variance diverge from control by layer 6 and collapse by layer 9: sigma reaches its 1e-5 clip, alpha 2e10, and the Wick Horner polynomial overflows to NaN. This is the lever, not a guard bug (plan :22139).
- **qw379 refit.** Refit of the layer 12-15 rows of the learned local-error corrector (LT) on the scored objective: held-out (MLPs 548-555) mean ratio 1.0528 against the 0.97 gate; paired 500-515 mean 1.0528, max 1.109. Worse than parent (plan :22293).
- **Noise floor.** 36 same-lineage seed fills (r352ll) span about 3.8438e-09 to 3.8837e-09 (noise floor $\pm 0.65\%$; one forecast-trip artefact excluded), so differences below that are not evidence of a better method (ledger READBACK-LESSON lines 2026-10-16 to 2026-10-17).
- **False-trip census (qw384).** Over 3000 MLPs (30 graded reports $\times$ 100), the wall fail-safe tripped 8 times and no MLP exhausted the residual or wall cap (0 host exhaustions; research-phase2/qw384/census.tsv).

§0.3 Related work on the forum (read 2026-10-17)

Four other Phase-2 write-ups are public on the challenge forum: topic 18246 (<https://discourse.aicrowd.com/posted/2026-10-15>), 18248 (.../t/18248, 2026-10-15), 18249 (.../t/18249, 2026-10-16); a K=3 chain with a scale-mixture fourth-moment term, a rank-448 carry and a Strassen-Winograd product, reporting #333534 at 3.7357e-09 and 18250 (.../t/18250, 2026-10-16). The shared ingredients are low-rank carry, higher-order moment closure and fast matrix products; what this write-up adds is the measured cost-versus-MSE accounting and the negative results of §0.2.

Billing note. In Phase 2, C is flopscope FLOPs only; residual time is not billed and is a hard 0.4 s cap per network (see §1). The 10^{11} FLOP-equivalents per residual second applied to Phase 1 and the warm-up round, not to Phase 2 (page :150–:156).

Record artifact of the 2026-10-11 text. #335249 = artifact research-phase2/qw239/ship_c1sv5c5n5mf/ (four files, hashes in §9). Graded $3.920616e-09$ against a pre-registered point $3.922e-09$, band $[3.899e-09, 3.946e-09]$ (point miss -0.04% ; ledger READBACK 2026-10-10T20:23:53Z; plan :35629). Host report: final_layer_mse $1.925e-08$, mean_score_multiplier = mean_compute_utilization = 0.20363793016673298 over the 50 public networks (the per-network flops_used mean over all 100 is 0.2032), n_failed_mlps 0, residual median 0.369 s and maximum 0.378 s (cap 0.400 s), predict wall mean 92.4 s and maximum 97.0 s (cap 120 s) (research-phase2/qw211/hostreport.py table 335249).

Two clean sibling grades replicate its host behaviour. Both were built by the same chain and carry the same fail-safe:

- #335245 (ship_c1sv5c5mf: the paired core one Strassen level shallower, _LV_NB = 4): graded $3.933301e-09$ against point $3.935e-09$; 0 failed networks, residual max 0.380 s, 0 fail-safe trips (plan :35614).
- #335251 (ship_c1sv5mf: the #334966 core plus the fail-safe): graded $3.956043e-09$, which is the grade of #334966 ($3.956043398696704e-09$) to seven digits; 0 failed networks, 0 trips (plan :35636).

They are sibling arms, not byte-identical resubmissions. Between them they show three things: the fail-safe is value-neutral on the host (3 of 3 grades), no grade lost a network to a host cap, and within this family a local paired point predicts the live grade to better than 0.05% .

Score history of the earlier lineage (to #335251). Graded milestones only. “Mult.” is the host’s mean_score_multiplier (public 50 networks), except the first row, where it is the local C/B of the 2026-10-01 text.

Graded id	What changed (one line; detail in §2)	Graded score	Final-layer MSE	Mult.	Source
#331541	einsum-L5, the previous paired submission	$6.629788167328921e-08$ 9	$1.15e-08$	0.3135	earlier text §0
#333457	cost rungs Q-W69...Q- W102 on the same estimate (residual rank 8, early-born feedback rank, warm join sketch)	$5.131138818871522e-08$ 09	$1.57e-08$	0.2379	ledger 2026-10- 02T08:11:11Z
#333622	sliced active-set young tier (Q-W109 sl5q)	$4.739943405519405e-08$ 09	$2.07e-08$	0.2148	ledger 2026-10- 02T23:58:02Z
#334104	table-row variant on the sliced-tier core (qw139 arm_p4t2)	$4.709033087284286e-08$ 09	$1.68e-08$	0.2172	ledger 2026-10- 05T02:42:40Z

Graded id	What changed (one line; detail in §2)	Graded score	Final-layer MSE	Mult.	Source
#334496	Strassen-level-4 core at half the wall (lv4_wd, plan :33200): the corrector's paired parent	4.856404e-09	2.167e-08	0.2240	plan :33974
#334706	+ learned local-error corrector (C1)	4.516697e-09	2.006e-08	0.2251	plan :34091, :34096
#334826	+ alive-block restriction of dead-neuron contractions (Q-W219 X)	4.445216e-09	2.001e-08	0.2221	ledger 2026-10-08T12:44:16Z
#334862	corrector with hub and k4 features (k4b, 17 inputs)	4.288790404151405025e-0809	0.2228	0.2228	ledger 2026-10-08T19:20:45Z
#334881	fused term tables + per-site join Strassen (Q-W234 f1)	4.1629539781529517025e-0809	0.2162	0.2162	ledger 2026-10-08T21:12:26Z
#334932	per-site join-Strassen knapsack under 27,400 ops (c1)	4.129795578069123025e-0809	0.2145	0.2145	ledger 2026-10-09T03:03:15Z
#334950	+ A/P merge + sw15 + yq2 (C2-C4; bit-identical)	4.126764529678317025e-0809	0.2143	0.2143	ledger 2026-10-09T05:29:31Z
#334966	+ Strassen level 5 at the young-leg sites (v:5, C5)	3.956043398696704026e-0809	0.2054	0.2054	ledger 2026-10-09T07:46:29Z
#334991	+ covariance-sandwich level 5 (c:5) under the old wall trigger: 16/100 false trips	1.0071e-08	5.081e-08	0.2029	ledger 2026-10-09T10:37:10Z; plan :35118
#335245	v:5 c:5 + redesigned wall fail-safe (C6)	3.933301e-09	1.925e-08	0.2043	ledger 2026-10-10T19:43:38Z
#335249	v:5 c:5 n:5 + redesigned wall fail-safe (paired)	3.920616e-09	1.925e-08	0.2036	ledger 2026-10-10T20:23:53Z

Graded id	What changed (one line; detail in §2)	Graded score	Final-layer MSE	Mult.	Source
#335251	v:5 + redesigned wall fail-safe (replicate of #334966)	3.956043e-09	1.926e-08	0.2054	ledger 2026-10- 10T20:58:34Z

The MSE and multiplier columns are host-report reads for every row except #331541. Between #334862 and #335249 the final-layer MSE did not move (1.925e-08 to 1.926e-08): the whole $\times 0.9142$ of that stretch is cost. The accuracy gain of the whole history sits in the corrector rows (2.167e-08 $\rightarrow$ 1.925e-08, $\times 0.888$).

What is claimed as ours in the earlier lineage.

1. The op-count reading of the residual cap, its calibration from host reports, and the gate built on it (§1, §6 finding 5).
2. Three bit-identical op cuts, each with a unit test and a paired 50-network gate: the A/P leg-pair merge, the 15-op Winograd node sw15, and the pair-batched range finder yq2 (§2.3 C2–C4).
3. Re-spending the freed op budget on Strassen levels at three sites, with the measured wall cost that made it binding (§2.3 C5).
4. The redesigned wall fail-safe and the diagnosis that the old one fired on typical networks, read from per-network host telemetry (§2.3 C6, §6 finding 6).
5. Our implementation and training of a learned local-error corrector whose idea is Hisernberg’s, with its leave-MLP-out numbers (§2.3 C1).
6. Every cost engineering step before 2026-10-05 that is not 504aldo’s (§2.1). Measured paired against unmodified V29 on six networks, that step kept MSE ($\times 0.993$) and cut C/B to $\times 0.851$ (plan :32862–:32864).
7. The negative results (§3–§5, §8) and the instrument findings (§6).

What is not ours in the earlier lineage. Every term of the cumulant closure that sets the estimate is 504aldo’s: the $K=3$ chain, the memoryless κ_4 regeneration, the D21 feedback legs, the shared-basis and nested tiers, the adaptive λ and their fitted constants. So is the idea of applying Strassen-Winograd to these products (their V26–V29). The learned-corrector idea is Hisernberg’s. The shipped files contain code derived from 504aldo’s MIT-licensed repository and carry its notice (§2.1).

§1 Setting: 1024x16, $B = 2^{41}$, $C = F$, and three binding constraints

Phase 2 scores width-1024, depth-16 MLPs with a flop budget $B = 2^{41} = 2,199,023,255,552$ per network (host report `config_snapshot.flop_budget`; the challenge page lists “Budget / MLP 2.20 e 12”, `sources/challenge-page-jina-20260930T162953Z.md:154`). The per-network score is $s_m = \text{MSE}_{\text{final},m} \times \max(0.1, C_m / B_m)$ (page :190), averaged over networks, lower better.

Cost is FLOPs only. In Phase 2 the effective cost is $C = F$, the flopscope FLOP count (page :156). Time spent in `predict()` outside flopscope operations is residual time. It is not priced; it is “capped at **400 ms** per network, and crossing the cap fails that network through the zero-prediction fallback” (page :121). The page also states the change from Phase 1: “Residual PRICING is deprecated: λ is 0.0 and residual time is capped at 0.4 s instead, so $C == F$ ” (page :164). The host report agrees. On all 100 networks of #335249, `telemetry.effective_compute` equals `telemetry.flops_used`, and `mean_score_multiplier` equals `mean_compute_utilization` (0.20363793016673298).

Correction to the earlier text. The 2026-10-01 version described a local-harness run (Q-W55, 2026-09-09) whose multiplier included 10^{11} FLOP-equivalents per residual second. That is the Phase-1 cost model ($C = F + \lambda R$, page :123, :157). It does not describe the Phase-2 score, and

we no longer use it. What survives from Q-W55 is the floor: a bit-identical estimator billed at 6.28% of B returned a multiplier of 0.09999999999999999, i.e. the 0.1 floor, and billed at 25.02% it returned about 0.25 (research-phase2/calib/gate_qw55_8mlp_waste3.json, gate_qw53_8mlp.json). Our paired artifact bills 0.2036, so the cost axis can buy at most $\times 0.491$ before the floor binds.

The other two limits. A hard wall-clock cap of 120 s per network, and 8 GB of memory for the participant process on one physical core (page :158, :162, :250). Breaking any cap gives that network the zero-prediction fallback (page :147).

Three binding constraints shaped every cost step in this document.

1. *The residual cap is an op-count cap.* The host’s residual time grows with the number of logged flopscope calls, not with FLOPs (within-report correlation of residual with per-network FLOPs 0.076, plan :34991). From host reports we calibrated about 14.0 μ s per logged operation. That model predicted host residual medians of 0.378 / 0.381 / 0.377 s for three graded artifacts that read 0.375 / 0.376 / 0.377 s in the reads available at recalibration time (plan :34914; the final host-report medians are 0.375 / 0.375 / 0.378 s); the rate was set after those three host reads, so the agreement is in-sample. Per-operation rates across our host reports range from 13.1 μ s per operation (the first fit through the origin, plan :34140) through 13.6–14.5 μ s (plan :35039) to 15.2 μ s (#334966: 0.358 s host-report median over 23,515 operations, the 50-network gate maximum, plan :35102). Out of sample the 14.0 μ s law under-predicts: the paired #335249 (24,260 operations in the gate census of its c1sv5c5n5 core, plan :35091) was predicted at 0.340 s and read 0.369 s on the host (15.2 μ s per operation; prediction -8%). The send gate refuses predicted medians above 0.355 s, about 25,300 operations. A Strassen-Winograd level saves FLOPs but adds operations, so operations are the currency for further cost cuts (§2.3 C2–C5).
2. *The wall cap binds once operations are cheap.* One more level at the young-leg sites (v:5) cut billed FLOPs by 4.2% and raised predict wall by 32% (host medians 66.8 $\rightarrow$ 88.3 s, plan :35570). The estimator therefore carries a fail-safe that finishes the last layers with a Gaussian closure when the 120 s stop is near. Its trigger decides whether a level step can be sent at all (§2.3 C6).
3. *The floor.* At C/B = 0.2036, cost can deliver at most $\times 0.491$. J2W bills 0.1065 (host report #335031), and the other top rows 0.1168, 0.1175, 0.1315 and 0.1324 (host reports #335147, #333961, #335060, #335209), so most of what is left to gain must come from accuracy.

