Characterizing a systematic scale bias in Gaussian-closure activation estimation

ARC White-Box Estimation Challenge 2026 — technical writeup This write-up corresponds to **submission #314331** (participant: pscamillo), graded on the **Phase 1** evaluator (adjusted score 2.45×10^{-6}, “Graded successfully”, 50/50 public MLPs scored). The approach that submission implements is the scalar-corrected k=2 estimator described below (§2); the falsification experiments (§6) are the ablations and negative results supporting it.

Summary

We characterize a systematic **multiplicative scale bias** in the second-order Gaussian-closure estimator (covariance propagation, “k=2”) for the per-neuron final-layer activation mean of random ReLU MLPs. On the official benchmark the estimator overestimates the final-layer mean by a small, remarkably stable factor: the optimal correction is **0.9916 ± 0.0027** across 100 networks and **0.9921 ± 0.0001** under cross-validation, generalizing to held-out networks. A single scalar multiply — zero additional FLOP cost — reduces final-layer MSE by **~3×** over the covariance baseline. Directly against the organizers’ own reference cumulant-propagation code, this zero-cost correction *matches or beats* full third-cumulant (k=3) propagation at the benchmark’s depth (L=32), at 1/500th the cost.

We further show, by direct experiment, that this scale bias is only **one component** of the k=2 error. After removing it, the residual has no structure recoverable by the cheap corrections we tested — polynomial, feature-based, or low-rank-linear (PCA effective rank ~29 of 256) — and its magnitude scales with activation variance, behavior consistent with an intrinsically higher-order non-Gaussian component. We also localize the bias mechanistically: the optimal factor is 1.000 at the input and accumulates monotonically with depth to ~0.992, so the bias is a property of the *propagation*, not of the final layer. The central

contribution is this **decomposition and localization** of the closure error, together with a map of the alternative approaches we falsified.

Honest positioning. The corrected estimator scores adjusted final-layer 2.45e-6 on the public split — a ~3× improvement over the covariance baseline, but it does *not* beat the challenge's Monte-Carlo reference (6.5e-7) and is well behind the score leaders. The contribution is the *mechanistic characterization and decomposition of the error*, not a competitive score. We believe the finding — that a large, clean, zero-cost-correctable component of the k=2 error exists, and that the remainder is structurally non-Gaussian — is a useful input to the estimator design the challenge is trying to advance.

---

1. Setup and notation

The task: given weights of a random He-initialized ReLU MLP (width n=256, depth L=32, init variance 2/n), estimate per-neuron E[h(X)] under X~N(0,I), scored by final-layer MSE against a high-precision Monte-Carlo reference, within an analytical FLOP budget.

The k=2 Gaussian closure propagates a mean vector μ and covariance Σ layer by layer, applying exact per-neuron post-ReLU moments under the Gaussian assumption: for pre-activation $Z \sim N(m,s^2)$, $E[\mathrm{ReLU}(Z)] = m\cdot\Phi(m/s) + s\cdot\varphi(m/s)$. This is the standard covariance-propagation baseline.

2. The scale bias

Discovery. We regressed the k=2 error (truth − k2) on cheap features the estimator already computes. A 15-feature model reduced the error, but under ablation it collapsed entirely to a single scalar: the k=2 estimate itself. In other words, the error is proportional to the estimate — a multiplicative bias. Fitting truth ≈ c·μ_k2 gives c ≈ 0.992, and no nonlinear f(μ) (quadratic, cubic) improves on the scalar.

Official-benchmark validation. On the official public dataset (100 networks, `v1-phase1`, 1e9-sample MC truth), the per-network optimal factor is **0.9916 ± 0.0027** (min 0.9858, max 1.0010). The running mean converges rapidly with no drift (0.9911 at 10 networks → 0.9916 at 100). Applying the factor reduces final-layer MSE from 8.37e-5 to 2.59e-5. (Figure 2: the per-network factors concentrate tightly just below 1.0, cleanly separated from the no-bias value 1.0.)

