Stabilizing Cumulant Propagation at Depth 32: A Trajectory-Calibrated Moment Chain

Technical write-up for the ARC White-Box Estimation Challenge 2026, Phase 1 (algorithmic contribution submission). James Henry (jamesrahenry). 2026-07-03, revised through 2026-07-24; erratum added 2026-08-04; revision history in "Disclosure and provenance" at the end.

Disclosure: this work was AI-assisted (Claude, Anthropic). The model proposed experiments, wrote the code, and drafted this document; the author directed the research and reviewed the results. Quantitative claims trace to named scripts and committed results files, with three marked exceptions at point of use, and the document was independently reviewed twice against the underlying scripts and data. Details are given in "Disclosure and provenance" at the end.

Erratum (2026-08-04): grader cost-model repricing under FLOPScope v0.10.0

On 2026-08-04 the organizers released FLOPScope v0.10.0 (forum post 18125) and regraded all affected submissions in place. The cost model is now dtype-aware — float64 operations bill at 2× the float32 rate, priced by output dtype — and data-movement operations (copy/fill/concatenate at 1 per element written; gather/sort/histogram at 4) are no longer free. The scoring formula, the λ = 1e11 FLOP/s wall-time rate, the per-MLP FLOP budget, and the 0.10 multiplier floor are unchanged. Every grader-reported multiplier and adjusted score quoted in this document was graded under the previous, dtype-blind model; the body text below is left as originally graded, and this erratum states what the official record now says. **No raw MSE in this document is affected** (the regrade left #314695's raw byte-identical at 5.88962e-6), and therefore none of the claimed contributions — the diagnosis (C1), the stabilization method (C2), the ground-truth cumulant tests (C3), the error budget (C4), or the negative results (C5) — is touched by the repricing.

Corrected grader-reported values:

submission	quoted in this document (old model)	official after regrade
#314695 (this write-up's artifact)	adjusted 9.97e-7, multiplier 0.169	**adjusted 1.76842e-6, multiplier 0.300**
our sampling entry	adjusted 3.17e-7	regraded 5.884e-7; superseded by a float32 resubmission, **#323492, adjusted 3.07e-7** (raw identical to 5 digits)
pscamillo, #314331 (§6)	adjusted 2.45e-6	2.4501e-6 — unchanged (at the 0.10 multiplier floor, insensitive to the repricing)

submission	quoted in this document (old model)	official after regrade
radiant-allomancer, #317660 (§6)	adjusted 4.47e-7	4.98e-7

Why #314695 moved the most of these: its hot path runs in default float64 (the probe draws float64 normals, so every matmul against the grader's float32 weights promotes to the 2× rate). Re-metered under flopscope 0.10.0 (`repricing_erratum_check.py`, committed alongside the other artifacts), the graded file bills 7.4031e10 FLOPs per net versus 3.7040e10 for a float32 port with identical arithmetic (ratio 1.999); the port changes the returned per-layer means by at most 2.7e-6 against a per-element RMSE of ~2.4e-3, so raw MSE is unaffected. Scaling the regraded multiplier by the measured bill ratio puts a float32 resubmission at multiplier ≈ 0.15, adjusted ≈ 0.9e-6 — essentially the numbers originally quoted. We have not resubmitted: the contribution claimed for this artifact is mechanistic, not positional, and the cost model does not touch it.

Two passages should be read with corrections. First, the wall-time paragraph following the #314695 summary table attributes the gap between the FLOP share (13.6%) and the multiplier (0.169) to Python-dispatch wall time; under v0.10.0 the regraded multiplier (0.300) is accounted for by the FLOP bill alone, the wall-time term having become negligible (v0.10.0 also moved the metering backend to seven physical cores). The paragraph's closing claim should now read: dtype, not wall-time batching, is the dominant multiplier lever for this estimator family. Second, the economics line in §3 ("multiplier ≈ 0.14–0.3 … adjusted score is ~1e-6") reads, under the new model with a float32 hot path, multiplier ≈ 0.15–0.30 and adjusted ≈ 0.9–1.8e-6; the conclusion — well short of the sampling frontier — is unchanged, and indeed strengthened: the repricing penalized dtype-naive mid-multiplier entries like this one hardest, while the frontier entries sit at the multiplier floor and barely moved. The §5 ceiling statement ("4–5e-7 adjusted at the multiplier floor") is unaffected, as the floor itself is unchanged; the sampling replica expectation (≈4.5e-7) was measured under the old model and shifts by only a few percent for a float32-clean implementation (our graded pair: 3.17e-7 → 3.07e-7).

Summary

The central finding of this work is that a deep moment-propagation chain behaves as an error-compensating dynamical system. Its per-layer errors are not independent; they are anticorrelated and partially cancel, so the chained estimator attenuates zero-mean noise while amplifying small coherent biases (a measured output exchange rate of approximately 16:1; §5). This single property accounts for four otherwise separate observations: step-exact Edgeworth corrections computed from true cumulants degrade the chained estimator (§1c); corrections succeed only when fitted on the chain's own rolled-forward trajectories (§2, the basis of the graded submission, an ~8.4× improvement over the published baseline); a variance-reducing shrinkage filter doubles the final error despite reducing noise as designed (§5); and an independent team's layerwise bias re-anchoring failed in the same manner (§6).

ARC's published cumulant-propagation algorithm ("kprop" [1]) is accurate at depth 4 but degrades with depth, per its authors; we take that claim from their paper and start directly at the Phase-1 spec. At depth 32 (width 256, He-Gaussian, bias-free ReLU), we measure plain kprop (ARC's reference implementation, k_max=2) at mean 6.28e-5, median 5.4e-5, std 3.8e-5 final-layer mean MSE over 40 real challenge nets (`kprop_depth32.py`). This is approximately 39× worse than our own sampling-family estimator's raw MSE at the same Phase-1 FLOP budget (1.61e-6, the expectation measured on the full 1,000-net public split with the starterkit local evaluation engine, rather than the more favorable single-seed graded number; see Disclosure and provenance).

This write-up makes five contributions, all on real challenge nets, with every number traced to a committed script and results file:

C1 — Diagnosis: the depth failure is error dynamics, not closure order (§1). With true input moments, a single k=2 closure step has error (1.16e-6) below the total raw error of a strong sampling estimator at full budget (1.61e-6, our own entry); chained, the same closure is 52× worse. The chain accumulates fresh mean bias each layer while its covariance errors partially compensate: re-anchoring the covariance to truth makes the estimate worse, and step-exact Edgeworth corrections from true cumulants degrade the chain. We argue this compensation structure is the reason naive corrections to deep moment chains fail generally.