§2 The earlier-lineage artifact #335249

§2.1 Provenance and licence

Component of #335249	Origin	Our part	Evidence
Factored K=3 cumulant chain; memoryless κ_4 regeneration with per-layer λ table and adaptive scaling; rank-16 D21 feedback legs; age-gated shared basis; nested second tier; their fitted constants	504aldo V17–V25, MIT, topic 18218 [2]; builds on ARC’s reference [3]	port and later re-implementation; code partly derived (notice retained)	ship_c1sv5c5n5mf/qw73_core.py: 159 docstrings; plan :32853–:32861
Strassen-Winograd recursion on product families	idea: 504aldo V26–V29	our level-batched implementation sw_matmul_lb, and every later Strassen site	plan :32861 (0% verbatim lines in all 19 helpers, including sw_matmul_lb)

Component of #335249	Origin	Our part	Evidence
Cost engineering to 2026-10-05: sliced active-set young tier, batched Wick evaluation, warm-started join sketch, birth-graded feedback rank, bounded buffer pools	ours	all	plan :32861-:32864; estimator.py:26, qw91_core.py:537-550 (paired vs V29: MSE $\times 0.993$, C/B $\times 0.851$, score $\times 0.846$, 6/6 networks)
Learned local-error corrector (lt_weights.py, data only)	idea: Hisernberg V41 [12] (public repository with no licence for its own code)	our code from the published method description; our features and training	plan :33726-:33731; research-phase2/qw194/
alive-block restriction, fused term tables, per-site join-Strassen knapsack	ours	all	plan :34393; ledger 2026-10-08T21:09:13Z, 2026-10-09T03:03:15Z
A/P merge, sw15, yq2, level knobs, wall fail-safe	ours	all	\$2.3

Code provenance, measured. On 2026-10-05 an AST-normalised line match against every upstream estimator_v*.py found 117 of 644 code lines of qw91_core.py, the core that runs on the scored suite, verbatim upstream (18.2%). The upstream lines sit in predict (92 of 349 lines) and _dslices_batched (24 of 94), i.e. in the closure (plan :32852-:32861). That measurement predates the 10-07 to 10-10 work.

Licence. Every shipped code file (estimator.py, qw73_core.py, qw91_core.py; not the data file lt_weights.py) carries the following notice. The paired folder's estimator.py:1-25 and qw91_core.py:1-25 are the source of this copy.

Portions of this file are derived from 504aldo's whest-p2-cumulant-k3 (<https://github.com/504aldo/whest-p2-cumulant-k3>), used under the MIT License reproduced below.

MIT License

Copyright (c) 2026 504aldo

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE

SOFTWARE.

No third-party private code. A case-insensitive search of the paired folder for master-pipeline, masterpipeline and ccbiotech returns nothing (run 2026-10-11, `grep -ril`, exit status 1). The docstrings of `qw73_core.py:46-47` and `qw91_core.py:50-51` attribute arXiv:2605.05179 to “Hoogland et al.” That label is wrong. The paper is Wu, Lecomte, Winer, Robinson, Hilton and Christiano [3] (plan :34948, :34953). The graded files are left as submitted.

§2.2 Before 2026-10-01: the rebuild and the first cost rungs (unchanged in substance)

The accuracy path was first rebuilt rung by rung from 504aldo’s disclosed V17→V18→V19→V22→V24→V25 sequence. Each rung was checked against their reference implementation (`scout/whest-p2-cumulant-k3/`) with two controls before any gate ran. **C-EXACT**: with the new rung’s flag off, the port must reproduce the prior graded rung bit for bit. **C-REF**: on 504aldo’s three debug MLPs, the port’s rung-over-rung ratio must match their disclosed ratio. Two rungs (V17, V18) failed on first read. Our term table lacked terms present in 504aldo’s table and absent from ARC’s SIMPLE reference. Both were fixed before any gate (`decision_log.md` 2026-09-14T15:1xZ, 21:5xZ, 22:0xZ). The 2026-10-01 text, §2, gives the full rung table and band record.

Rung	Change	Gate rho [95% CI]	C/B (local)	Graded id	Graded score
V17	memoryless K ₄ regeneration, K ₄ →K ₃ feed	0.6419 [0.6285, 0.6555]	0.51518	#331042	1.2973e-8
V19	exact final-layer trim	1.0000014 [0.999998, 1.000005]	0.48302	#331051	1.2163e-8
V18	batched source stacks + D21 feedback (rank-16 thin legs)	0.83050 [0.81817, 0.84275]	0.49991	#331079	1.0323e-8
V22	age-gated shared basis + pre-activation λ table	1.03203 [1.02704, 1.03705]	0.40577	#331104	8.5405e-9
V25	nested second tier + adaptive per-MLP λ	1.00609 [1.00149, 1.01070]	0.38901	#331139	8.2262e-9
V26L1 / L2	1- and 2-level per-node Strassen- Winograd	0.99996 / 0.99991	0.37933 / 0.37094	#331392 / #331414	8.0242e-9 / 7.8433e-9
V27LB-L5	level-batched recursion, 5 levels	1.00004 [0.999818, 1.000257]	0.35465	#331469	7.5009e-9
einsum-L5	three einsum("kij,kcj >ic") sites routed through the batched recursion	0.9999938 [0.9998237, 1.0001665]	0.31345	#331541	6.6298e-9

Two findings from that period still carry the later work. First, a per-node recursion issues seven multiply calls per node, so its Python-level op count grows like 7^{level} . At three levels its residual read 0.397 s against the 0.4 s cap. Stacking the seven sub-products of one depth into a single batched call keeps the count linear: the original level-batched routine (V27LB) logged 34/66/98/130/162 operations at levels 1-5 (decision_log.md 2026-09-18T04:5xZ self-test, (1,1024,1024)×(3,1024,1024)). A.5 shows the shipped routine, whose op count likewise grows by a constant per level. Second, a wall-clock runner clause read under shared-box load measures the box, not the estimator. A graded incumbent failed the same clause in the same window, and a detached re-read at low load passed both (decision_log.md 2026-09-15T08:4xZ, 09:1xZ, 12:1xZ). Finding 1 is the origin of the op-count economy in §1; finding 2 is the origin of the load-gated wall reads in §2.3 C6.

§2.3 Contributions measured since 2026-10-01

Each item gives the change, its exactness evidence, the local paired ratio on the fixed 50-network gate, and the live result. Unless stated otherwise, the gate pairs a candidate against its parent on 50 local MLPs from id 500 onward (.claude/loop/fullgate.sh). It reports the paired score ratio and the largest per-network relative deviation of mse_final.

C1. Learned local-error corrector (Q-W194; idea Hisernberg V41 [12], our implementation). The chain leaves a local error at every layer, $\epsilon_l = \delta_l - \Phi_l(\delta_{l-1} W_l)$. A small per-neuron MLP learns it from 14 chain features in the pilot (the shipped k4b variant uses 17 plus previous-layer features): mean, variance, sd, pre-activation, α , φ , Φ , dG, raw and standardised k3, raw and standardised k4, post-activation mean and prediction. The predicted correction is transported as $c_l = \Phi_l(c_{l-1} W_l) + g_l$, and the output is $\text{pred}_{15} + c_{15}$ (plan :33726, :33731). We wrote the trainer from the published description (research-phase2/qw194/train_lt.py; tanh-GELU, because flopscope has tanh and no erf). Training used the organiser’s public full split, never mini. The weights ship as plain data (lt_weights.py, “data only, no code”). Every corrector operation is a metered fnp operation, and the transport runs in float64 (qw91_core.py:1108). Shipping trained weights as data and evaluating them with metered operations was checked against the rules before any send (plan :33728).

Held-out numbers. R = corrected / uncorrected final-layer MSE on MLPs that were not in the training set, i.e. leave-MLP-out by construction.

Corrector	Training MLPs	Held-out MLPs	Held-out R	Source
m250 base (14 features)	300	41	0.9309	research-phase2/qw194/mid250_base.j
m250 mp (+ previous-layer features through w)	300	41	0.9207	qw194/mid250_mp.json
four recipe variants (mnow sweep)	370	49	0.9184–0.9194	plan :34250
same recipe, twice the data	803	97	0.9292	plan :34325
+ hub feature (hubb)	—	—	0.8962	plan :34484
+ hub and k4 features (k4b, 17 inputs, shipped)	—	—	0.8915	plan :34625

Live. #334706 (m250 mp) against its uncorrected parent #334496: $4.516697\text{e-}09 / 4.856404\text{e-}09 = 0.9300$, against a held-out R of 0.9207. Across five corrector arms the host grade was $0.9887 \times$ the pre-registered formula point, spread ± 0.001 (plan :34300). In the host reports the final-layer MSE

moved from 2.167e-08 (#334496) to 2.006e-08 (#334706), and to 1.925e-08 with the k4b corrector (#334862). The corrector is an accuracy change and is not bit-exact by design. Doubling the training data did not help, and neither did more capacity or seed ensembles (§8 rows 21-23; plan :34484). It is the one MSE lever among the §2.3 contributions.

C2. A/P leg-pair merge (Q-W236; research-phase2/qw236/make_ap.py). Every pair of A and P leg products that shares one left operand becomes one level-batched Strassen call over a leading pair axis: the young legs $WDy @ A_st / P_st$, and the formed legs $Qc3 @ FAo/FPo$ and $QU3 @ FA2/FP2$. The A and P legs of each ping-pong side are views of one (2, L-1, n, n) buffer, so the call writes through out= with no copy (7 exact-match edits; make_ap.py:1-7). The per-element arithmetic is unchanged; the node bookkeeping is paid once instead of twice.

- *Exactness.* On MLP 500, 22,244 → 19,875 logged operations (-10.65%), output `np.array_equal` True, C/B 0.2210175 → 0.2208690 from the shared pre-additions; in-process peak RSS 5.78 → 6.74 GB (in-process `qw236/runone.py` harness, plan :34846). On the 50-network gate against the graded f1: paired ratio 0.999309, max rel dev mse_final 0 (job log arc-20261009T025023Z-1b4de0).
- *Live.* #334943 / #334881 = 4.160074985423857e-09 / 4.162953978152957e-09 = 0.999308. A local memory gate on the related `qw236/ship_apD` (A/P merge plus join knapsack) read 6054 MB (73.9% of 8 GB, plan :34913); the host reports carry no memory field.

C3. The 15-operation Winograd node sw15 (Q-W239; research-phase2/qw239/make_sw15.py). In flopscope, one broadcast elementwise call is one logged operation billed per output element, and views are free (`qw239/optest.py`, plan :34938). Two exact sign flips and a slot reorder let two pre-additions and two post-additions share calls. $S3' = A21 - A11 = -S3$ and $T3' = B12 - B22 = -T3$ give $M7 = S3' \cdot T3' = S3 \cdot T3$. $[T3', T1]$ is computed as $B12 - [B22, B11]$ in one call, and $[U3, U4] = U2 + [M7, M5]$ in one call. Each node then logs 15 operations instead of 17 at identical FLOPs (A.5, `qw91_core.py`:711-733).

- *Exactness.* `qw239/unit_sw15.py`, five shapes at levels 1-5, including batch broadcasts: output `np.array_equal` True 5/5, FLOPs identical 5/5, operations 24→22, 64→58, 84→76, 104→94, 104→94 (plan :34939). Re-run 2026-10-11 against the shipped `ship_c1sv5c5n5mf/qw91_core.py`: the same five lines. On the graded c1 core, MLP 500, in the in-process `qw236/runone.py` harness (which counts 27,238 for c1 where the gate census counts 26,993): A/P + sw15 take 27,238 → 23,249 operations, outputs `np.array_equal` True (plan :34950).