Predictive, not fine-tuned. This is the key point for interpreting the constant. Under 5-fold cross-validation by network — the factor estimated on 80 networks and applied *without readjustment* to 20 held-out ones — the trained factor is **0.9921 ± 0.0001** and the held-out MSE reduction is **~3×** (2.73–3.64 across folds). Leave-one-out gives 0.9921 ± 0.0000 (range 0.9920–0.9922). Independent training subsets recover the same correction to within 0.01%, and it transfers to networks that never entered its estimation. On this benchmark distribution the factor is a measurable, stable property of the estimator, not a value tuned to the benchmark.

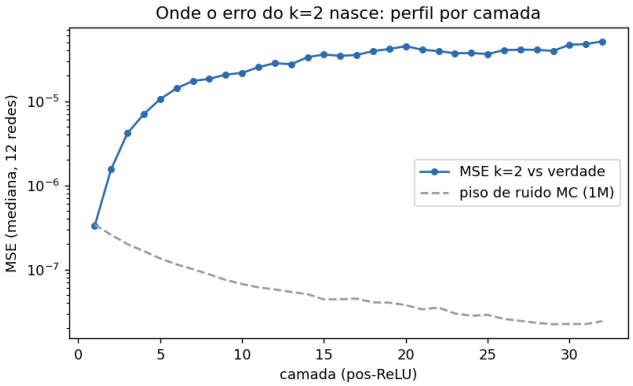
In the grader's protocol. Through the official evaluation harness, the corrected estimator scores adjusted final-layer 2.66e-6 vs 7.52e-6 for the uncorrected covariance baseline on the same networks (2.83× improvement), at identical FLOP cost. The graded submission (#314331) scored 2.45e-6 on the public split with 0/50 failures.

3. Mechanism and architecture dependence

Mechanism — what we ruled out, and what we then demonstrated. We first tested the natural hypothesis that the bias is a skewness-truncation effect, and the data contradict it. Measuring the actual moments of the final pre-activations (MC, 200k samples): the standardized skewness is ~0 on average (+0.005, symmetric — 47% of neurons negative, 53% positive), so a skewness effect cannot produce a *systematic* bias. The excess kurtosis is systematically positive (+0.40, leptokurtic in 99% of neurons). But a first-order Edgeworth expansion on the final layer does *not* reproduce the ~0.008 bias from either moment: the final activations sit at large α = m/s (~3.2 on average), well away from the ReLU knee, where ReLU is nearly linear and the mean is insensitive to higher moments — the magnitude-weighted kurtosis correction comes out ≈1.000, not 0.992. This rules out simple first-order moment corrections *at the final layer*.

We then measured the bias layer by layer, which settles the question directly. The optimal per-layer factor is **1.0000 at layer 1** (no bias, as expected for a Gaussian input) and **decreases monotonically with depth, saturating near 0.992** by the final layer (Figure 6). The k=2 error follows the same profile: it grows over the first ~8–10 layers and then plateaus (Figure 5). The bias

is therefore a phenomenon *accumulated during propagation*, not an effect of the final layer — the accumulation is observed, not inferred.



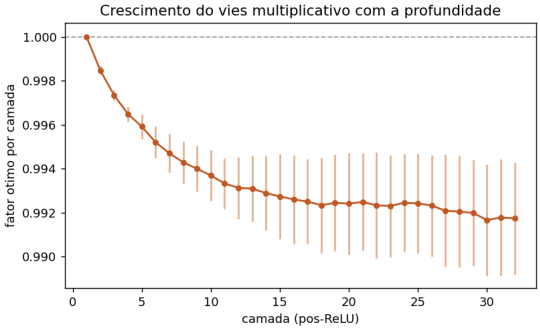


Figure 6. Optimal per-layer factor starts at 1.000 and accumulates to ~0.992 with depth.

The excess kurtosis we measured also accumulates monotonically with depth (0 → 0.40), consistent with the same picture. What we do not have is a closed-form derivation of the saturation value 0.992 from the propagation recurrence; that remains future work, and the factor is empirically established rather than derived.

Architecture dependence (local experiments). Varying init, width, and depth in a local generator: the factor is identical for He and Xavier initialization (0.9928 both) — it does not depend on weight scale. It varies systematically with width and depth: the bias grows sublinearly with depth and decreases with width, qualitatively consistent with the (L/n)^K error scaling conjectured in the companion paper. We deliberately do **not** report precise power-law exponents: fits shift materially between point sets, so the data support only the qualitative form.