C2 — Method: trajectory-fitted stabilization (§2). Fitting per-layer linear corrections on the chain's own rolled-forward state (DAgger-style), rather than against step-local truth, reaches 7.4e-6 held-out at seed-resampled expectation (5.2e-6 on the fixed development probe seeds; the distinction, quantified during the §5 work, is explained in §2). This is ~8.4× better than the plain-kprop baseline (~12× on the development seeds), within 13.6% of the FLOP budget, graded as submission #314695 (summary below). This is a mechanistic-contribution result rather than a competitive one: 5.9–7.4e-6 raw remains well above our own sampling entry's 1.61e-6 (§3).

C3 — Ground-truth joint-cumulant tests (§4). Against keenanpepper's published full third-cumulant tensors [2] (Hugging Face dataset `keenanpepper/whest-k3-tensors-2026`; locked IDs; evaluation-only policy and promotion gate committed up front, §4), we show the joint structure missing from the closure is real and compressible, and that an analytic carrier of it, despite good reconstruction (cosine 0.83), is still 4.5× worse than the low-cost MC probe already feeding the estimator, on 10 of 10 IDs. These are tensor-level rather than probe-based checks of the hypothesis.

C4 — An error budget for the graded artifact (§5). A seed-variance decomposition shows roughly half of #314695's final error is probe-sampling noise, attributed by field (the pair field dominates; its noise anticorrelates with κ3's, so denoising κ3 alone makes the result worse), with a measured denoising ceiling (0.48–0.58×) and three failed exploitation attempts. The general mechanism: an error-compensating recurrent chain tolerates variance far better than bias. Measured end-to-end, our shrinkage filter traded 4.2e-7 of variance for 6.6e-6 of squared bias, an exchange rate of ~16:1 against the filter, so bias-for-variance denoisers are structurally unsuited to chained estimators.

C5 — Negative results with mechanisms, and an open problem (§§5–7). Each

closed avenue carries its measured reason, and three are independently corroborated by concurrent write-ups (§6). The open problem is well-defined: a lower-variance unbiased estimator of $\{\kappa 3, E[zc^2_i zc_j]\}$ at fixed sample count would be worth up to ~2× raw MSE to any probe-fed correction scheme; shrinkage-family estimators are excluded by the bias-amplification dynamics above, not by their variance performance.

Sampling methods win Phase 1's raw metric, and this document says so plainly. The case for the analytic track is the motivating regime rather than the current benchmark: settings where sampling itself is unreliable, including rare events, distribution shift, and the evaluation of non-random models whose behavior under test may differ from deployment (§8). The diagnosis, the stabilization method, and the error budget are offered as contributions to that program.

The referenced graded submission: #314695

This write-up refers to submission #314695 (graded 2026-07-03): `estimator_chain.py`, a single-file `fnp` estimator implementing the method of §2, i.e. a (mean, covariance) propagation chain with a K=10 Hermite-series bivariate Gaussian ReLU closure, corrected per layer by the trajectory-fitted stabilizer (512 coefficients, embedded), with cumulant inputs from an N=4096 plain-MC probe seeded deterministically from each MLP's own seed.

grader-reported (all independently rounded)	value
raw final-layer MSE	5.89e-6
effective-compute multiplier (FLOPs 13.6% of budget + wall time)	0.169
adjusted score	9.97e-7

The gap between 13.6% (FLOPs) and 0.169 (multiplier) is the grader's wall-time term, which bills residual wall-clock seconds at 1e11 FLOP-equivalents per second. For this estimator it is dominated by Python-level dispatch of the 32 per-layer Hermite-series loops under `fnp`, not by the probe's few large matrix multiplications; an earlier version of this chain family (v6.2-v6.3 in the working ledger) reduced this term roughly tenfold by batching, and further reduction toward the 0.10 floor is an engineering matter rather than an algorithmic one.

The graded raw score sits between the fixed-development-seed validation value (5.2e-6, nets 40-49) and the seed-resampled expectations (7.45e-6 on the same nets; 8.13e-6, SEM 0.79e-6, on 56 fresh nets); §2 gives the full accounting. It is an independent draw on ~100 grader nets disjoint from the public atlas, with per-net deterministic probe seeds, so the graded number is reproducible rather than a re-rollable realization. The post-hoc decomposition of §5 applies to exactly this artifact: roughly half of its remaining error is probe-sampling noise, attributed by field, with a measured ceiling on what denoising could buy and measured reasons the inexpensive routes fail. As stated in the Summary, this submission is the mechanistic artifact this write-up

concerns; our own sampling entry scores better (graded 3.17e-7 adjusted; scoring-replica expectation ≈4.5e-7, the graded draw being a favorable pairing of a fixed scramble with the fixed public eval set) and is not the subject here. The claimed contribution is the diagnosis, the stabilization method, and the error budget, not the leaderboard position.

1. Diagnosis: the depth problem is dynamics, not physics

Three experiments, all on real challenge nets with 1e8–1e9-sample ground truth:

(a) One-step vs full-chain decomposition (`kprop_drift.py`). Feed each layer's closure step the true input moments and measure output error, layer by layer:

	layer 32
one-step mean MSE (true inputs)	1.16e-6
full-chain mean MSE	6.0e-5 (52×)

Second-order state (mean + covariance) is sufficient at depth 32: the one-step Gaussian closure error is already below the total raw error of our own full-budget sampling estimator (1.61e-6, Summary). This falsifies the natural hypothesis, which we initially held and which the "K3/K4 needed at depth" framing suggests, that closure order is the depth bottleneck.

(b) Which state component drifts (`chain_reset.py`). Re-anchoring one component to truth at every layer boundary:

reset	final mean MSE
nothing (plain chain)	6.0e-5
mean only	1.64e-6
variances only	1.8e-4 (worse)
full covariance only	1.0e-4 (worse)

The chain accumulates fresh mean bias every layer, while its covariance errors partially compensate the mean bias; fixing the covariance alone breaks the compensation and makes the estimate worse. We believe this compensation structure is the reason naive corrections to moment chains at depth have failed generally (including our own earlier 25-variant stabilization sweep, and plausibly others').

(c) Step-exact corrections degrade the chain (`edgeworth_ceiling.py`, `chain_v2.py`). Edgeworth corrections with true marginal cumulants reduce the per-step biases substantially in isolation: mean bias 2.7e-6 → 1.5e-8 (κ3+κ4), post-second-moment bias 4.7e-6 → 3.4e-9 (closed forms), off-diagonal covariance residual $R^2 = 0.954$ with the derived cross-cumulant term (fitted coefficient 1.0038 vs the theoretical 1.0). Yet injecting these same corrections into the running chain makes it worse than no correction (8.8e-5 vs 6.3e-5). Step-local accuracy does not compose in an error-compensating dynamical system.