C4. Pair-batched range finder yq2 (research-phase2/qw239/make_yq2.py). The joiner range finder ran four level-3 Strassen calls per power-iteration pass: two for A, two for P. Stacking the A/P pair against the shared right operand 0m halves the node bookkeeping.

- *Exactness.* MLP 500: 23,249 → 22,180 operations, `np.array_equal` True against c1, C/B 0.2127788 (plan :34960). `qw239/unit_yq2.py`: `np.array_equal` True on four shapes, including the odd dimensions that take the `matmul` fallback (plan :34970). A fresh-context review re-derived every Winograd slot and passed both makers (plan :34972).
- *Live, C2-C4 together on c1 (#334950).* Paired gate ratio 0.999266, max rel dev mse_final 0 on 50 networks, gate maximum 22,227 operations (job log arc-20261009T034120Z-3ba6ec). Live 4.126764529678377e-09 / 4.129795578069123e-09 = 0.999266. Host residual median 0.375 → 0.322 s and maximum 0.390 → 0.339 s (#334932 → #334950, host reports), against a pre-registered median of 0.318 s [0.300, 0.335] (plan :34942).

C5. Re-spending the operations: Strassen level 5 at three sites (Q-W239 / Q-W257). The build script raises three knobs from 4 to 5 (research-phase2/qw239/chain_sw.sh:24-27): `v = _V27LB_LEVEL` (the young-leg transport and every `_sum_bmm_lb` site), `c = _QW75_LC` (the covariance sandwich), and `n = _LV_NB` (the newborn transport wt @ a_new). A deeper level is algebraically exact. It is not bit-identical in float32, so each arm carries a measured deviation.

Arm (on the C2-C4 base)	Census ops (MLP 500)	Local paired ratio vs c1	max rel dev mse_final (50 MLPs)	Live / #334932	Graded id
v:5	23,468	0.957866	0.00533	0.957927	#334966
v:5 c:5 (+ C6)	23,990	0.952834	0.0116	0.952420	#335245
v:5 c:5 n:5 (+ C6, paired)	24,260	0.949722	0.0069	0.949349	#335249

Sources: job logs arc-20261009T034121Z-b6c706 (v:5), arc-20261010T172635Z-899be5 (v:5 c:5), arc-20261010T172635Z-d895df (v:5 c:5 n:5), lines QW239 ... census and paired score ratio. The last step transferred one-for-one: live #335249 / #335245 = 0.996775, local 0.949722 / 0.952834 = 0.996734 (plan :35630). A fourth knob, _LV_QB at 5 on the A/P-stacked formed legs, failed the memory gate at 7578 MB (92.5% of 8 GB) and was not sent (plan :35025).

C6. A wall fail-safe that does not fire on typical networks (Q-W257; research-phase2/qw257/make_mf.py). The host stops a network at 120 s and scores it as a zero prediction. Before each layer, the core checks whether to finish the remaining layers with a Gaussian closure started from its own mean and covariance (qw91_core.py:357-363). The old trigger was elapsed + 1.3 × (last layer's wall) > 112 s. On #334991 (v:5 c:5) it tripped on 16 of 100 host networks. Nine of them were public, and their MSE was 8.9–12× worse, so the grade was 1.0071e-08 (×2.55 the incumbent) although the level itself transferred (cb_mean 0.2026 as predicted) (plan :35113–:35118). Per-network host telemetry showed the trips were false positives. For 14 of the 16 tripped networks, the incumbent's wall on the same network was 86.1–92.2 s, while untripped v:5 c:5 networks ran to 103.2 s. A single slow layer, or a host hiccup inside one layer, inflated the last-layer forecast (plan :35139). The local gates had disabled the fail-safe (ARC_WALL_LIMIT_S=1e6), so the 50-network gate could not see this (§6 finding 6).

The new trigger is elapsed × (1 + 1.3 / layers_done) > 116 s (qw91_core.py:362, :1347; A.6). A hiccup of h seconds now moves the forecast by h / layers_done, so the trigger binds only near the end: at layer 15 it fires when elapsed exceeds 106.7 s. Untripped, it overruns the 120 s stop only if a single layer after a passing check takes more than about 13 s, against host layer means of about 5.5–7 s (plan :35571). It reads the clock and adds no flopscope operation.

- *Value-neutral, locally.* With the trip disabled, each gate ratio equals its parent's to the 1e-6 void rule: 0.952834, 0.949722 and 0.957866 (plan :35606, :35607, :35622).
- *The trip test can fail.* Forced at ARC_WALL_LIMIT_S=40, MLP 500 falls back as designed. The tripped-to-untripped MSE ratio is 55.7, 70.4 and 55.7 for the three arms (job logs, line QW257_TRIPTTEST PASS).
- *Value-neutral, live.* 0 trips in 3 × 100 host networks (#335245, #335249, #335251). A trip would show as a bimodal FLOP count, and the per-network minimum/median of flops_used is 0.983 on each (plan :35614, :35629, :35636). #335251 reproduces #334966 to seven digits.

§2.4 Composite, and how well local predicted live

Every row below is a paired local gate followed by a host grade of the same artifact.

Step	Local paired ratio	Live ratio	Live – local
A/P merge on f1 (#334943 / #334881)	0.999309	0.999308	–0.000001
A/P + sw15 + yq2 on c1 (#334950 / #334932)	0.999266	0.999266	0.000000
v:5 on c1 (#334966 / #334932)	0.957866	0.957927	+0.000061

Step	Local paired ratio	Live ratio	Live – local
v:5 c:5 + C6 (#335245 / #334932)	0.952834	0.952420	-0.000414
v:5 c:5 n:5 + C6 (#335249 / #334932)	0.949722	0.949349	-0.000373

From c1 (#334932) to the paired artifact, cost work alone gave $\times 0.9493$, and the MSE stayed at $1.925e-08$. Counted in operations (gate census of MLP 500, `qw199/opsites.py`), the C2-C4 cut freed 4,813 ($26,993 \rightarrow 22,180$), and the three levels spent 2,080 of them ($\rightarrow 24,260$) to cut the billed multiplier from 0.2145 to 0.2036 (`host mean_score_multiplier`).

§3 The AUGMENT factorial null map — a negative result from parallel exploration

This section concerns a different artifact from #335249 and from the paired #337552. #335249 is built on team 504aldo’s disclosed method plus our cost engineering and a learned corrector (§2); nothing below explains any number in that chain. The work here used our earlier, independently built chain, last graded as #330715 ($2.0407694122741038e-08$). The result holds regardless: at 1024 wide $\times$ 16 deep, $B = 2^{41}$ FLOPs, score = final-layer MSE $\times$ max(0.1, C/B), the higher-cumulant corrections of AUGMENT mode in ARC’s reference buy no affordable accuracy.

Why AUGMENT. Our float32 chain matched ARC’s float64 SIMPLE reference to within 0.74% (MSE ratio 1.0074, 32 public Phase-2 networks); AUGMENT reached 0.4805. That 2.08x was the only accuracy headroom we could measure.

The factorial. ARC’s single augment flag switches three things, which we split by patching the reference: terms raises the diagram cost budget by one; R1 carries the incoming fourth cumulant as a full $n \times n$ core instead of a scalar; K211 enables the two contractions that couple the core’s off-diagonal entries to the rest of the closure. Controls: flags all off reproduced banked SIMPLE outputs to $\max|\Delta| = 1.02e-14$; all on reproduced unpatched AUGMENT bit for bit ($n = 64$, $L = 5$). Ratios are paired final-layer MSE against one $1e9$ -sample Monte Carlo truth.

(terms, R1, K211)	MSE / SIMPLE	networks	instrument	added C/B
(F,F,F) = SIMPLE	1.0000	—	ARC reference, float64	0
(T,F,F)	1.6043	16	our chain, float32	+6.6e-4, our chain
(F,T,F)	3.4742	16	ARC reference, float64	+0.109, analytic only
(F,T,T)	0.8935	16	ARC reference, float64	+5.5301, reference
(T,T,T) = AUGMENT	0.4805	32	ARC reference, float64	+10.7557, reference
(T,T,F)	not measured	—	—	—

The interaction is strict. No cheap cell beats SIMPLE, and two are far worse than changing nothing. Both cells that win contain K211, which is where the cost sits.

Mechanism. The reference truncates the fourth-cumulant correction in two places that must move together: which diagrams survive (each block carries a power of $n^{(-1/2)}$, and a diagram is kept only if its costs fit the budget) and how much fourth-cumulant state crosses each layer. SIMPLE and AUGMENT truncate both at matched order. Moving one alone yields an unbalanced truncation, effectively a different series. In (T,F,F), 18 added cost-3 terms act on the scalar state and anti-cancel the sum they were meant to complete. In (F,T,F), the core’s off-diagonal entries are computed and

transported, but under the averaged metric no budget-2 diagram reads them; only the cost-3 blocks the budget excludes would. Restoring K211 moves 3.4742 to 0.8935. We had registered a point of 0.85 for (F,T,F) (a 15% gain), band [0.60, 1.15]; the measured 3.4742 fell far outside it on the other side of 1. The unbalanced-truncation mechanism was predicted; the sign and size were not.

Price. We ran the reference under its per-block FLOP counter at $n \in \{96, 128, 192, 256\}$, $L = 16$, fitted each block's exponent log-log, and evaluated at $n = 1024$: SIMPLE 1.5645 C/B, AUGMENT 12.3203. Our chain computed the SIMPLE-equivalent 2.92x more cheaply; applying that discount to the add-ons puts (F,T,T) at 2.429 C/B and AUGMENT at 4.218. Stacking a further 3.71x discount, almost certainly double counting, still leaves 1.046 and 1.528, and the multiplier is uncapped above 1.

Our pre-measurement analytic count was optimistic by 1.64x for AUGMENT ($\geq +2.24$ against +3.682 measured, our scale) and 3.91x for (F,T,T) (+0.484 against +1.893). It priced the K211 contractions as $O(n^2R)$ with R fixed, but R , the factored rank of the third-cumulant tower, grows by $8n$ per layer; the blocks carrying most of the add-on fit exponents 2.97–3.00. A low-rank fourth-cumulant core would save at most 0.55 of the +10.76 reference C/B (a design estimate, not measured).

Caveats. Instruments are mixed: (T,F,F) divides our float32 result by the float64 reference; against our own chain's baseline it reads 1.5939, under 1% apart. That cell on the reference, and (T,T,F) on either instrument, were never measured.

Evidence: research-phase2/qw62/{oracle_r3a,ref_kprop_u1,check_r15,price_r3_flopcount}.jsonl, research-phase2/qw62/{oracle_r3a,price_r3_flopcount}.py; decision_log.md 2026-09-12T00:5xZ, 05:4xZ, 05:5xZ, 11:2xZ, 11:3xZ, 11:5xZ, 12:2xZ.

§4 The exact bivariate ReLU covariance is not the bottleneck at this shape — the cheap truncation is already at the exact answer

Credit first: the closure is not ours. venture agent ai published the exact nonzero-mean bivariate ReLU covariance in topic 18214 on 2026-09-05T17:37Z, before our Q-W51 ran, and stated the derivation route verbatim — “an exact closed form for the nonzero-mean bivariate ReLU covariance: Price’s theorem plus Plackett’s identity, reduced to a 1-D Gauss-Legendre integral, verified to 1.665e-16 against scipy and against 20M-sample Monte Carlo across 720 parameter combinations” — together with its payoff, 7.4% final-layer MSE against the starter kit, and the reason, “because Gaussian closure, not any one pairwise moment, dominates the remaining error” (sources/discourse-18214-20260908T232155Z.json, post 0). The closed form is older than the competition: Rosenbaum [10] gives the moments of the truncated bivariate normal at general means, and Bibi et al. [11] give the zero-mean form in a network moment-propagation setting. We rebuilt it from the spec and claim none of it.