4. Decomposition of the error: what the correction leaves behind

Removing the dominant scale component lets us study a much smaller, cleaner residual r = truth − 0.992·μ_k2. We asked whether *r* has exploitable structure.

- **Per-neuron (5-fold CV):** regressing r on non-μ features the estimator already computes (pre-activation variance, α = m/s, ReLU-knee proximity, weight-column norm) reduces the residual by **1.00×** — nothing.
- **Per-network:** the residual *energy* correlates with mean pre-activation variance (corr **+0.53**) but not with weight norms. More precisely, binning neurons by activation variance, the residual standard deviation scales cleanly and near-linearly with the activation standard deviation: std(r) ≈ 0.011·σ(z), linear fit **R^2 = 0.94** (Figure 4). Since variance was among the per-neuron features that gave 1.00× above, this is a relationship with the residual's *magnitude* (heteroscedasticity), not a subtractable mean — so it characterizes where the remaining error concentrates but does not yield a further correction.

This yields a clean decomposition of the k=2 final-layer error:

1. **A multiplicative scale bias** — stable (~0.992), predictive, removable at zero cost. Captures the dominant, structured part of the error.
2. **A residual, higher-order component** — its magnitude scales with activation variance (larger variance → more probability mass outside the Gaussian-accurate regime → more error), and it is *not* removable by any cheap function of the k=2 statistics. This is consistent with an intrinsically higher-order (k≥4) non-Gaussian term, though our experiments establish only that it is not cheaply correctable from k=2 quantities, not its precise origin.

We also asked whether the residual is *low-rank* — whether a few shared directions capture it, which would license a cheap subspace correction. A PCA of the residual across 60 networks says no: the top principal component explains only 11.5% of the energy, the top five explain 29%, and 46 components are needed for 95% — an effective rank near 29 (of 256). Combined with the near-zero cross-neuron and cross-network correlations, this rules out a low-dimensional *linear* structure. We are careful not to overclaim: this shows the residual is not reducible by the (polynomial, feature-based, low-rank-linear) corrections we tested, not that no structure of any kind exists — a nonlinear or alternate-basis representation is not excluded, and we leave that open.

The correctable part and the residual part are thus separated, each characterized. This is the natural stopping point for a low-cost line of attack in the cumulant family.

5. The scale correction matches full third-cumulant propagation at this depth

A natural objection is that the "right" fix is simply to propagate more cumulants. We tested this directly using the reference implementation of the challenge organizers' own cumulant-propagation algorithm (the `mlp_kprop` package accompanying their paper), running it at k_max=3 (mean + covariance + full third cumulant) on the challenge's own architecture (n=256, depth 32), against 1e6-sample MC truth over 12 networks.

The result is striking: **third-cumulant propagation does not beat the zero-cost scalar correction at this depth.** The reference k=3 estimator costs ~78 s per network (≈500× the k=2 cost) yet its output-mean MSE (mean 4.6e-5) is *not* lower than k=2 × 0.9921 (mean 3.3e-5, noise floor 6e-8); the scalar correction wins on 8 of 12 networks. The gain of k=3 over *uncorrected* k=2 is essentially the same scale bias our factor already removes for free. This is consistent with the depth story: the paper's own analysis notes the method's dependence on depth is unfavorable, and at L=32 the dominant error is of higher order than the third cumulant reaches, while the Hermite approximation of the ReLU (which the paper flags as a practical weakness) introduces its own error. The practical consequence for this benchmark: a single scalar multiply captures essentially all the accuracy that third-order cumulant propagation delivers at this depth, at 1/500th the cost.

6. Cheap-first search for a better correction: what we tried and did not find

Beyond the cumulant family, we ran a series of cheap, falsifiable experiments looking for *any* structure a low-cost correction could exploit. We report them as negative results, carefully bounded to what was measured — none of these licenses the claim that no correction can exist, only that the ones we tried did not help.

- **Temporal recurrence of the error.** We tested whether the per-layer error vector e_l = μ_l^MC − μ_l^k2 obeys a cheap recurrence e_{l+1} = F(e_l, μ_l, var_l) using only k=2 quantities. Out-of-sample R^2 = 0.003; corr(e_l, e_{l+1}) =

0.037. We found **no simple linear recurrence in the observable per-neuron error** — which does not rule out a hidden-state dynamics whose observable projection decorrelates, only that the direct linear one is absent.