2. Method: trajectory-fitted stabilizer

The remedy is to fit corrections on the chain's own trajectories rather than against step-local truth, so that compounding and compensation are part of the training signal. This is not a new principle; it is the idea behind DAgger [3] and scheduled sampling [4] in imitation learning (train the correction against the model's own rolled-out state, not ground-truth teacher-forced state, so the training distribution matches the deployment distribution) and behind innovation-based recalibration in adaptive filtering [5] (correct against the filter's own residual trajectory, not an idealized one). What is new here is narrow and specific: showing that a deterministic analytic propagation chain, rather than a learned sequence model or a stochastic filter, has the same compounding-and-compensation failure mode, and that the same remedy applies.

The estimator's structure, per net:

```
MC probe (once per net, N=4096, ~6% of the FLOP budget):
    x ~ N(0, I) forwarded through all 32 layers; at each layer l it records
    the cumulants κ3(l), κ4(l), and the pair field E[zc²ᵢ zcⱼ](l)

per-layer update (l = 1 .. 32):
    (μ, Σ)  ──W_l──►  (μ_pre, Σ_pre)
    (μ_pre, Σ_pre)  ──K=10 Hermite ReLU closure──►  raw closure (m̂, Ŝ)
    (μ_pre, Σ_pre, probe cumulants at l)  ──►  8 mean + 6 variance +
                                               2 off-diagonal features
   (m̂, Ŝ) + per-layer linear corrector on those features  ──►  (μ, Σ) for l+1

corrector coefficients: 16 per layer × 32 layers = 512 total, fitted once
offline (DAgger-style, described below) and embedded in the submission file
```

- Chain state: (mean, covariance), propagated by a Hermite-series bivariate Gaussian ReLU closure [6, 7] (exact diagonal; series `E[relu(u)relu(v)] = σuσv Σk ak(αu)ak(αv)ρ^k` with `a1 = Φ(α)`, `a2 = φ(α)/2`, `ak = He_{k−2}(−α)φ(α)/k!`, truncated at K=10).
- After each layer's closure step, apply linear corrections to the mean (8 features), the variance diagonal (6 features), and the covariance off-diagonals (2 matrix features built from the derived cross-cumulant term), with per-layer coefficients (512 numbers total).
- Cumulant inputs (κ3, κ4, $E[zc_i^2\, zc_j]$ per layer) come from a small plain-MC probe (N=4096, ~6% of the FLOP budget, seeded from the per-MLP seed). QMC probes are worse here; low-discrepancy point sets help means, not higher central moments (measured: 8.2e-6 vs 6.5e-6 held-out).
- Fitting is sequential per layer (DAgger-style): fit layer l's coefficients on training states rolled forward with corrections 0..l−1 already applied. Training data: 40 nets from keenanpepper's public per-layer moment atlas [8] (Hugging Face dataset `keenanpepper/arc-whestbench-higher-moments-2026`); validation: 10 held-out nets (indices 40-49).

On that validation set. Nets 40-49 were reused as the held-out check across all five design iterations below (v3 through the portable-step refit) while the correction family and features changed each time. "Train ≈ val" at any one step shows the

fit is not overfitting the training nets, but does not rule out the design sequence having drifted toward what happens to work on these particular 10 nets. To check that directly, we scored the final, frozen coefficients, with no refitting, against every atlas net never touched by any design iteration (indices 50-105, 56 nets; `chain_port_refit_freshval.py`): mean 8.13e-6, median 6.49e-6, std 5.89e-6, SEM 0.79e-6, against an uncorrected-baseline mean of 7.55e-5 on the same 56 nets. This is an ~9.3× improvement that never influenced any design decision (a 40-net subset check agrees: mean 7.99e-6, SEM 0.96e-6).

Seed sensitivity of the validation figures. The 5.2e-6 held-out value is a favorable realization of the single fixed development probe-seed family (`seed=1000+idx`) reused across design iterations: re-scoring the identical refit on the same nets 40-49 with fresh probe seeds gives 7.45e-6 (`chain_refit_filtered_probe.py`, `val_mse_fresh_seeds`, 4 seeds × 10 nets), consistent with §5's finding that roughly half the per-seed MSE is probe-noise variance. The gap to the 56 fresh nets therefore decomposes as: 5.2 → 7.45 (~1.43×) is probe-seed variation on the development seeds, and 7.45 → 8.13 (~1.09×, within the 56-net SEM) is the net-selection cost of reusing the validation set during design. Where this document quotes 5.2e-6 it is the fixed-development-seed value; the seed-resampled expectation 7.4e-6 is the representative single-run figure. The grader score on #314695 (raw 5.89e-6) is an independent draw on disjoint nets with per-MLP seeds and sits between the two, consistent with this picture.

Result trajectory (uncorrected baseline through progressively refined stabilizer, all on the same nets 40-49 validation set unless noted): our portable K=10 Hermite-closure reimplementation's own uncorrected baseline 7.3e-5 (a from-scratch reimplementation, required because the grader sandbox has no `torch`/`mlp_kprop`; it is not ARC's reference kprop measured in the Summary, and the two closures agree to within ~15%, as expected for the same closure order with different truncation detail) → v3 1.07e-5 → v4 6.5e-6 → portable-step refit 5.2e-6 held-out on the fixed development seeds; 7.45e-6 seed-resampled (see the seed-sensitivity paragraph above; train ≈ val throughout, so the remaining gap is correction-family capacity rather than overfitting within a single fit; see the caveat above on the fitting sequence).