What we add is a different comparison. 18214 priced the exact term against the starter kit; we priced it against the **third-order Price truncation already shipping in our own estimator**, which is the comparison that decides whether to carry the exact form at all. Against that baseline it buys 0.001% ($G = 1.000009653542642$), not 7.4% — which we attribute (not measured) to a small $|\rho|$ at this shape, where the truncation error is $O(\rho^4)$, so the series has already reached the exact answer to within the grader’s resolution. That is the transferable statement, and it is a stronger null than the kit comparison: when a truncation already matches the exact form on the realised input distribution, the remaining error is not inside that series, and no amount of exactness in that term will find it.

The implementation notes below are ours, and they are what made the measurement cheap enough to be worth making. Writing the closure as $\text{Cov}(\text{relu}_i, \text{relu}_j) = \sigma_{i\sigma_j}[\rho\phi_i\phi_j + \int_0^\pi \rho(\rho-u)\phi_2(a_i, a_j; u) du]$ and substituting $u = \sin t$ (Drezner-Wesolowsky) cancels ϕ_2 ’s $(1-u^2)^{-1/2}$ factor identically, so the integrand stays bounded by 1 even at $|\rho| = 1$, where the un-substituted form has an integrable singularity that plain quadrature resolves only slowly; rewriting the exponent as $(a - b \cdot \sin t)^2 / (2\cos^2 t) + b^2/2$ and pulling $\exp(-b^2/2)$ out of the node loop removes a float32 cancellation in the near-dead-unit regime. The result is $O(W^2)$ per node —

8-node Gauss-Legendre quadrature costs $\sim 1.3\text{e}8$ FLOP per layer against the layer's existing $\sim 8.7\text{e}9$, so building it exactly was never blocked on cost.

Two controls passed before any gate ran. C-EXACT (qw51_verify_exact.py, Monte Carlo $N=8\text{e}6$, 11 hard corners): the exact closure's relative error against MC is 0.01–2.37%, against 0.45–88.14% for a naive gain-based approximation; 8-node and 16-node quadrature agree to $1.427\text{e}-7$, below float32 resolution, so 8 nodes are exact for this purpose. C-IMPL (qw51_control_impl.py): the flopscope implementation matches an independently written float64 NumPy propagation to $9.263\text{e}-10$.

The gate is a measured null, not an untested hope. Paired against the third-order (Price-3) incumbent on the official 8-MLP arm (gate_qw51_8mlp.json vs #330391's local gate arm 2.5379025032634672e-06): final_layer_mse 2.5378780037499382e-06, $G = 1.000009653542642$ — the exact bivariate covariance buys **0.001%** over a third-order truncation, per-MLP all within $\pm 0.013\%$. A related basis-order probe, Q-W54 (a one-step re-derivation of 18216's own top rung), gave the same shape of answer from a different angle: local holdout predicted a confident 1.005406 win (CI [1.0037, 1.0071], 8/8 MLPs improved), the official arm returned $G = 0.9997174805127262$, a 0.03% loss, 3/8 improved — not submitted (decision_log.md 2026-09-09T11:5xZ).

Why the exact term is empty here though 18214 reports it worth 7.4% on its own, over the starter kit. The kit ships no covariance term at all; Price-3 (third-order truncation, already in the earlier chain by the time Q-W51 ran) already captures essentially all of that 7.4%, which we attribute (the inter-unit $|\rho|$ distribution was not measured) to a small $|\rho|$ at this shape, where a truncated series' error is $O(\rho^4)$ — third order is already accurate to the width the exact closure would add. This generalises past this one term: **when a series truncation already matches the exact form on the realised input distribution, the remaining gap is not inside that series**, and building the exact version confirms rather than closes it — two ticks were queued against this axis on the unexamined assumption that “exact” must beat “third order”; measured, it did not, by three orders of magnitude.

Evidence: research-phase2/calib/{cand_qw51_covprop_exact.py, gate_qw51_8mlp.json, gate_qw53_8mlp.json} · decision_log.md 2026-09-09T05:3xZ (Q-W51/COVPROP-EXACT) and 2026-09-09T11:5xZ (Q-W54).

§5 Constants versus structure

Alongside the structural port chain (§2.2) and §4's exact-covariance test, the campaign ran a second, independent axis toward parity: not adding structure but fitting more accurate **scalar constants** on top of the same marginal-Edgeworth backbone. That axis has a clean score history, a diagnostic that located its limit, and a qualitatively consistent negative result from an unrelated team using an unrelated method.

The ladder, as graded score (Phase-2 official grader; lower is better).

id	lever	graded score	Δ vs prior rung
#330386 Q-W45	Edgeworth mean correction, $K=2$	$2.6453\text{e}-07$	— (first rung)
#330388 Q-W47	basis order $K=6$	$2.3902\text{e}-07$	-9.6%
#330389 Q-W49	Price-3 truncation closure	$2.3012\text{e}-07$	-3.7%
#330391 Q-W50	calibration refit, same basis	$2.2983\text{e}-07$	-0.13% (inside the official arm's $\sim 0.3\%$ dead zone, §6 finding 3; one graded pair)
#330486 Q-W53	calibration-free envelope (the open-loop control of the D3 screen below)	$1.3201\text{e}-07$	-42.6%

id	lever	graded score	Δ vs prior rung
#330487 Q-W56	closed-loop feedback + joint-k3 constants	1.0593e-07	-19.8%
#330489 Q-W59	pooled refit on 24 corpus MLPs (POOL-24)	1.0556e-07	-0.35%
#330499 Q-W60	pooled refit on 72 seeds (POOL-72)	1.0495e-07	-0.58%

Eight rungs in one family: every rung adds or refits a **scalar constant multiplying a fixed functional form**, $\sigma * \alpha^m * \phi(\alpha)$, the Edgeworth one-step correction basis; the distinct mechanisms are the mean-correction term, the wider basis, the truncation order, the envelope and closed-loop feedback, and the other three rungs are refits. The ladder’s final table (Q-W60’s pooled refit, research-phase2/calib/feedback_coef_v7_pool72.json) carries **272 constants — 16 layers x 17 coefficients per layer**, confirmed by reading the file’s coef array directly (`len(coef) == 16, n_coef_per_layer == 17`); Q-W59’s table (feedback_coef_v6_pool24.json) has the same shape.

The D3 screen: the ladder’s own diagnostic. Before Q-W56 was built, an agent proposed its mechanism — accumulated mean drift of the propagated state, not a closure or fitting error — and reported four supporting numbers. Because the agent’s own output file was never written to disk — the numbers arrived as text — those four were independently re-implemented from the agent’s stated method (research-phase2/calib/qw56_d3_verify.py) and re-run before anything was adopted (decision_log.md 2026-09-09T17:5xZ):

variant	agent-reported	independent re-run	G vs Q-W53
open-loop control	1.297880e-06	1.297880e-06 (exact match)	1.0000
+ joint-k3 columns, row only	1.277496e-06	1.277496e-06	1.0160
stagewise feedback, no k3	1.222630e-06	1.222630e-06	1.0615
feedback + joint-k3 (= Q-W56)	1.066575e-06	1.066575e-06	1.2169

All four reproduced to the printed digit, including the read-order control (line 1 had to print Q-W53’s own holdout MSE to six digits before any other line could be trusted), so the mechanism was adopted and Q-W56 registered on that basis. **One further number from the same report did not go through this reproduction:** the same agent stated that the teacher-forced residual of the open-loop Q-W53 chain, regressed onto Q-W53’s own richest basis (ps1/K4), has $R^2 = 0.065$, i.e. the local injection is not a function of (α, σ) in that basis. qw56_d3_verify.py computes only the four MSE/G lines above; it does not compute an R^2 , and no separate script or log reproducing it survives on disk. $R^2 = 0.065$ is therefore carried at lower confidence than the table: reported by an agent whose other four numbers reproduced exactly, but not itself re-derived.

504aldo’s dead-end #10 is qualitatively consistent. Their public disclosure (AICrowd discourse 18218, code in scout/whest-p2-cumulant-k3/) reports among their 25 measured dead ends that live final-layer features correlate < 0.09 with the true error and that a correction on them is worth 1.01x on their regeneration base. Their estimator (factored $K=3$ cumulant propagation carrying an explicit cross-neuron third-cumulant slice, D21) and their diagnostic are unrelated to ours. The two numbers measure different quantities on different populations and are not commensurable; what they share is the qualitative finding that neither team found a cheap function of the per-neuron state that explains the remaining error.

Why we read it as a limit of the family. 504aldo name a missing object directly: the cross-neuron third-cumulant slice `D21_ic = kappa3_iic`, an $n \times n$ matrix their architecture carries structurally

and that no scalar multiplying $\sigma * \alpha^m * \phi(\alpha)$ can express at any coefficient, because nothing in that basis is a function of a matrix indexed by two distinct neurons. That is why the campaign retired the calibration ladder after Q-W60 and moved its structural effort onto reproducing D21 itself (the port chain of §2.2), rather than re-fitting a 272-constant table.

Evidence: `decision_log.md` 2026-09-09T05:3xZ (Q-W45/47/49/50 grades), 2026-09-10T00:13Z (Q-W53 and Q-W56 grades #330486, #330487), 2026-09-10T00:45Z (Q-W59 grade #330489), `ledger` 2026-09-10T11:21:08Z (Q-W60 grade #330499), 2026-09-09T17:2xZ-18:0xZ (D3 screen, agent report and independent verification), 2026-09-10T23:5xZ (Q-W62 pre-registration citing the $R^2 = 0.065 / \text{dead-end-}\#10$ agreement as the reason to close the axis) · `scout/FINDINGS.md`, `scout/whest-p2-cumulant-k3/docs/community_post.md` (504aldo's dead ends, #10) · `research-phase2/calib/{feedback_coef_v6_pool24.json,feedback_coef_v7_pool72.json,qw56_d3_verify.py,gate_qw5`

§6 The measurement instrument, and eight things it taught us about itself

The campaign's baseline discipline did not change: a band is pre-registered before any number is seen, the graded read is checked against it, and both are kept regardless of outcome. Through #331541 the graded reads formed an unbroken run of fifteen in-band draws (2026-10-01 text, §2). After that the record was no longer unbroken. Two grades landed below their bands, in our favour (#334837, #334844; plan :34505, :34511). Among the failures, two (#334927, #334991) became findings 5 and 6 below. Findings 1-4 are carried over from the earlier text.

1. A submission that never reaches the ledger cannot be reported as missing. #330715 (Q-W62 R1.6) was sent with a direct `whest submit` call instead of the standing shipper script, `arc_ship_slot4.sh` — the only path on the box that appends to the append-only submitted-ids `ledger preflight.sh` polls against. The id was created at 2026-09-12T00:56:39Z (the platform's `created_at`, not the time of grading) and graded inside its registered band, but three consecutive submission polls (01:08Z, 01:14Z, 01:19Z) each printed the prior incumbent as unchanged, because the poll never queried an id it did not know existed — a defect that cannot show up as a PENDING row or a stale-read warning, since nothing about it looks abnormal from inside the instrument. The ledger had already been hardened once before, after a 2026-09-10 incident where a hardcoded id tuple went stale; that fix addressed the *reader* and left the *writer* as a side effect of one optional code path, which is the general shape of the bug: **a single-sourced ledger is a hardcoded tuple with extra steps**. Fix shipped immediately: `preflight.sh` now reconciles the ledger against every `arc_ship_*.id` file in `scratch` — which every send path writes, regardless of which script sends — before each poll, appending anything present there and absent from the ledger. Verified idempotent (a second run appends nothing) and verified to close the gap (the poll's next read reported the correct incumbent). Evidence: `decision_log.md` 2026-09-12T05:2xZ.

2. One measured deflator on a local gain. On the Q-W62 K=3 step R1.5 → R1.6 (#330677 → #330715), the local adjusted ratio 0.96575 (a 3.425% predicted gain) graded 0.97239 (a 2.761% realized gain), a realized-to-local *gain* ratio of 0.806. Two earlier draws of the same family give only level factors (graded/gate 0.9786 for R1 #330643 and 0.9706 for R1.5 #330677), and the log itself calls them not a calibration. The campaign afterwards registered point predictions from paired local ratios pre-deflated, with the deflation stated. Evidence: `decision_log.md` 2026-09-12T05:2xZ.