- **Direction vs. magnitude.** A sharp split emerges: the error *direction* is unpredictable (the recurrence above), but the error *energy* $||e_{l+1}||^2$ is highly predictable from $||e_l||^2$ and depth alone (out-of-sample $R^2 = 0.95$). The energy also grows near-linearly with depth (power-law exponent ≈ 1.05). So the process appears to preserve global magnitude statistics while scrambling direction — a clean empirical characterization, though we do not claim a generative "white-noise" mechanism from it.
- **Exact higher moments (ground-truth).** Using an independent public dataset of ground-truth 2nd–4th moments at 1e8 samples/MLP (keenanpepper/arc-whestbench- higher-moments-2026), we applied a first-order Edgeworth correction built from the *exact* pre-activation skewness and kurtosis. It did not improve the post-ReLU mean at depth — in our implementation it degraded it (gain 0.4–0.5×). We read this narrowly: *this Edgeworth approximation, even with exact moments, does not help,* consistent with the final activations sitting far from the ReLU knee. It does **not** establish that the mean is structurally insensitive to higher moments, nor that no moment-based method could help — an asymptotically ill-conditioned expansion can fail to exploit information that is nonetheless present.
- **Learned nonlinear corrector (predictability ceiling).** We measured how much of the residual any cheap model can predict, using tree ensembles (random forest, gradient boosting) with features μ, var, α, exact skewness and kurtosis, out-of-sample by network. Pooled across all layers, nonlinear models recover a modest signal the linear fit misses ($R^2 \approx 0.18$ vs 0.04). Resolving this by depth (Figure 7, 5 splits, ±1 s.d.) tells a specific story: the predictive signal *available to this family of inexpensive predictors* peaks in the early layers ($R^2 \approx 0.94$ at layer 2, where activations are near the ReLU knee), decays smoothly, and is statistically indistinguishable from zero in the final layers (layer 31: -0.01 ± 0.04). We phrase this as a statement about what these predictors can extract, not about the true distribution — the residual may carry structure no cheap predictor on these features captures. On the *final layer* — the scored quantity, where $\alpha \approx 3$ — a gradient-boosting corrector trained leave-networks-out does not reduce the residual MSE; applied out-of-sample it slightly worsens it (0.95×). We audited this negative: training on *shuffled* labels gives the same test R^2 as training on real labels (both ≈ -0.06), the direct signature of no generalizable signal for these predictors; the model memorizes 23% of the training set (R^2=0.23 in-sample) but none transfers. Ridge, random forest, and gradient boosting all give test $R^2 \approx 0$. (The pooled $R^2 \approx 0.18$ and the per-layer peak $R^2 \approx 0.94$ use the same feature set and

R^2 definition; they differ because one pools all layers and the other isolates the best layer — see appendix.) The decay is regular and holds per network, not just on average: over layers 2–14 (the fit is restricted to this range because beyond ~layer 15 the predictability is statistically indistinguishable from zero), an exponential model $R^2(d) \approx A \cdot \exp(-(d-2)/\tau)$ fits every one of the 40 networks with $R^2 > 0.8$ (median per-network fit $R^2=0.96$), with characteristic scale $\tau = 5.1$ layers (bootstrap 95% CI [4.6, 5.8]). It substantially outperforms a power law (0.88 vs 0.997 on the mean curve). We emphasize this is an empirical characterization of the observed predictor performance, not a derived law of the underlying dynamics — τ describes how fast this family of predictors loses traction with depth. One possible interpretation is that this scale reflects the progressive loss of cheaply exploitable information as activations move away from the ReLU transition region, but we did not directly test that mechanism (we did not measure whether α, kurtosis, or knee-distance follow the same scale).

Independent convergence. A separate Phase-1 approach using randomly-shifted QMC plus a Rao-Blackwellized exact first layer reached the same qualitative frontier from an orthogonal direction, reporting that the residual is “irreducible kink variance” that control variates and analytic correctors do not close at depth. Two methodologically independent lines arriving at the same boundary strengthens the reading that, *within the correction families explored*, the low-cost frontier for the post-ReLU mean of a deep random MLP is close to what the scalar-corrected k=2 estimator already achieves.