3. Limitations

- Perfect-cumulant control (v4-era, refit and evaluated with atlas-true $\kappa 3/\kappa 4/E[zc^2_i zc_j]$): 4.36e-6 vs 6.52e-6 probed (`chain_v4_stabilizer.py`), an early indication that probe noise is material. §5 sharpens this with a per-seed variance decomposition on the deployed estimator itself, a different instrument (frozen coefficients, seed-resampled probe) giving a larger share; the two are consistent in direction.
- The correction family plateaus, and we tested the obvious escapes. Quadratic (nonlinear) correction heads are worse at 40 training nets (6.40e-6 vs 5.21e-6, overfitting); with per-net values (`chain_v6_nonlinear.py 96 10`, 10 held-out nets, n=10 each): quad mean 7.457e-6 (std 2.63e-6) vs lin mean 7.517e-6 (std 3.93e-6), a paired difference of −6.0e-8 ± 8.3e-7 SEM, i.e. indistinguishable from zero rather than merely close in rounded means. Neither more capacity nor 2.4x more data moves the plateau: the residual is structural to per-neuron/per-pair corrections and appears to live in joint covariance structure they cannot express. We

also tested corrections parameterized in the covariance eigenbasis, i.e. an additive rank-32 term reconstructed from the probe's own κ3/κ4, projected onto the top-32 eigenvectors of each layer's analytic pre-activation covariance (directions an earlier diagnostic showed carry 72% of the true third-cumulant mass). Negative both ways: added alongside the existing T1/T1·ρ features it is flat-to-worse (paired diff −5.1e-7 ± 2.7e-7 SEM at N=16384 probe, n=10 held-out nets; the effect became more negative, not less, as probe noise fell, ruling out insufficient probe quality as the cause), and used in place of T1/T1·ρ it is ~15× worse (6.02e-5 vs 3.99e-6). The naive probe-reconstructed rank-k term is not a valid closure correction the way T1 is (T1 derives from the conditional-expectation structure of the Hermite closure; the eigenbasis term is a rescaled sum of outer products with no such grounding). This closes the tested form of the lever, not necessarily every eigenbasis parameterization. Full ledger: `moment_chain/README.md`.

- Economics: at 5–7e-6 raw and multiplier ≈ 0.14–0.3 (FLOPs 13.6% + wall time), the adjusted score is ~1e-6, well short of the sampling frontier. The contribution here is the diagnosis and the stabilization method, not the leaderboard position.

4. Ground-truth tests of joint third-cumulant structure

keenanpepper published full post-ReLU third-central-moment tensors (256^3 per layer, 32 layers, two independent 1e8-sample MC streams per net) for challenge IDs 800–809 (Hugging Face dataset `keenanpepper/whest-k3-tensors-2026`). We treat these ten IDs as a permanently locked mechanistic test set: no fitting, coefficient selection, or submission selection ever uses them. This policy, a four-criterion promotion gate, and explicit stop gates were committed in the experiment ledger alongside the first two-ID sanity readout and before every aggregate finding below (`moment_chain/FULL_K3_FOLLOWUP.md`, first commit 2026-07-17 with policy and gates in place; the ten-ID sweeps, depth grid, and all §4 numbers landed in later commits; the gate was subsequently declared failed and nothing from this line was deployed). This allows the "joint non-Gaussianity is the missing ingredient" hypothesis to be tested against ground truth rather than probe reconstructions. Four findings:

The signal is present and material. At layer 8, a first-order Edgeworth mean correction through the true next-layer weights reduces the local one-step mean MSE from 3.795e-6 (Gaussian closure) to 8.78e-7 using the full tensor (mean over all ten IDs, improved on 10/10), while the `iij/iii` planes available in marginal atlases reach only 2.13e-6 (`full_k3_carrier_eval.py`). The strictly-distinct-index bulk that marginal moment datasets omit carries most of the recoverable signal. Across a depth grid, full K3 beats Gaussian closure on 10/10 IDs at every layer ≤ 8 and on 7–10/10 deeper, where the true residual approaches zero and the published tensors' own MC noise dominates.

The signal is compressible, depth-dependently. A rank-64 covariance-eigenbasis projection retains 90.1% of cross-seed K3 signal energy at layer 8 (89.7% after contraction through the next weights), but only 25% at layer 0, rising to ~100% by layer 24. A practical carrier must be wide early and can compress aggressively late.

An analytic carrier reconstructs it adequately. A CP-form carrier (local `(1,1,2)` Hermite source + exact pair planes + a window of 3 origins transmitted through

per-layer gains, built from mean/covariance only; `full_k3_combined_source_gate.py`) reaches mean cosine 0.83 (range 0.74–0.90) to the true transported tensors when fed exact per-layer statistics, and 0.745 when fed its own self-propagated statistics (`combined_carrier_selfprop_check.py`), vs 0.54 (oracle-fed) / 0.44 (self-propagated) for the bare local source, better on 10/10 IDs in both regimes.

The carrier nevertheless loses to the probe. Substituting the carrier's κ3 into the graded estimator's own unchanged correction formula is 4.5× worse (per-ID range 3.5–5.5×) than the deployed N=4096 MC probe, on 10/10 IDs (`estimator_k3_substitution_test.py`). The probe's κ3 has 3.2% relative error against ground truth (range 2.1–4.5%); the carrier's has 42.6% (range 37–47%). An unbiased, history-untruncated simulation at 4096 samples outperforms a window-3, six-Hermite-term analytic carrier at this task. Conversely, an oracle-exact κ3 in the same formula gives 3.4× better one-step MSE (per-ID range 2.5–5.0×), so the headroom is real, and it points at the probe's sampling noise rather than at better analytic carriers. That question is taken up in §5.

This complements pscamillo's low-rank cumulant-projection negative (§6): theirs used probe-based reconstructions; ours reaches the same practical conclusion from ground-truth tensors, while also establishing that the joint-cumulant signal itself is real and compressible. The failure is in inexpensive portable carriers, not in the physics.

5. A probe-noise decomposition of the remaining error

Decomposition. Run the graded estimator (`estimator_chain.py`, imported unmodified with its grader-only dependencies stubbed) on local full-split nets with several probe seeds per net; per-seed final MSE decomposes as bias² + across-seed variance, correcting the K-seed mean's residual noise/K so the decomposition closes exactly (`probe_noise_share.py`). On 30 nets × 8 seeds: 6.27e-6 total = 2.64e-6 systematic + 3.62e-6 probe noise (58%; per-net share sd 0.12, SEM 0.02, range 0.41–0.95). One caveat: those 30 nets overlap the estimator's 40 training nets. The variance term is unaffected by the overlap, but the systematic term is optimistic there; on the held-out nets 40–49 the measured share is 46.6% (`chain_refit_filtered_probe.py`). The share is roughly half in either case. The local totals are broadly consistent with the graded raw 5.89e-6.

Attribution. Using a bit-exact injectable replica of `predict()` (verified against the real one before use), varying one probed field across seeds while holding the others fixed (`probe_noise_attribution.py`; 10 nets × 4 seeds; these and the ceiling/filter runs below all use nets 0–9, inside the training set, which is acceptable for variance-side measurements per the caveat above but is flagged here so no ratio below is mistaken for a fresh-net figure): the pair field $E[zc^2_i zc_j]$ carries a ~112% main-effect share, κ3 ~47%, κ4 ~9%. The shares exceed 100% because the κ3 and pair-field noises anticorrelate: the field's diagonal is κ3 (identical probe entries) and the two enter the chain through opposite-signed paths. Measured confirmation: denoising κ3 alone makes the final MSE 1.13× worse (`probe_denoise_ceiling.py`).