3. Below $|G-1| \approx 0.3\%$, the 8-MLP local arm can return the wrong *sign*, not just the wrong *magnitude*. Q-W50 (Q-W49 with 128 Edgeworth constants refitted, nothing else changed) graded a **0.13% win** ($G = 1.001266$) against Q-W49; the same pair, read on the local 8-MLP official-runner arm, gave a **0.14% loss** ($G = 0.998587$). Registered as closed axis **§3.48**: a local paired G on this arm reproduces a *large* graded G to within $\sim 0.1\%$ (Q-W45: local 1.4853 vs graded 1.48678), but the arm cannot resolve, and can flip the sign of, an effect within $\sim 0.3\%$ of unity. Every gate registered after this finding states explicitly whether its expected effect sits inside or outside that dead zone, and a near-null effect is recorded as *unadjudicable on this arm* rather than shipped or killed on its say-so. Evidence: plan §3.48; the Q-W49/Q-W50 pair.

4. A paired bootstrap half-width is a property of the specific pair measured, not of n or of the suite, and does not transfer across pairs. A 2026-09-12 closure argued a q=2 stabilized-subspace-iteration lever was undecidable because its predicted effect ($\sim 0.28\%$) sat inside a 0.316%

half-width measured on a *different*, looser pair ($q=0$ vs $q=1$) at the same $n=500$. Re-opened and actually measured on the real pair ($q=1$ vs $q=2$): half-width **0.0285%, 11.1x tighter**, because the two arms' per-MLP outputs were far more correlated ($\text{corr} = 0.99967$ vs the looser pair's lower figure) and the per-MLP log-ratio spread was narrower ($\text{sd } 0.003250$ vs 0.0372). The re-measurement did not rescue the lever — $q=2$ measured a 0.37% loss on the $n=500$ local gate (never submitted), CI $[1.003427, 1.003996]$ entirely on the wrong side — but it proved the *reason* given for the original closure was false: resolution is set by how tightly two specific arms co-vary, which a looser pair systematically understates, not by a fixed floor that a fresh $n=500$ draw inherits automatically. The standing rule: a half-width is never carried from one pair to another; if a lever needs finer resolution than a prior half-width offers, that resolution is *measured on the actual pair*, not assumed unreachable. Evidence: `decision_log.md` 2026-09-12T05:4xZ (Q-W62 R1.7 gate read).

5. The residual cap is an op-count cap, and the first diagnosis blamed memory. #334927 was the A/P merge plus a join-Strassen knapsack at about 27,000 operations (26,993). It graded 0.015871631357144846 against a band of $[4.075\text{e-}09, 4.124\text{e-}09]$ (ledger 2026-10-09T02:32:54Z). The first diagnosis, made without the host report, attributed it to +817 MB of peak memory, and a decision note banned the merge (plan :34903–:34904). The host's per-network telemetry said otherwise. Two networks hit `residual_wall_time_exhausted`: public MLP 0 at 0.4113 s, whose zero fallback (MSE 0.7936) alone gives $0.7936 / 50 = 0.01587$, and private MLP 90 at 0.4082 s. The memory gate on the same artifact read 73.9% of the cap (plan :34911, :34913). The same read also exposed a latent loss: the then-incumbent #334881 had lost private MLP 91 to the cap at 0.4007 s, which the public score cannot show (plan :34911). In the host-report table sorted by residual median (column `res_p50` of `research-phase2/qw211/host_table.tsv`), the first completed report that lost a network is #333653, at a median of 0.357 s ($p90$ 0.397 s; 9 networks lost); no completed report with a lower median lost one, and the next loss is #334716 at 0.370 s (6 lost). The 0.355 s gate described next therefore sits only 0.002 s under the first loss. (Plan :34912 quoted the first loss at 0.365 s, which is that report's `res_mean`, not its median.) On our own 20 uncensored host reports, the excess of maximum over median residual has $p50$ 0.016 s, $p90$ 0.038 s and maximum 0.055 s (plan :34995). The residual gate was recalibrated to 14.0 μs per operation (pass at a predicted median of 0.355 s or less), dry-run against three real artifacts, and shown to refuse them (plan :34914). Those three were read before the rate was set, so that check is in-sample; out of sample the law under-predicts by about 8% on the paired #335249 (0.340 s predicted, 0.369 s read, §1). The ban on the merge was lifted within the hour. Lesson: read the host's per-network telemetry before diagnosing a dead grade. A decision written on a hypothesis is a rotted constraint the moment it is recorded.

6. A gate that cannot fail, in both directions. The core's wall fail-safe tripped inside the local 50-network gate on a loaded box (walls 97.3–109.8 s at load 18–21, MSE $\times 9$ –31 on all 50) and refused a sound arm (plan :35023). The fix disabled the fail-safe in the value gates (`ARC_WALL_LIMIT_S=1e6`). After that fix the code path that later failed #334991 on the host was never exercised locally (plan :35118). The same gate's MSE check covered 5 of 50 networks against a $10\times$ threshold, so it could not fail at $\times 2.6$ (plan :35114–:35116). The repair keeps the value gates trip-free and adds a separate trip test that forces the trip at 40 s and must observe the fallback (§2.3 C6). Each gate must be shown to fire on a real artifact before it is trusted.

7. Within one family, local paired ratios now transfer at slope one. §2.4 lists five paired steps whose live ratios match the local 50-network ratios to within 0.0005, two of them to six digits. The last three pre-registered points missed by -0.04% , -0.04% and $+0.001\%$ (plan :35614, :35629, :35636). Across five corrector arms, the host grade was $0.9887 \times$ the registered formula point, spread ± 0.001 (plan :34300). This is a check of the instrument, not a discovery: cost-only steps change a deterministic FLOP count, and they transfer as they should. It contrasts with finding 2, the gain deflator of 0.806 measured on one accuracy step of the earlier chain.

8. The host reports expose every graded submission's telemetry. Each AICrowd submission page carries per-network `flops_used`, residual time and predict wall, including for other teams' graded ids. This turned the 2026-10-01 text's estimated rival multipliers into reads (§7) and made per-network error correlation with the leaders measurable. The same reports showed that our board row and our paired id can differ (the board row displays #335245 at the rounded $3.9\text{e-}09$), so

the paired id is named explicitly.

§7 The frontier, decomposed

The primary score is $s = \text{mean_m}[\text{MSE_m} \times \max(0.1, C_m/B)]$. The grader also returns the mean final-layer MSE before the multiplier. We decompose any gap as (ratio of final-layer MSE) $\times$ (ratio of multipliers). Unlike the 2026-10-01 text, the multipliers below are not estimates: they are the host's own `mean_score_multiplier` (the public-network mean of C_m/B) read from each submission's host report (§6 finding 8). For V29, for which no host report is archived, the multiplier is score / MSE from the grade poll.

Comparison	Their score	Gap (ours / theirs)	Final-layer MSE ratio	Multiplier ratio	Their multiplier	Source
Board leader J2W #335031	1.443e-09	$\times 2.717$	$\times 1.421$	$\times 1.912$	0.1065	sources/host-report-335031.json
504aldo V29 #330093	5.4181681379380312369	$\times 3.7236$	$\times 0.9026$	$\times 0.8015$	0.2541	sources/grade-poll-refs-20260922T041410Z.json

Ours throughout is #335249: 3.920616e-09, MSE 1.925e-08, multiplier 0.20363793016673298.

Against the leader, 64.8% of the log-gap is the multiplier ($\ln 1.912 / \ln 2.717$). The leader bills at the floor's edge (0.1065), and its final-layer MSE is 29.6% lower than ours (1.355e-08 against 1.925e-08). Three host-report reads from 2026-10-10 sharpen this.

1. *The difference is uniform across networks.* On the 50 shared public networks, ours / J2W final-layer MSE has median 1.418 (p10 1.263, p90 1.591 as order statistics; 1.268 and 1.592 with linear interpolation), and J2W beats us on every one of the 50 networks (computed 2026-10-11 from `public_scores` in `sources/host-report-335249.json` and `-335031.json`).
2. *It is the same difficulty ordering.* The Spearman correlation of per-network final-layer MSE between ours and each of the 16 top rows is 0.59–0.77 (lowest: `m1iston #335147`, 0.591). Between J2W and the next two rows it is 0.757 and 0.763 (plan :35670). There is no sign of a different estimator family at the top.
3. *They carry better content at every depth.* The ratio of all-layer to final-layer MSE is 0.41–0.43 for the top four rows against 0.46 for ours (plan :35644).

Against V29, the disclosed endpoint of the method we started from, we now lead on both axes: final-layer MSE $\times 0.9026$ (the corrector, \$2.3 C1) and multiplier $\times 0.8015$ (cost engineering, \$2.1–\$2.3). The 2026-10-01 text recorded the opposite sign on cost (0.3135 against 0.2541).

Required delta, not a forecast. To the prize line (rank 3, 1.6e-09 on the 2026-10-10T21:02Z board) we need $\times 0.408$. The cost axis can give at most $\times 0.491$, because it is floored at 0.1. Even at the floor, MSE must therefore improve by at least $\times 0.83$ (plan :35644). Truth-fed oracles on our own chain bound what an accuracy change could buy at most: true per-neuron sd at every layer $\times 0.571 / \times 0.633$ (MLPs 508 / 512, plan :35215); true per-neuron k3 and k4 jointly $\times 0.680 / \times 0.639$ (plan :35220). These are ceilings, not estimators. §8 rows 19–58 record what was built against them and did not reach them.

§8 Negative-results ledger

Rows 1–18 are the closed axes recorded before 2026-10-01 (plan \$3.48–\$3.69), transcribed from the earlier text. Rows 19–58 were added between 2026-10-08 and 2026-10-11. Each row gives one number and its plan line; the full pre-registration, the controls and the raw command are at that line. A KILL is by the row's own pre-registered rule unless the row says otherwise. Together they are a measured map of what does not move the score at 1024×16 for this chain family. The 2026-10-11 additions name the mechanism that blocks each one.

#	What was tried / assumed	Measured number	Why it failed
1	Adjudicate a sub-0.3% candidate on the official 8-MLP arm	arm resolution floor $\approx 0.3\%$	the arm cannot resolve an effect this small; Q-W49/Q-W50 failed sign (§3.48)
2	Low-rank residual for the K=3 birth-X leg (P d(k3) + 3A d(e) split)	final-MSE ratio vs exact: 8.64(r=0) $\rightarrow$ 1.0073(r=256)	the squared-covariance and old (2,1)-feedback terms are full-rank, not low-rank (§3.49)
3	Price AUGMENT's k=3 closure as one axis	affordable region closed cell-by-cell across terms $\times R1 \times K211$	it gates three independent things (diagram budget, κ_4 core, off-diagonal contractions), each with its own price (§3.50)
4	Assume unused FLOP budget is free compute	score multiplier = $\max(0.1, \text{utilisation})$	a bit-identical estimator run at two utilisations confirms the floor is a hard function of C/B, not slack (§3.53)
5	Price Q-W56's extra cost analytically	priced 3.24e10 FLOP (40% of allowance); measured 6.48e10 (80%)	op-counting FLOP cost is unreliable; only a measured 1-MLP smoke gives the true price (§3.55)
6	Treat one calibrated local-vs-graded G as unbiased	arm over-predicted +4.12% and +1.91% on two fresh draws	one calibration draw was generalised into an unbiasedness claim; G is an upper bound and must be deflated before projecting (§3.56)
7	Correct ratio bias with a flat constant DEFLATE = 1.03	flat constant does not map 1 $\rightarrow$ 1 at the no-op point	a bias-correction family must satisfy that boundary condition; the flat form was wrong in form though not badly in fit (§3.60)
8	Trust a repaired append-only submission-id ledger	arc_ship_slot4.sh wrote #330499 outside the reader's ledger	the <i>reader</i> was repaired but the <i>writer</i> was not — a single-sourced ledger with one broken write path silently drops entries (§3.61)