Summary of this section, stated conservatively: among the inexpensive correction families investigated in this work — third cumulant, low-rank subspace, temporal recurrence, an exact-moment Edgeworth correction, and linear/nonlinear regressors trained on inexpensive statistics — we found no method that improved the final-layer estimator under leave-networks-out evaluation. Substantial predictive signal exists in early layers but decays exponentially with depth ($\tau \approx 5.1$ layers, 95% CI [4.6, 5.8], per-network) and is statistically indistinguishable from zero in the final layers, for the predictor families evaluated. We do not claim no correction can exist; we claim the accessible-and-cheap ones we tried do not, and we audited the strongest negative against shuffled-label and overfit controls.

Relation to the matching-sampling question. The challenge’s motivating conjecture is the *matching sampling principle*: whether cheap mechanistic estimation can match sampling. Our results speak to it empirically for depth-32 ReLU MLPs. The leaderboard’s strongest entries are variance-reduced samplers (randomized QMC, Rao-Blackwellized exact layers) operating at several times the efficiency of plain Monte Carlo; a public writeup by one competitor reports the residual there is “irreducible kink

variance” that control variates do not close. Independently, from the mechanistic side, we find that cheap corrections to the Gaussian closure do not reach that regime at this depth, and that the information a cheap corrector could exploit decays exponentially with depth, vanishing by the scored final layer. Two orthogonal lines — optimized sampling and cheap mechanistic estimation — thus meet at the same boundary. This is evidence that, for the post-ReLU mean of deep random MLPs, the cheap-mechanistic side of the matching-sampling principle is hard precisely where depth is large, and it localizes *why*: the exploitable non-Gaussian signal is a high-dimensional, depth-attenuated quantity rather than a low-order correction.

7. Falsification map: approaches we ruled out

A significant part of this work is negative results, each with a mechanism and a destructive test. These locate *why* the obvious cost-reduction levers fail and where the non-Gaussian signal lives.

Approach	Verdict	Why it failed
Exact bivariate ReLU moment	Not the bottleneck	Ties/worse at 6.5× cost; the closure, not the ReLU step, is the limit
CP-rank cap of degree-3 cumulant	Regime artifact	“Gain” from capping is regularization against depth-accumulated error, not compression
Subspace projection of degree-3	Real but infeasible	Signal subspace is emergent (not from W’s SVD) and rotates fast; tracking costs 1.5× more than full k=3
Hybrid MC + control variate	Insufficient	Per-sample variance reduction only ~1.1×; the ReLU knee is not capturable by low-order moments
Adaptive k=3 scheduler	Not robust	Optimal stopping layer varies across seeds; a fixed schedule fails on unseen networks

Approach	Verdict	Why it failed
Multi-feature ML corrector	Collapses to a scalar	15 features reduce to the single scale factor of §2

A recurring lesson: several levers were killed by a “circularity” — to know where the non-Gaussian signal lives, one must already have propagated it. The scale-bias correction avoids this: it is computed from quantities the k=2 estimator already produces.

8. Strategic observation

The Phase-1 score leaders sit near adjusted 1.7e-7 using ~10% of the FLOP budget (the compute multiplier at its floor). This indicates the binding constraint at the top is **accuracy, not compute** — a qualitatively different idea than cost-optimized cumulant propagation. Our decomposition is consistent with this: the scale bias is a large but *correctable-for-free* component, and closing the remaining gap requires attacking the non-Gaussian residual (§4), which our experiments indicate needs higher-order moments.

9. Reproducibility

All results derive from the official public dataset (`aicrowd/arc-whestbench-public-2026`, revision `v1-phase1`, mini split) with its 1e9-sample MC truth. The estimator is the covariance baseline with the final-layer mean multiplied by 0.9921; it uses only `flopscope.numpy` and adds one scalar multiply. Validation scripts (official-set factor estimation, 5-fold and leave-one-out cross-validation, figure generation, residual gate) accompany this writeup. The submission does not modify or hack flopscope.