Ceiling. With 16×-larger unbiased reference probes substituted: coherent κ3+field denoising gives 0.58× the baseline MSE; denoising everything gives 0.48× (3.0e-6;

both ratios measured on the training-overlap nets, where the bias denominator is optimistic). On genuinely fresh nets the systematic floor is higher (≈4.5e-6, inferred from the 56-net fresh validation's 8.13e-6 total minus the ~3.6e-6 noise term), so even a perfect probe leaves this family around 4–5e-7 adjusted at the multiplier floor: a large improvement that would only reach parity with our sampling family's own replica expectation (≈4.5e-7 adjusted), and only at an unreachable ceiling. The value of the ceiling is mechanistic: the probe, not the closure or the correction family, is now this line's binding constraint.

Three failed exploitation attempts, each with a measured mechanism.

1. *Zero-FLOP self-calibrating Wiener filter, frozen coefficients.* Split the N=4096 probe into halves (same total FLOPs); estimate per-layer, per-correlation-bin gains from cross-half products; apply to the full-sample field (`probe_wiener_filter.py`). It reduces noise variance as designed (0.87×) and doubles the final MSE (1.95×). Mechanism: shrinkage converts zero-mean noise into a small coherent multiplicative bias, and the 32-layer recursion amplifies coherent bias while partially canceling zero-mean noise, the same compensation dynamics as §1(b) measured from the noise side. In the decomposition's own units (total = bias² + variance, both committed per net): the filter removed 4.2e-7 of output variance (3.23e-6 → 2.81e-6) and introduced 6.6e-6 of output squared bias (3.26e-6 → 9.83e-6), an exchange rate of approximately 16:1 against the filter. We consider this the most general finding of the section: an error-compensating recurrent chain tolerates variance far better than bias, so denoisers that trade bias for variance (the standard trade) are structurally unsuited here, and any correction stack calibrated on unbiased inputs will degrade under biased ones.
2. *Refit with the filtered probe.* Refit all 512 stabilizer coefficients with the filtered probe substituted, under the exact original protocol; the control leg reproduces the published 5.2e-6 anchor to three digits (5.212e-6). Filtered: 6.34e-6, 22% worse; refitting cannot absorb per-entry heterogeneous bias (`chain_refit_filtered_probe.py`).
3. *Raw + filtered features fit jointly* (the fit free to ignore the filter): 5.50e-6, better on only 4/10 held-out nets, an overfitting cost with no signal.

Also closed en route: antithetic probe pairing, ~7.5% variance reduction at matched forward passes on a one-hidden-layer toy net (`research/arc/antithetic_probe_toy.py`; a toy suffices as a bound because layer-0 linearity is the only exploitable input symmetry and ReLU breaks it immediately after, so the pairing is at its strongest one ReLU deep and cannot rescue a 32-layer probe); zeroing the analytically-exact layer-0 cumulants (an exact no-op; the fitted coefficients already null that noise); and the §4 analytic carrier as a κ3 source (4.5× worse). QMC probes were already negative (§2). Raising N pays linearly in FLOPs against a √N-variance return, a known losing trade under the multiplier.

Net. Probe noise is measured, attributed, ceiling-bounded, and resistant to every zero-cost instrument we tried. We state it as an open problem: a lower-variance unbiased estimator of $\{\kappa 3, E[zc^2_i zc_j]\}$ at fixed sample count would be worth up to ~2× raw MSE to any probe-fed correction scheme, ours included; shrinkage-family estimators are excluded by the bias-amplification dynamics above, not by their variance

performance.

6. Relation to concurrent write-ups

pscamillo's Phase-1 write-up [9], "Characterizing a systematic scale bias in Gaussian-closure activation estimation" (submission #314331, adjusted 2.45e-6; we have read the full PDF, not only the forum summary), reports a single-scalar multiplicative correction (~0.992, cross-validated as predictive rather than fit-to-benchmark) to a k=2 Gaussian-closure baseline, plus an extensive falsification map: third-cumulant propagation against ARC's own `mlp_kprop`, low-rank subspace projection of the degree-3 cumulant, temporal recurrence of the per-layer error, an Edgeworth correction built from exact ground-truth higher moments (the same public atlas we use; see below), and learned nonlinear correctors (random-forest / gradient-boosting on per-neuron features, audited against shuffled-label and overfit controls). They report the predictive signal available to these correctors decays exponentially with depth ($\tau \approx 5.1$ layers, 95% CI [4.6, 5.8]) and is statistically indistinguishable from zero by the scored final layer. Their write-up is explicit that this is a claim about the specific accessible-and-cheap families they tested, not a general impossibility result: *"We do not claim no correction can exist; we claim the accessible-and-cheap ones we tried do not."*

Our graded submission #314695 (raw 5.89e-6, adjusted 9.97e-7) does not contradict that narrower claim; it tests a mechanistically different correction strategy that their falsification map does not cover. Their nonlinear correctors regress the already-compounded final-layer residual from per-neuron features against a fixed k=2 baseline, evaluated leave-networks-out; ours applies small corrections sequentially at every layer, refitting against the chain's own already-corrected rolled-forward state (DAgger-style), so no single correction has to explain 32 layers of compounded error at once. We independently reproduce their central negative result in isolation: `chain_v2.py` shows step-exact Edgeworth corrections, even from true cumulants, make the chained estimator worse (§1c), because the chain's mean and covariance errors mutually compensate and a locally-correct fix breaks that compensation. We read this as the same mechanism seen from the plain-chain frame; trajectory-fitting sidesteps it. Their exact-moment Edgeworth test (§6 of their PDF) also used keenan-pepper's atlas and found the same null we did, independent convergent evidence that exact marginal moments alone do not fix a propagated estimate. Their narrower finding that nonlinear correctors add nothing over linear ones is confirmed in our setting as well (quadratic heads tied with linear at matched data, §3); in both lines of work the binding ingredient is the fitting protocol, not model capacity. We would welcome a cross-check of trajectory-fitting against their falsification suite.

evaaaz's write-up [10] independently locates the sampling-side frontier at irreducible ReLU-kink variance; our sampling-family experiments (rotation, 1D control variates, per-net adaptivity, all null) are consistent with theirs, and we adopt their framing of where unbiased-sampling improvements must come from.