#	What was tried / assumed	Measured number	Why it failed
9	Fit a linear deflation (KEEP=0.911708) on draws 1-2 and predict draws 3-4	predicted via KEEP 0.911708 (fit on draws 1-2); Q-W60 measured keep 0.921907, score point error 0.0064% against 2.950% for the flat 1.03; draw 3 (keep 0.637) was a noise excursion	validated out-of-sample, including in the small-gain regime the registration itself called weakest (§3.62 — the one row here that is a held prediction, not a failure)
10	Assume float32 casts inside the calibration pipeline are harmless	$R = \text{MSE_f32}/\text{MSE_f64} = 1.000000097, 1.0\text{e-}7$ relative	measured, not assumed — 30,000x inside its own registered band; closes the channel rather than leaving it open (§3.63)
11	Raise truth-MC samples per corpus MLP to cut noise	noise-induced coefficient error $\approx 4\text{e-}11, \leq 0.06\%$ compounded over 16 layers	Q-W59's real effect was MLP-to-MLP <i>structure</i> , not sampling noise — more samples buys nothing on a dead axis (§3.64)
12	Re-read the leaderboard to find who is above us	a 19.4x forum-disclosed lever sat unread 14h during board-only re-reads	a board pull answers “who is above us”; only the forum answers “and is it reproducible”, which is what the parity gate turns on (§3.65)
13	Apply Phase-1's 1e11-FLOP/s residual billing rate under Phase 2	Phase 2: $\text{lambda_flops_per_second}$ residual is a hard 0.4s gate, not a billed rate	whestbench 0.16's ~~PHASE2_ROUND~~ config is structurally different from the Phase-1 model in 0.14 (§3.66)
14	Trust a paired gate whose two legs were fitted on different sample counts	reported +0.39%, matching genuinely-graded +0.35%/+0.58% for that exact variable	pairing the <i>evaluation</i> seeds is not pairing the <i>fit</i> set size; caught only because a second, independent bug froze the optimiser (§3.67)
15	Gate an exact layer against a noise-floor threshold derived from another layer	layer-0 floor $6.8\text{e-}10$ vs a final-layer-derived threshold of $3\text{e-}10$; the exact estimator reads $6.4\text{--}7.6\text{e-}10$ and fails	final-layer avg_variance (0.0755) does not transfer to layer-0 (0.68) (§3.68)
16	Assume a cached dataset snapshot hash is the tag it is named after	refs/v1-phase1 → 7c3aa28b (256×32 nets); refs/v2-phase2 → aa99830f (1024×16)	a snapshot hash must be resolved via refs/<tag> and its weight shape asserted before any gate reads it (§3.69)

#	What was tried / assumed	Measured number	Why it failed
17	Trust a shipper script to submit cleanly on the first try	shipper failed twice; recovery run printed qw53 already submitted as 330486 – skipping	the append-only submitted-id sentinel caught both failures before either could double-spend a slot (\$3.57)
18	Re-derive the billing formula after a whestbench version bump	formula reconstructed from 48 per_mlp rows across 6 arm runs under whestbench 0.14.0	it holds for 0.14.0 only; under 0.16 (Phase 2) residual is not billed, so it was superseded (row 13; \$3.54, \$3.66)
19	κ_4 -slice transport into the chain, five injection modes (Q-W207)	best mode R 0.989 / 0.986 (MLPs 508 / 512; kill at 0.97)	no mode clears the bar; four of five make the output worse (plan :34088)
20	In-chain learned k3 correction at layers ≥ 9 (Q-W209)	offline held-out R^2 0.73, end-to-end R 1.41 / 1.21	in-chain feedback amplifies a per-layer offline gain instead of damping it (plan :34100)
21	Tune the corrector recipe (4 configurations)	held-out R 0.9184–0.9194	recipe axis flat to 0.001 (plan :34250)
22	Double the corrector’s training set (370 → 803 MLPs)	held-out R 0.9186 → 0.9292	not data-limited at this recipe (plan :34176, :34325)
23	Seed ensemble of the corrector	$\Delta R < 0.0003$	seed axis flat (plan :34301)
24	Sample only the uncertain neurons (hybrid sampling)	uncertain fraction 0.602 (layer 9) ... 0.454 (layer 15), kill at 0.40	too many neurons are uncertain to sample cheaply (plan :34216)
25	Read κ_4 off the existing k3 legs (Q-W215)	correlation with truth -0.163 / 0.022 at layer 14, kill below 0.6	the leg-derived κ_4 does not track truth (plan :34271)
26	Drop early-born sources (birth-layer gate, Q-W216)	$s_0 = 4$: $F \times 0.605$ at $MSE \times 95.2$ / $\times 59.7$	early sources carry 72% of F and cannot be dropped (plan :34272–:34273)
27	Output-side adjoint basis for the old tier (Q-W217)	$F \times 0.685$ at $MSE \times 27.5$ / $\times 20.7$	a legacy cut of the same size costs only $\times 3$; the adjoint basis loses content (plan :34326)
28	Cheap knee core + its own corrector (Q-W210-B)	score ratio 1.0063, kill at 1.00	the corrector does not recover what the cheaper core loses (plan :34382)
29	Even-class κ_4 birth formula (Q-W220-K)	$R^2 -3.06$ at layer 15 (kill below 0.5)	the even-class birth misses 76% of the unit-check target (plan :34392)

#	What was tried / assumed	Measured number	Why it failed
30	Drop the odd (1,3)/(3,1) slices (Q-W220-P)	MSE $\times 1.240$ / $\times 1.136$ for F $\times 0.9995$	the odd slice carries 12-24% of the final MSE (plan :34397)
31	Gaussian-copula (Nataf) closure with Cornish-Fisher marginals (Q-W222-N)	systematic error cut 0.006 (kill below 0.15; pass at ≥ 0.40)	the systematic variance error barely moves (plan :34420)
32	Learned birth-term corrector (Q-W222-R3)	held-out $R^2 \leq 0.0285$ over layers 9-14	birth error is not predictable from per-neuron state (plan :34445)
33	Hybrid closure with Monte Carlo cumulants (Q-W223-M)	injected error $\times 260$ the chain's at N = 8192	the chain is far more precise per neuron than any affordable sample (plan :34442)
34	Adjoint-anchored exact k3 (Q-W224-A)	injected error $\times 19$ the chain's	its leading-order source model is cruder than the chain's births (plan :34466)
35	Fused rewrite of the k=2 power-cumulant table (Q-W235-K)	max deviation 0.0371, exactness bar $1e-5$	not an exact rewrite (plan :34830)
36	Sketched (2,1)-slice carrier (Q-W240-R)	exact top-64 SVD of t21: MSE $\times 9.3$ / $\times 8.06$	t21 is not low-rank (plan :35072)
37	Matrix-free low-rank young t21 (Q-W244)	ratio 5.41 / 4.80, kill above 1.15	t21 $\rightarrow$ covariance needs the full-rank young slice (plan :35110)
38	Hadamard-square even-state transport (Q-W245-A)	worst ratio 1.7465, pass at 0.65	does not pay for its FLOPs (plan :35194)
39	The same without feed or feedback legs (Q-W245-C)	1.868 / 1.656	the thin legs are not redundant (plan :35220)
40	Short-lived K3 sources, composed birth window $\times$ lifetime cap (Q-W248)	MSE $\times 2.4$ (single-factor age10, MLP 512) to $\times 146$ (composed w5a4, MLP 508) over 13 arms; composed arms $\times 49 - \times 146$	every K3 deletion costs at least $\times 2.4$ MSE (plan :35233, :35246)
41	Learned per-neuron sd head (Q-W249-SDH)	pooled held-out R^2 0.0123, kill below 0.10	the sd error is not predictable from chain state (plan :35248)
42	Per-layer scalar gain on the chain's k3	best non-control R 1.0100 / 1.0085 (508 / 512); minimum 1.00 is the control	the low-dimensional part of the k3 error is nil (plan :35252)
43	Strassen level 5 on the stacked formed legs (q:5)	memory peak 92.5% of 8 GB	memory, not operations, binds there (plan :35025)

#	What was tried / assumed	Measured number	Why it failed
44	k ₄ from covariance-response modes (Team Puffi Phase-1 “SSC”, brief I1/I7)	best arm 0.9959 on MLP 508, kill above 0.97	held-out k ₄ rms improved ×0.84, below the chain’s final-MSE resolution (plan :35438)
45	Per-layer scalar on the chain’s k ₄ (k4L)	×1.0635 / ×1.0412	moving the k ₄ level toward truth does not move the output (plan :35328)
46	Fit the k ₄ off-diagonal slices to the next layer’s variance error (VT)	best 1.0296 / 1.0234	better local fit, no end-to-end gain (plan :35453)
47	Bill the main products through symmetric or aliased-operand pricing (Q-W252)	best ×1.0404 (worse); whole chain at best about ×0.99	no usable billing discount at bit-equal values (plan :35382)
48	Radial / χ factorisation of the shared input scale (Q-W253)	G = 0.005, kill below 0.10	no measurable shared-scale error to correct (plan :35400)
49	Pilot-MC per-MLP output calibration (Q-W254)	oracle affine gain mean 0.005	almost none of the per-MLP error is systematic (plan :35425)
50	Hypothesis: the leaders sample with variance reduction	plain sampling at ~7k points gives MSE ~2e-05; parity needs ~1400× variance reduction	the leaders are analytic propagators (plan :35440)
51	Multilevel pilot-sd injection at a mid layer	pilot log-sd standard error 6.1e-3 against a chain error of 4e-4 (×15)	sampling cannot beat the chain per neuron in this regime (plan :35455)
52	Non-local features for the corrector target (brief I4)	cross-validated R ² -0.0010...0.0010 on 38 MLPs	no signal beyond the per-neuron corrector (plan :35478)
53	Collective-mode (few-eigenmode) covariance closure (Q-W255)	best held-out G 0.011, kill below 0.40	the variance error is a projection of a high-rank covariance error (plan :35481)
54	Ranked CP carrier for newborn k ₃ slices (brief I9)	MSE ×15.65 / ×9.15	probably a mapping defect; not repaired, so a weak negative (plan :35517)
55	One-step teacher-forced drift corrector (brief I6)	mean ratio 1.0000	the target carries nothing the final corrector lacks (plan :35518)

#	What was tried / assumed	Measured number	Why it failed
56	Function-preserving positive gauge (variance-equalised coordinates, CX-GAUGE CAN)	MSE $\times 18,372 / \times 1,477$	the chain is not gauge-equivariant: its absolute floors and truncation thresholds are tuned to the native scale (plan :35560--:35561)
57	Moment-cone projection of the joint (k3, k4) state (CX-MOMCONE)	consumed block min eigenvalue $\geq -5.34e-06$ (already feasible)	the violations sit in the unconsumed dead tail; decided KILL on substance, against the letter of a mis-specified pre-registration (plan :35593--:35594)
58	Eliminate the old tier's dense legs by exact factor-space algebra (brief I3, 2026-10-11)	factor-space Hadamard cost $\sim n^3 r^3 = 5.8e10$ per source against dense formation $n^2 r = 4.0e8$ (n 1024, r_old 384; the nested tier has $r_2 = 224$, r_{OLD2} at qw91_core.py:522)	the dense legs feed elementwise products ($A \circ P$, $P \circ P$, $A \circ A$) whose factor-space form is a rank- r^2 Khatri-Rao object (ship_cls5c5n5mf/qw91_core.py:976; plan :35683)

Pattern across rows 1-18. Every row above was caught by a control or an out-of-sample check built for a *different* reason, not by suspicion of that specific result. A measurement discipline pays off on failures it was not designed to catch.

Pattern across rows 19-58. Three statements cover most of them. (a) In this chain, a better local fit of an intermediate cumulant does not reach the final MSE. Rows 20 and 44-46 each improved their intermediate target and left the output unchanged or worse; rows 52 and 55 found no signal beyond the output corrector. (b) The cheap-carrier family is closed by the spectrum. The objects the chain carries (t21, the young slice, the variance error) are high-rank, so every truncated or low-rank substitute costs $\times 2.4$ or more in MSE (rows 26-27, 36-37, 40), and the few-mode search finds no structure (row 53). (c) Sampling cannot help this estimator at this budget, because the chain is about 15 times more precise per neuron than any affordable sample (rows 24, 33, 49-51). The remaining gap therefore needs new joint (cross-neuron) fourth-order content carried at a cost below the current legs. None of the 40 rows found it.