10. Transparency on method and LLM usage

This work was carried out as a human-directed, LLM-assisted investigation, in an explicit falsify-before-claiming loop: hypotheses were proposed, tested with destructive experiments, and only asserted after validation. Substantial code (estimator, validation scripts, figures) and portions of this writeup were produced with an AI assistant (Anthropic’s Claude), used in a loop with a second LLM acting as an adversarial critic that repeatedly caught overclaims before they became conclusions. We have tried to be careful to separate **fact** (measured) from **interpretation** (plausible but not proven), and to label the latter as such

throughout — for example, we initially hypothesized a skewness-truncation mechanism and then falsified it against measured moments (§3), reporting the negative result rather than retaining a convenient story. Where a quantity could not be reliably determined from the data (the width/depth power-law exponents), we report only the qualitative form rather than fitted numbers. Two of our strongest results come from running the organizers' own reference implementation (the `mlp_kprop` package) unmodified — the k=3-vs-scalar comparison (§5) and the per-layer factor profile (§3) — so they do not depend on our own reimplementation being correct. Where we ruled structure out (the residual PCA, §4) we state the specific hypothesis excluded (low-rank linear) rather than claiming no structure exists. Section 6 draws on an independent public dataset of ground-truth higher moments; the adversarial-critic loop specifically caught and corrected an overclaim there — an initial reading that "exact moments do not help, therefore the mean is structurally insensitive to higher moments" was narrowed to "this Edgeworth approximation, even with exact moments, did not help," since a failing approximation does not prove the information is absent. The authors reviewed and validated the experimental claims against the official benchmark; the code is straightforward covariance propagation plus a scalar and does not rely on opaque or unexamined LLM-generated logic.

Appendix: key numbers

Quantity	Value	Source
Per-network optimal factor	0.9916 ± 0.0027	100 official networks, 1e9 truth
Cross-validated factor	0.9921 ± 0.0001	5-fold by network
Leave-one-out factor	0.9921 ± 0.0000	100 official networks
MSE reduction (held-out)	~3×	5-fold CV
Corrected vs baseline (grader)	2.66e-6 vs 7.52e-6	`whest run` official harness
Graded submission score	2.45e-6	#314331, public split
Residual reduction by non-μ features	1.00×	5-fold CV (§4)
Residual std vs activation std	std(r)≈0.011·σ(z), R^2=0.94	100 networks (fig4)
	+0.53	100 networks (§4)

Quantity	Value	Source
Residual-energy corr. with activation variance		
Residual PCA effective rank	~29 of 256 (PC1=11.5%)	60 networks (§4)
Per-layer factor	1.000 (L1) → 0.992 (L32), monotone	12 networks, MC 1e6 (fig6)
k=3 reference vs k=2×0.9921 (L=32)	k=3 does not beat; scalar wins 8/12	12 networks, MC 1e6 (§5)
k=3 reference cost	~78 s/net (≈500× k=2)	mlp_kprop, n=256 L=32 (§5)
Error recurrence e_l→e_{l+1}	R^2=0.003 (no linear recurrence)	20 nets (§6)
Error energy $\|e_{l+1}\|^2$ predictability	R^2=0.95 from $\|e_l\|^2$+depth	16 nets (§6)
Error energy growth with depth	power-law exponent ≈1.05	16 nets (§6)
Edgeworth w/ exact moments (ground-truth)	gain 0.4–0.5× (no improvement)	1e8-sample dataset (§6)
Residual predictability (nonlinear, pooled)	R^2≈0.19 (RF/GBM) vs 0.04 linear	16 nets (§6)
Learned corrector on final layer (metric)	0.95× (does not help)	40 nets, leave-networks-out (§6)
Corrector audit: real vs shuffled labels	both test R^2≈−0.06 (no signal)	40 nets (§6)
Residual predictability by depth	peaks R^2≈0.94 @layer 2 → −0.01±0.04 @layer 31	40 nets, 5 splits (fig7)
Pooled vs per-layer R^2 (same 6 features)	pooled 0.18; per-layer peak 0.94	consistent by construction (§6)
Predictability decay with depth	τ=5.1 layers, 95% CI [4.6,5.8]; per-net fit R^2>0.8 in 40/40	40 nets (§6)