radiant-allomancer's write-up [11], "A variance-bias budget for deep white-box mean estimation" (submission #317660, adjusted 4.47e-7; read 2026-07-22; the numbers cited here are their claims from the forum post, not independently re-run by us),

is the closest concurrent work to §§4–5 and converges with them from the opposite architecture. They run a sampling-heavy hybrid (RQMC prefix + moment shrinkage toward an analytic chain + analytic final layer + ridge corrector) and decompose its residual with a two-budget N-scaling fit ($m(N) = b^2 + cN^{-p}$), finding 58–73% of their error is bias; we run an analytics-heavy chain and decompose with seed-resampled probes, finding roughly half our error is variance (§5). These are two different instruments and two architectures at opposite ends of the same bias-variance trade-off, each dominated by the component the other has minimized. Three of their negative results match ours mechanism-for-mechanism: their layerwise bias re-anchoring made results 4–9× worse because "the biased anchor compounds through layers" (the same bias-amplification asymmetry we measure in §5, and the §1b reset result seen from the sampling side); their corrector trained on converged features regressed at deploy time ("noisy features must be used"), the same lesson as our filter-then-refit failure (§5), namely that correction stacks must be trained on the input noise distribution they will see; and their diagonal third-cumulant transport "lost 98% of the cross-cumulants", which our §4 quantifies against ground-truth tensors rather than probes (pair-planes reach 2.13e-6 where the full tensor reaches 8.78e-7). Their stated open problem, better deterministic cumulant propagation with low-rank refinements and per-layer closure corrections, is the direction our §4 compressibility measurements inform: rank-64 suffices at layer 8 but nothing compact does at layer 0, so the rank schedule, not the rank, is the design variable. We also tested the one component of their stack that appeared transferable to our sampling entry, the analytic conditional final layer, and found it negative in our stack (§7): architecture components here are not modular, a caution relevant to readers of both write-ups.

Relation to the organizers' town hall (2026-07-15) [12]. Jacob Hilton framed the challenge's core question as whether white-box estimates can outperform black-box sampling, noted that Phase-1 leaders are mostly QMC variants, and offered one concrete recovery hypothesis: *"as the depth increases the effective rank of the covariance matrix decreases, and so it may be possible to recover the deficit between the two classes of methods."* We can confirm the premise quantitatively on real challenge nets: the pre-activation covariance participation ratio contracts $128 \to 5.2$ (25×) from layer 1 to layer 32 (`research/arc/covariance_pr_depth.py`, 8 nets, computed with the verified analytic chain, zero forward passes). By the scored layer the covariance is effectively rank ~5 of 256. Our §4 compressibility profile is the third-order analog of the same contraction (rank-64 needed at layer 8, trivial by layer 24). We then tested the hypothesis in its most natural form, a subspace-split hybrid (one estimator in the top-r eigenspace of the analytic final-layer covariance, the other in the complement), by measuring both estimators' error fields in that eigenbasis on 50 fresh nets (`research/arc/subspace_split_probe.py`). The result is negative with a structural mechanism: the sampler's error concentrates in the same top subspace as the signal, matching the lambda_i/N theory direction-by-direction, and the analytic chain's error is larger than the sampler's at every eigen-rank (1.3× in the leading direction, ~5-6× per direction in the tail), so no split at any rank or orientation beats pure sampling (best 1.08×). The tail that the rank contraction would hand to an analytic method carries only ~1.7% of the sampler's total error, so even a perfect analytic complement is capped at a ~2% improvement. The rank contraction is real, but it discounts sampling noise by exactly the factor it compresses analytic state, leaving no niche for

the split architecture; a method exploiting the hypothesis must instead either beat lambda_i/N in the leading directions or convert the low rank into a FLOP reduction. Two further cautions from our negatives: the naive form of a low-rank correction (probe-reconstructed covariance-eigenbasis terms, §3) is measurably harmful without a closure-derived grounding, and per §5 the binding constraint of our probe-fed analytic track is cumulant sampling noise, which low-rank state compression does not by itself address. Within the analytic track, the trajectory-fitted stabilizer is itself a mechanistic analysis that materially improves the score (~8.4× over the published kprop baseline at matched budget); the error budget in §5 states what still separates it from the sampling frontier, and the rank measurements here indicate where a Phase-2 attempt to close that gap should allocate its parameters.

7. Additional negative results

- QMC/scrambled-Sobol probes for cumulant estimation: worse than plain MC (above).
- Active-subspace input rotation for the sampler: aligned rotations 1.05–1.09× worse than identity (the unrotated coordinate system preserves per-dimension stratification; random rotations are 1.12–1.21× worse). The linearized chain's top singular values are barely separated at width 256 (s2/s1 ≈ 0.84), so there is no low-dimensional input subspace to align to.
- Per-net adaptive estimator selection: excluded in principle for scramble-luck reasons. Per-net QMC convergence rate has cross-scramble correlation ≈ −0.05; "hard nets" are (net, scramble) pairings, not net properties.
- Subspace-split hybrid built on the depth rank-contraction (§6): measured on 50 fresh nets in the analytic final-layer covariance eigenbasis, the deployed sampler's error matches the lambda_i/N profile and the deployed chain's error exceeds it at every eigen-rank, so every split orientation at every rank is worse than pure sampling (best 1.08×), and the complement subspace carries only ~1.7% of sampling error, capping any such hybrid at ~2% even with a perfect analytic side (`research/arc/subspace_split_probe.py`).
- Rao-Blackwellized (analytic conditional) final layer for the Sobol N=8192 sampling estimator, prompted by radiant-allomancer's stack (§6): replacing the final-layer sample mean of relu(z) with a rectified-Gaussian closure of the pre-activation's sample moments is 1.35× worse (paired on identical samples, 100 fresh nets, 1/100 wins); adding sample-k3 and k4 Edgeworth terms recovers most of that (1.007×) but remains a statistically significant slight loss (paired Δ +2.5e-8 ± 0.5e-8 SEM, 27/100 wins; `research/arc/conditional_final_layer_test.py`). At this N the sample mean's final-layer kink variance is already below the analytic closure's residual non-Gaussianity bias, and QMC point sets estimate the higher central moments the closure needs worse than iid MC (the same Sobol-probe negative as above). A component that pays in their RQMC-prefix + moment-shrinkage stack does not pay when dropped into ours.

8. Discussion: toward non-random networks

The organizers frame the random-network challenge as the tractable version of a harder goal: estimating properties of trained models in settings where sampling cannot be trusted, either because the behavior of interest is rare or because the model's behavior under evaluation differs from its behavior in deployment. It is worth stating which of this write-up's findings we expect to transfer to that setting, which we do not, and what the random-network machinery becomes when the weights are no longer random.