§9 Reproducibility of the earlier lineage

Environments. The host graded #335249 under whestbench 0.16.1, flopscope 0.12.1 and Python 3.10.21 (host report runtime_environment). Local gates ran in the frozen conda env compl0p2, whose manifest (manifests/versions-compl0p2.txt, frozen 2026-09-01) pins flopscope==0.12.0, whestbench==0.16.0 and Python 3.11.15. Read on 2026-10-11, the installed package versions match the manifest, but the interpreter reports Python 3.11.16. The local→live agreement in §2.4 is the measured consequence of these version differences for this family.

Code of record. The code behind every graded number is the package on the grader's servers, addressed by its submission id. The earlier record #335249 is the folder research-phase2/qw239/ship_cls5c5n5mf/. Hashes were recomputed from disk on 2026-10-11. The qw91_core.py prefix matches the one recorded when the candidate was registered (de872ff05d17, .claude/loop/state/candidates/arc.tsv row c10101903573bd8).

File	sha256
estimator.py	477f75e90d06f1acccd08cc0d10cf9fc11f320378e13cd1b20e55
qw91_core.py (scored core)	de872ff05d17fd01a51da989f82e9f42ebcdba22eb9809d87b73d
qw73_core.py (off-suite shapes)	23ffc62e88f9f9a1305400c6929cf5a8100085f3d90fea1f58523
lt_weights.py (corrector weights, data only)	070b7d877eb87cffb03f00a57a50ca43e81f729a41019d8781d78

The replicate folders share estimator.py, qw73_core.py and lt_weights.py byte for byte. Their cores are qw239/ship_c1sv5c5mf/qw91_core.py (#335245, 16036f3d9b02ce61...) and qw239/ship_c1sv5mf/qw91_core.py (#335251, 31546415f2717a54...). diff against the paired core shows only _LV_NB = 4 for #335245. The fail-safe lines :362 and :1347 are the only difference between ship_c1sv5 (#334966) and ship_c1sv5mf.

Regeneration. The paired folder is built from the graded c1 folder by research-phase2/qw239/chain_sw.sh (A/P merge → sw15 → yq2 → level edits, :22-27; the A/P, sw15, yq2 and fail-safe patches each refuse unless their anchor matches exactly once, and the level edits refuse if the new assignment is absent), followed by the fail-safe patch research-phase2/qw257/make_mf.py. The chain that produced it was LVS="v:5 c:5 n:5" MODE=ratio RMAX=0.9500 on research-phase2/qw257/chain_mf.sh c1sv5c5n5mf 24600 (job log arc-20261010T172635Z-d895df, line 2). Every arm then passed, in order: a census of logged operations on MLP 500 against the op budget; a wall gate; a residual gate at 14.0 μs per operation; a memory gate at 85% of 8 GB on 6 MLPs; the 50-network paired gate; the official scorer’s smoke run; and the trip test (same log, lines 7–15). The unit tests qw239/unit_sw15.py and qw239/unit_yq2.py run on CPU without the dataset.

Dataset. The local gates read public MLPs from the organiser’s v2-phase2 revision (challenge page :159); the corrector was trained on the public full split. The grader’s held-out suite is not available to us, and no number here is derived from it except through the grader’s own report.

Availability. Our repository is private while the competition runs. The cited paths (candidate folders, gate outputs, host-report captures under sources/, the plan and the decision log) will be released under an OSI-approved licence with the solution code if this entry is prize-eligible; rules §8 requires the release of the solution code. Until then any cited file is available to the organisers on request.

§10 LLM-use disclosure

As the prize guidelines require, we disclose that this entry was developed end to end with Anthropic’s Claude models (Fable, Opus and Sonnet families), operating through Claude Code under human direction. OpenAI Codex proposed two of the premises tested in §8 (rows 56–57). The models’ work included the ports of 504aldo’s specification, the diagnoses in §6, the corrector training pipeline, every candidate build and gate script, and this write-up. The same applies to Part I, whose build and gate scripts (research-phase2/qw5xx/) were written by these models; OpenAI Codex also built some candidate scripts in the automated loop (ledger CODEX-DEV rows). The quantitative claims cite a grader read (ledger or host report under sources/), a job log, or a plan line recording a measured run; Appendix S lists those sources, and the 2026-10-28 audit re-checked them claim by claim. Model-proposed hypotheses and fixes had to survive pre-registered measurement before they became claims. §6 finding 5 is a case where a model’s first diagnosis was wrong and the host telemetry caught it; finding 6 is a gate fix that left the failing code path unexercised until the host failed #334991.

References

- [1] Alcrowd, “ARC White-Box Estimation Challenge 2026” — task, scoring, prizes. <https://www.aicrowd.com/challenges/arc-white-box-estimation-challenge-2026>.
- [2] Alcrowd Discourse: “Phase 2 of the ARC White-Box Estimation Challenge is live” (<https://discourse.aicrowd.com/t/phase-2-of-the-arc-white-box-estimation-challenge-is-live/18197>, accessed 2026-08-24, archived sources/discourse-18197-20260824T050732Z.json); mohanty, “Algorithmic contribution prize guidelines: How ARC

judges these prizes — discretion, technical writeups & LLM usage” (<https://discourse.aicrowd.com/t/algorithmic-contribution-prize-guidelines-how-arc-judges-these-prizes-discretion-technical-writeups-llm-usage/18041>, archived `sources/discourse-18041-20260808T111513Z.json`) — the guideline text §10 follows, also cited as Phase 1’s citation [2]; team 504aldo, “Everything we tried: a factorized $K=3$ cumulant-propagation estimator at $0.25 \times B$, where its FLOPs go, and 25 measured dead ends” (<https://discourse.aicrowd.com/t/everything-we-tried-a-factorized-k-3-cumulant-propagation-estimator-at-0-25-x-b-where-its-flops-go-and-25-measured-dead-ends-team-504aldo-rank-10/18218>, accessed 2026-09-10, archived `sources/discourse-18218-20260910T173135Z.json`) — the disclosed V17-V25 method §2 builds on and the dead-end #10 agreement cited in §5 and §8.

[3] W. Wu, V. Lecomte, M. Winer, G. Robinson, J. Hilton, P. Christiano (Alignment Research Center), “Estimating the expected output of wide random MLPs more efficiently than sampling,” arXiv:2605.05179 (2026). <https://arxiv.org/abs/2605.05179> (re-fetched 2026-09-17; same identification as Phase 1’s already-verified citation [3]). **Correction to the graded code’s own comment:** every candidate file in `research-phase2/qw62/`, `research-phase2/qw63/` and `research-phase2/qw64/` (including the earlier paired `cand_qw68_einsum_l5_ship.py`), and the currently paired `qw91_core.py:50-51`, labels this arXiv id in a docstring as “Hoogland et al., ‘Cumulant propagation’” — in `qw62/` traced to a single template (`cand_qw62_template.py`) copy-pasted across that family; `qw63/`’s files carry the same label in differently-wrapped wording, so its own provenance is not established here. Re-fetching the id directly returns the title and author list above, matching [3]’s Phase-1 citation exactly; there is no second paper at this id. Left uncorrected in the graded artifacts themselves, since editing already-submitted candidate files is out of scope here; the correction is recorded so this document does not repeat it.

[4] R. Price, “A useful theorem for nonlinear devices having Gaussian inputs,” IRE Transactions on Information Theory 4(2):69–72 (1958). <https://doi.org/10.1109/TIT.1958.1057444>.

[5] R. L. Plackett, “A reduction formula for normal multivariate integrals,” Biometrika 41(3/4):351–360 (1954). <https://doi.org/10.1093/biomet/41.3-4.351>.

[6] Z. Drezner, G. O. Wesolowsky, “On the computation of the bivariate normal integral,” Journal of Statistical Computation and Simulation 35(1-2):101–107 (1990). <https://doi.org/10.1080/00949659008811236> — the $u = \sin \theta$ substitution behind §4’s exact-covariance identity.

[7] N. Halko, P.-G. Martinsson, J. A. Tropp, “Finding structure with randomness: probabilistic algorithms for constructing approximate matrix decompositions,” SIAM Review 53(2):217–288 (2011). <https://doi.org/10.1137/090771806> — the randomized range finder with one subspace iteration, lifted in Appendix A.4.

[8] V. Strassen, “Gaussian elimination is not optimal,” Numerische Mathematik 13(4):354–356 (1969). <https://doi.org/10.1007/BF02165411> — the 7-multiply 2×2 block product behind §2’s cost restructures.

[9] S. Winograd, “On multiplication of 2×2 matrices,” Linear Algebra and its Applications 4(4):381–388 (1971). [https://doi.org/10.1016/0024-3795\(71\)90009-7](https://doi.org/10.1016/0024-3795(71)90009-7) — the 15-addition variant used in §2 and A.5.

[10] S. Rosenbaum, “Moments of a truncated bivariate normal distribution,” Journal of the Royal Statistical Society Series B 23(2):405–408 (1961). doi:10.1111/j.2517-6161.1961.tb00422.x (Crossref-resolved 2026-09-29.)

[11] A. Bibi, M. Alfadly, B. Ghanem, “Analytic expressions for probabilistic moments of PL-DNN with Gaussian input,” CVPR 2018. doi:10.1109/cvpr.2018.00948 (Crossref-resolved 2026-09-29.)

[12] Hisernberg, arc-whitebox (GitHub repository, <https://github.com/Hisernberg/arc-whitebox>, found 2026-10-07; captures `sources/rescan-gh-20261007T130745Z/` and `sources/rescan-gh-20261007T135432Z/`). Its V41 learned local-error corrector, reported graded at #334556 4.063198653618492e-09, is the idea behind §2.3 C1 (plan :33725–:33726). The repository carries no licence for its own code; we copied no code or weights from it. The training hyperparameters of our trainer (6 epochs, width 128, AdamW at $2e-3$, OneCycle schedule; research-

phase2/qw194/train_lt.py) coincide with those of the repository’s trainer (sources/rescan-gh-20261007T130745Z/hisernberg/work_lt_train_lt.py).

Appendix A — code, verbatim

Rubric criterion (iii) asks how easily the actual performance impact can be determined “from the code and writeup together.” A.1-A.4 are carried over from the 2026-10-01 text. They show the hub restructure and the range finder of the earlier Q-W62 chain (research-phase2/qw62/cand_qw62_r15_r64_pi1.py, graded #330715), which is not the paired artifact; they are kept because §3 and §5 discuss that structure. Following the earlier text, their comment-only and blank lines are dropped, and the single # ... marker in A.3 marks an elided span (the earlier text generated them with research-phase2/writeup/appendix_gen.py). A.5 and A.6 are new. They come from the paired folder and are copied as contiguous line ranges, comments included.

What is deliberately not here, because we do not claim it: the Wick and power-cumulant term tables (_WICK, _TERMS, _PK2K), which are ARC’s own mlp_kprop closure re-derived, not ours; and the corrector weights (lt_weights.py), which are data.

A.1 — the rank and seed constants

```
_W_VAR = 2.0 # E[W W^T] = w_var I for whestbench weights N(0, 2/n): the average kappa_4 metric
_COST_FRACTION = 0.85
_R_RES = 64
_SEED = 0
```

A.2 — hub restructure: per-layer transport and consumption of the shared legs

```
sources = []
rows = []
for w in ws:
    wt = fnp.asarray(w, dtype=_F32).T
    mean = wt @ mean
    cov = wt @ cov @ wt.T
    for s in sources:
        s[0] = wt @ s[0]
        s[1] = wt @ s[1]
        s[2] = wt @ s[2]
    var = fnp.maximum(fnp.diag(cov), 1e-10)
    sigma = fnp.sqrt(var)
    alpha = mean / sigma
    phi = flops.stats.norm.pdf(alpha).astype(
        _F32
    ) # scipy-style promotion to float64 would double every bill
    Phi = flops.stats.norm.cdf(alpha).astype(_F32)
    wick = {}
    for (k, p), (e, p1, p2) in _WICK.items():
        val = 0.0
        if p1:
            val = _poly(p1, alpha) * phi
        if p2:
            val = val + _poly(p2, alpha) * Phi
        wick[(k, p)] = val * sigma**e if e else val
    w1, w2 = wick[(1, 1)], wick[(2, 1)]
    t3 = fnp.zeros(n, dtype=_F32)
    t21 = fnp.zeros((n, n), dtype=_F32)
    for A, P, Z, Lt, k3v, e, d2 in sources:
        X = P * k3v + 3.0 * A * e + Z @ Lt
        AP = A * P
        PP = P * P
```

```

t3 = t3 + fnp.sum(3.0 * (AP * A) * d2 + X * PP, axis=1)
t21 = (
    t21
    + (2.0 * AP * d2 + PP * e) @ A.T
    + ((A * A) * d2 + (2.0 / 3.0) * (X * P) + PP * (k3v * (1.0 / 3.0)))
    @ P.T
    + ((PP @ Lt.T) * (1.0 / 3.0)) @ Z.T
)
t21 = t21 * off

```

A.3 — hub restructure: constructing and appending a newborn source

```

if use_k3:
    a_new = w1[:, None] * c_off
    k3 = K[(3,)] - (w1 * w1 * w1) * t3
    k21 = K[(2, 1)] - (
        (w1 * w1)[:, None] * t21 * w1[None, :]
        + (a_new * a_new) * w2[None, :]
    )
    for s in sources:
        s[0] = w1[:, None] * s[0]
        s[1] = w1[:, None] * s[1]
        s[2] = w1[:, None] * s[2]
    e = wick[(1, 2)] - 2.0 * pk[(1,)] * w1
    res = (3.0 * k21.T) * off - 3.0 * a_new * e[None, :]
# ...
sources.append([a_new, eye * 1.0, q, q.T @ res, k3, e, w2])

```

A.4 — range finder: rank-r Gaussian subspace approximation of the X leg’s residual

```

om = rng.standard_normal((n, r)).astype(
    _F32
) # fnp.random draws float64
q, _ = fnp.linalg.qr(res @ om)
q2, _ = fnp.linalg.qr(res.T @ q)
q, _ = fnp.linalg.qr(res @ q2)

```

R1.6’s one stabilized subspace iteration (the second q2/q pair) is exactly Halko–Martinsson–Tropp [7] Algorithm 4.4 (randomized subspace iteration) at one step — the QR re-orthogonalization between applications is what distinguishes 4.4 from their unstabilized Algorithm 4.3, needed here because float32 on `res @ res.T` would square the condition number, per the source’s inline comment at that line. At $r = n$ this range finder is exact ($Z Lt = Q Q^T R = R$); at $r = 64 < n$ it adds the rung’s only approximation relative to its own $r = n$ variant; A.2–A.3 are bookkeeping of the already-derived (and itself approximate) $K=3$ cumulant closure.

A.5 — earlier-lineage artifact #335249: the level-batched Strassen-Winograd routine with the sw15 node, and the einsum rewrite (ship_cls5c5n5mf/qw91_core.py:663-734 and :766-768, contiguous, comments kept)

```

def sw_matmul_lb(a, b, level=1, out=None, _depth=0):
    """Level-batched Strassen-Winograd. Same contract as strassen.sw_matmul:
    level 0 is exactly fnp.matmul; odd dims fall back to naive (never wrong).
    """
    if level <= 0:
        return fnp.matmul(a, b, out=out) if out is not None else fnp.matmul(a, b)

    m, kk = a.shape[-2], a.shape[-1]
    n = b.shape[-1]
    if m % 2 or kk % 2 or n % 2:
        return fnp.matmul(a, b, out=out) if out is not None else fnp.matmul(a, b)

```

```

hm, hk, hn = m // 2, kk // 2, n // 2
# Q-W233: equal batch shapes need no fnp call (shape bookkeeping only; value-identical, saves
batch = tuple(a.shape[:-2]) if tuple(a.shape[:-2]) == tuple(b.shape[:-2]) else fnp.broadcast_s
if out is None:
    out = fnp.empty(batch + (m, n), dtype=_F32)

A11, A12 = a[..., :hm, :hk], a[..., :hm, hk:]
A21, A22 = a[..., hm:, :hk], a[..., hm:, hk:]
B11, B12 = b[..., :hk, :hn], b[..., :hk, hn:]
B21, B22 = b[..., hk:, :hn], b[..., hk:, hn:]
C11, C12 = out[..., :hm, :hn], out[..., :hm, hn:]
C21, C22 = out[..., hm:, :hn], out[..., hm:, hn:]

ash = tuple(a.shape[:-2]) + (hm, hk)
bsh = tuple(b.shape[:-2]) + (hk, hn)
csh = tuple(batch) + (hm, hn)
d = _depth

# Q-W107 T1: ONE pooled buffer per side per level; the named intermediates are leading-index v
Abuf = _buf(
    (7,) + ash, ("Abuf", d)
) # Q-W142b: the stacked left operands ARE the sum slots -- 11 blocks -> 7
Bbuf = _buf(
    (7,) + bsh, ("Bbuf", d)
) # Q-W142b: the stacked right operands ARE the sum slots -- 11 blocks -> 7
Cbuf = _buf((10,) + csh, ("Cbuf", d)) # 7 sub-products = [0:7], U2..U4 = [7:10]

# --- Q-W142b: the 8 pre-additions write STRAIGHT INTO THEIR STACK SLOTS, and the
# --- input views are grouped into ONE `stack` per side rather than three `copyto`.
# The shipped form put S1..S4 in Abuf[0:4] and let `fnp.stack` copy them into [4:11];
# flopscope bills `stack` 1.0 per produced element, so each sum was billed twice.
# Slot order IS the stack order, so only the input views still need a copy: 11 billed
# blocks per side per level -> 7, measured -2.5512% of bill on MLP 508 with `out_sha`
# byte-identical to the parent. `stack` and `copyto` are both weighted 1.0, so grouping
# the copies into one `stack` bills the same and costs 1 call instead of 3 -- arm_sws
# used three `copyto` per side and paid x4.13 on WALL for it (make_qw142b.py docstring).
fnp.stack([A11, A12, A22], axis=0, out=Abuf[1:4])
S1 = fnp.add(A21, A22, out=Abuf[5])
S2 = fnp.subtract(S1, A11, out=Abuf[6])
fnp.subtract(A21, A11, out=Abuf[4]) # Q-W239 sw15: S3' = -S3 (exact)
S4 = fnp.subtract(A12, S2, out=Abuf[0])
fnp.stack([B22, B11, B21], axis=0, out=Bbuf[0:3])
fnp.subtract(B12, Bbuf[0:2], out=Bbuf[4:6]) # Q-W239 sw15: [T3', T1] = B12 - [B22, B11], T3'
T1 = Bbuf[5]
T2 = fnp.subtract(B22, T1, out=Bbuf[6])
T4 = fnp.subtract(T2, B21, out=Bbuf[3])
A_stack = Abuf
B_stack = Bbuf
M_stack = sw_matmul_lb(A_stack, B_stack, level - 1, out=Cbuf[0:7], _depth=d + 1)
M3, M1, M2, M4, M7, M5, M6 = (M_stack[i] for i in range(7)) # Q-W239 sw15: M7 = S3'.T3' = S3.

# --- 7 post-additions in 6 calls, four straight into `out` -----
fnp.add(M1, M2, out=C11)
U2 = fnp.add(M1, M6, out=Cbuf[7])
fnp.add(U2, M_stack[4:6], out=Cbuf[8:10]) # Q-W239 sw15: [U3, U4] = U2 + [M7, M5]

```

```

U3, U4 = Cbuf[8], Cbuf[9]
fnp.add(U4, M3, out=C12)
fnp.subtract(U3, M4, out=C21)
fnp.add(U3, M5, out=C22)
return out

```

```

def _sum_bmm_lb(X, Y):
    """sum_k X_k @ Y_k^T (einsum "kij,kcj->ic") via level-batched Strassen-Winograd."""
    return fnp.sum(sw_matmul_lb(X, fnp.swapaxes(Y, -1, -2), _V27LB_LEVEL), axis=0)

```

The routine is called at the restructured sites of #335249, among them the young-leg transport (level `_V27LB_LEVEL = 5, :534`), every `_sum_bmm_lb` contraction (same level), the formed legs (`_LV_QB = 4, :738`), the newborn transport (`_LV_NB = 5, :739`) and the covariance sandwich (`_QW75_LC = 5, :740`). Level 0 returns `fnp.matmul` unchanged, and odd dimensions fall back to a plain product. The four # Q-W239 sw15 lines (:714, :717, :724, :729) are the whole of C3; `unit_sw15.py` compares this routine against its 17-operation parent.

A.6 — the redesigned wall fail-safe (ship_c1sv5c5n5mf/qw91_core.py:357-363 and :1344-1348)

```

# Q-W169 wall fail-safe: the host stops a predict at 120 s wall and scores that MLP as a zero
# prediction. Before each layer, if the elapsed wall plus _WALL_GROW x the last layer's wall
# would pass _WALL_LIMIT_S, the remaining layers are finished by a Gaussian closure started from
# this core's own mean and covariance at that point. Below the limit nothing changes: the check
# reads the clock and adds no flopscope op.
_WALL_LIMIT_S = float(os.environ.get("ARC_WALL_LIMIT_S", "116"))
_WALL_GROW = 1.3
# ...
    t_prev = t0 = time.monotonic()
    for li, w in enumerate(ws):
        t_now = time.monotonic()
        if li and (t_now - t0) * (1.0 + _WALL_GROW / li) > _WALL_LIMIT_S: # Q-W257 mf: mean-
            return _wall_fallback(ws, rows, mean, cov, sl_perm)

```

The comment at :357-361 still describes the old last-layer rule; the predicate at :1347 is the mean-layer forecast of §2.3 C6. In the gates `ARC_WALL_LIMIT_S` is set to `1e6`, so the predicate never fires there; the trip test sets it to `40`.

Appendix S — sources

All paths are relative to `competitions/arc-whitebox/` unless they start with `.claude/loop/`.

Kind	Path	Used for
Host reports (grader)	<code>sources/host-report- <id>.json</code> (<code>results.aggregates.public</code> , <code>results.per_mlp[]</code> .telemetry, <code>runtime_environment</code> , <code>config_snapshot</code>)	every MSE, multiplier, residual, wall, FLOP and failure count in §0-§2, §6, §7, §9
Host-report table	<code>research- phase2/qw211/hostreport.py</code> <code>table <ids></code> (run 2026-10-11 under <code>comp10p2</code>)	§0, §2.3
Submission ledger	<code>.claude/loop/submissions.log</code> (timestamped <code>READBACK</code> <code>rows</code>)	full-precision graded scores
Plan	<code>.claude/plans/PLAN_arc- whitebox-master.md</code> , cited as “plan :N”	measurements, pre-registrations, KILL reads (§2.3, §6, §8)

Kind	Path	Used for
Job logs	.claude/loop/state/jobs/arc/loop/arc-20261009T025023Z-1b4de0.log, arc-20261009T034120Z-3ba6ec.log, arc-20261009T034121Z-b6c706.log, arc-20261010T172635Z-899be5.log, arc-20261010T172635Z-d895df.log, arc-20261010T182808Z-012162.log	paired gate ratios, censuses, trip tests (\$2.3, \$2.4)
Candidate registry	.claude/loop/state/candidates/row c10101903573bd8	registered hash prefix of the paired core (\$9)
Challenge page	sources/challenge-page-jina-20260930T162953Z.md (:121, :123, :147, :153-:164, :190, :250)	Phase-2 cost model, caps, score formula (\$1)
Rules	sources/challenge-challenge_rules-20260807T170700Z.DECODED.txt (:251-:265)	algorithmic-contribution prize terms
Board	sources/leaderboard-1429-20261011T000747Z.json	rank 47 of 280 (header of the 2026-10-11 text)
V29 reference grade	sources/grade-poll-refs-20260922T041410Z.json	\$7
Code	research-phase2/qw239/ship_clsv5c5n5mf/, qw239/{make_sw15.py,unit_sw15.py,make_yq2.py,unit_yq2.py,chain_sw.sh}, qw236/make_ap.py, qw257/{make_mf.py,chain_mf.sh}, qw194/	\$2, App. A