Expected to transfer. The error-compensation property (§1, §5) is a consequence of deep recursive moment estimation, not of weight randomness: whenever approximate per-layer closures are chained, their errors correlate through the propagated state, and the arguments for bias amplification and noise cancellation do not invoke the weight distribution. We therefore expect, though have not measured, that correction schemes for trained networks will also need to be trajectory-fitted rather than step-local, and that they will inherit the same intolerance of coherent bias (the ~16:1 asymmetry of §5, in some form). The probe-noise error budget (§5) transfers directly as an instrument: the seed-resampled decomposition applies to any probe-fed estimator on any network, and we would recommend it as a standard check before attributing residual error to model structure.

Not expected to transfer. The closure itself, the rank-contraction profile ($128 \to 5.2$, §6), and the K3 compressibility schedule (§4) are mean-field results that depend on the Gaussian weight ensemble. Trained weights violate those premises by construction, and the depth profiles should be re-measured, not assumed.

The role reversal. For trained networks the random-weight chain acquires a second use: it is the natural null model. Structure in a trained network can be defined as deviation from the analytically predicted random baseline computed from the architecture alone, with zero forward passes. Preliminary measurements in this repository indicate the need for such a baseline is not hypothetical: random-initialization MLPs fed pure noise already exhibit 24-31 "significant" dimensions per layer under a Marchenko-Pastur [13] test (`research/arc/mp_null_check.py`), i.e. architecture-induced anisotropy that a naive spectral census miscounts as structure, and in transformers the raw significant-dimension count of a random-initialization control can exceed the trained model's (`research/arc/reinit_census.py`: gpt2 reinit 111 vs trained 24 mean significant dimensions), while supervised concept-coverage metrics separate trained from random cleanly (roughly 35% vs 2% after outlier-robustness checks, `research/arc/census_noise_budget.py`). Spectral counts alone do not discriminate learning; deviation from an analytic baseline is the better-posed quantity, and the stabilized chain developed here is a candidate implementation of that baseline for the MLP components of larger models.

Open question. Trajectory fitting requires a population of networks with shared statistics to fit on. Random nets supply this population trivially; for trained networks the candidates (checkpoint families, architecture families, fine-tune populations) each change what the fitted correction means. We consider this the main obstruction to transferring the §2 method as-is, and an appropriate subject for Phase 2 alongside the rank-scheduled carriers discussed in §6.

9. Artifacts

Code availability. Every script and results file referenced in this document is published as a replication package at `github.com/jamesrahenry/arc-whitebox-replication` (public as of this document's posting; MIT). It includes the graded submission file itself (`estimator_chain.py`) and a provenance record with the pre-registration evidence for §4. Our separate sampling-family estimator is excluded during the competition and available to the organizers on request; the full working-repository history likewise.

- `estimator_chain.py` — the graded single-file fnp submission: #314695, adjusted 9.97e-7, raw 5.89e-6, multiplier 0.169 (all three grader-reported and independently rounded; they need not multiply exactly; grader, 2026-07-03). Consistent with the seed-resampled held-out expectation (7.45e-6 on nets 40-49, 8.13e-6 on 56 fresh nets; see §2's seed-sensitivity paragraph; the graded raw sits between the fixed-seed and resampled figures).
- `repricing_erratum_check.py` — the erratum's re-metering of `estimator_chain.py` under FLOPScope v0.10.0: float64 bill 7.4031e10 vs 3.7040e10 for a float32 port (ratio 1.999), output delta ≤ 2.7e-6 (raw MSE unaffected).
- `chain_port_refit.py` — portable-step chain + stabilizer fitting (reproduces the 5.2e-6 fixed-development-seed value; seed-resampled expectation 7.45e-6, §2).
- `chain_port_refit_freshval.py` — eval-only check of the frozen coefficients against atlas nets never touched by any design iteration (reproduces the 8.13e-6 fresh-validation number on 56 nets, §2; `uv run python chain_port_refit_freshval.py 50 56`).
- `kprop_depth32.py`, `kprop_drift.py`, `chain_reset.py`, `edgeworth_ceiling.py`, `chain_v2.py`, `k2_step_v2.py`, `chain_v3/v4/v5_stabilizer.py` — the diagnosis series.
- `chain_v6_nonlinear.py` — quadratic-vs-linear correction head comparison (reproduces the paired-difference numbers in §3; `uv run python chain_v6_nonlinear.py 96 10 4096 lin/quad`).
- §4 (ground-truth tensor tests): `FULL_K3_FOLLOWUP.md` (policy + full ledger, including the two mid-session bug fixes and superseded readouts), `full_k3_carrier_eval.py`, `full_k3_combined_source_gate.py`, `combined_carrier_selfprop_check.py`, `estimator_k3_substitution_test.py`, `estimator_baseline_gate.py`, with committed `*_results.json` for each.
- §6 (town-hall relation): `research/arc/covariance_pr_depth.py` — covariance participation-ratio depth profile (128 → 5.2 over 32 layers, 8 nets); `research/arc/subspace_split_probe.py` (+ results) — the subspace-split hybrid test.
- §8 (null-model measurements): `research/arc/mp_null_check.py`, `research/arc/reinit_censu` (+ `reinit_census_results/`), `research/arc/census_noise_budget.py` (+ results; includes the outlier-robustness check behind the corrected coverage figure).
- §5 (probe-noise decomposition): `research/arc/probe_noise_share.py`, `probe_noise_attribut` (bit-exact injectable replica), `probe_denoise_ceiling.py`, `probe_wiener_filter.py`, and `moment_chain/chain_refit_filtered_probe.py` (three-variant refit; control reproduces the 5.2e-6 anchor), with committed `*_results.json` for each.
- Data: public challenge dataset + keenanpepper's public moment atlas

(keenanpepper/arc-whestbench-higher-moments-2026) and ground-truth tensors (keenanpepper/whest-k3-tensors-2026; IDs 800-809, used evaluation-only under the locked-set policy in FULL_K3_FOLLOWUP.md). ARC's kprop code (alignment-research-center/mlp_cumulant_propagation) used as the reference chain during diagnosis; the submitted estimator is self-contained.

References

[1] Alignment Research Center (Wu et al.), the challenge's cumulant-propagation paper, arXiv:2605.05179 (2026). Reference implementation: github.com/alignment-research-center/mlp_cumulant_propagation.

[2] keenanpepper, *whest-k3-tensors-2026*: full post-ReLU third-central-moment tensors for challenge IDs 800-809, two independent 1e8-sample MC streams per net. Hugging Face dataset keenanpepper/whest-k3-tensors-2026 (published 2026-07).

[3] S. Ross, G. Gordon, D. Bagnell. "A Reduction of Imitation Learning and Structured Prediction to No-Regret Online Learning." AISTATS 2011.

[4] S. Bengio, O. Vinyals, N. Jaitly, N. Shazeer. "Scheduled Sampling for Sequence Prediction with Recurrent Neural Networks." NeurIPS 2015.

[5] R. K. Mehra. "On the Identification of Variances and Adaptive Kalman Filtering." IEEE Transactions on Automatic Control, 1970.

[6] Y. Cho, L. K. Saul. "Kernel Methods for Deep Learning." NeurIPS 2009 (the arccosine kernel / Gaussian ReLU moment closure).

[7] B. Poole, S. Lahiri, M. Raghu, J. Sohl-Dickstein, S. Ganguli. "Exponential Expressivity in Deep Neural Networks Through Transient Chaos." NeurIPS 2016 (mean-field covariance propagation).

[8] keenanpepper, *arc-whestbench-higher-moments-2026*: per-layer moment atlas for the public challenge nets. Hugging Face dataset keenanpepper/arc-whestbench-higher-moments-2026 (published 2026-06).

[9] pscamillo. "Characterizing a systematic scale bias in Gaussian-closure activation estimation." AIcrowd forum, posted 2026-07-06, read in full 2026-07-09. discourse.aicrowd.com/t/18063.

[10] evaaaz. Sampling-frontier write-up. AIcrowd forum topic 18053, ~2026-07-02. discourse.aicrowd.com/t/18053.

[11] radiant-allomancer. "A variance-bias budget for deep white-box mean estimation." AIcrowd forum, read 2026-07-22. discourse.aicrowd.com/t/18085.

[12] ARC White Box Estimation Challenge Town Hall (J. Hilton, P. Christiano), 2026-07-15. youtu.be/pvCgbUJ-nTo.

[13] V. A. Marchenko, L. A. Pastur. "Distribution of Eigenvalues for Some Sets of Random Matrices." Matematicheskii Sbornik, 1967.

Acknowledgments

This work depended on artifacts and write-ups that other participants published openly during the competition, before this document. keenanpepper published the two datasets without which most of it could not exist: the per-layer moment atlas (`keenanpepper/arc-whestbench-higher-moments-2026`), which supplies the training and validation targets for the §2 stabilizer, and the full third-cumulant tensors (`keenanpepper/whest-k3-tensors-2026`), which made the §4 ground-truth tests possible at all. Publishing artifacts of that size and quality mid-competition is a substantial service to every participant, and we have tried to repay it with careful evaluation-only handling (§4) and full attribution at each point of use. We also read and engaged with the prior public write-ups of pscamillo (posted 2026-07-06) and evaaaz (~2026-07-02), and the concurrent write-up of radiant-allomancer (2026-07-22), all before finalizing this document; §6 details what we take from each, where our results corroborate theirs, and where we test their components in our setting. The ARC organizers' published kprop reference implementation and the 2026-07-15 town hall shaped §1 and §6 respectively.

Disclosure and provenance

This work was AI-assisted (Claude, Anthropic). The model proposed experiments, wrote the code, and drafted this document; the author directed the research and reviewed the results.

Verification protocol. Before the original submission, the write-up and its supporting code and logs were reviewed end-to-end by a separate model instance with no prior context, tasked as an adversarial reviewer. It cross-checked every number against the scripts that produced it and flagged unsourced claims, small-sample results reported without spread, and one place where a competitor's hedged claim had been paraphrased more strongly than they stated it (fixed; see §6). Every quantitative claim traces to a named script and a logged run. Three exceptions, marked at point of use: ARC's claim that kprop degrades with depth (Summary; taken from their paper); evaaaz's characterization of the sampling-side frontier (§6; a competitor's claim consistent with our own null results, not independently proved); and the sampling family's scoring-replica figures (raw 1.61e-6, adjusted ≈4.5e-7 expectation; measured 2026-07-02 on the full 1,000-net public split with the starterkit local evaluation engine and logged in the working repository, but the run script was not preserved as a committed named artifact, so this number does not meet the tracing standard the rest of the document holds itself to).

Second session and second review. Sections 4–5, the radiant-allomancer and town-hall material in §6, and the final §7 item were added in a second session (2026-07-17 → 2026-07-24); every number in them traces to a named script and committed results file. The whole document was then re-reviewed end-to-end on 2026-07-23 by a separate model instance with no prior context, which recomputed the new sections' numbers from the committed results files (all reproduced) and surfaced: the seed-sensitivity issue now analyzed in §2 (the original 5.2e-6 headline was a favorable fixed-seed draw; the seed-resampled expectation is 7.45e-6), one previously-uncommitted supporting experiment (now committed: `research/arc/antithetic_probe_toy.py`),

and the training-overlap caveats now stated inline in §5.

Revision history. 2026-07-03: original submission draft (§§1-3, 6-7 in their then-numbering). 2026-07-22: §§4-5 added (ground-truth joint-cumulant tests; probe-noise decomposition); §6 extended with the radiant-allomancer comparison; §7 extended with the conditional-final-layer transfer negative. 2026-07-23: second adversarial review; seed-sensitivity corrections in §2 and the Summary; referenced-submission summary added. 2026-07-24: §6 town-hall relation with the covariance-rank measurement and the subspace-split negative; §8 discussion of non-random networks; editorial pass for register; acknowledgments, sampling-expectation corrections, and code availability added. 2026-08-04: erratum for the FLOPScope v0.10.0 cost-model repricing (all grader-reported multiplier/adjusted figures regraded in place; raw MSEs and all claimed contributions unaffected; body text left as originally graded, corrections stated in the erratum at the top; supporting measurement in `repricing_erratum_check.py`).

Bugs found and disclosed. Two bookkeeping bugs in our own evaluation gates (a missing per-neuron σ factor, and post-ReLU statistics fed where pre-ReLU inputs were required, both in `analytic_k3_source_gate.py`) were caught mid-session: the first by an independent Gauss-Hermite quadrature cross-check, the second because transmitted origins decorrelated anomalously with age, confirmed by a Gaussian-closure-mean sanity check. Both were fixed before any number they could affect was recorded. The full discovery trail, including the superseded intermediate readouts, is preserved in `FULL_K3_FOLLOWUP.md` and the repository history.