Measuring the Ceiling of a Method Class

How to find out whether a family of estimators can reach your target — before building any member of it

trim_qewas · AIcrowd submission 326725 · ARC White-Box Estimation Challenge 2026, Phase 1

17 August 2026

Abstract

A family of estimators has a ceiling, and you can measure it without building a single member of that family. Take your estimator, replace one internal quantity with its ground-truth value, leave everything else running, and measure. The result is an upper bound on every idea whose effect passes through that quantity — including ideas nobody has written yet. If the bound sits below your target, the family is closed, and no cleverness inside it will help.

One such measurement costs about fifteen minutes. We ran the protocol 54 times against the White-Box estimation task — predicting $\mathtt{E[ReLU(W_{31}{}^{\top}h_{30})]}$ for depth-32 random ReLU networks — and it closed 54 method families, most of which we would otherwise have built. The ones that matter:

```
substitute the exact …        ceiling   what it closes
per-step formula error         ×1.19   γ-schemes, quadrature, learned steps
mean and covariance            ×3.36   the entire two-moment class
covariance alone               ×1.10   every "better covariance" idea
per-network personalisation    ×1.03   all per-instance constants
exactness through 3rd degree   ×3.4    what third-order exactness buys
exactness through ANY degree   ×7–9    EVERY smooth approximation, any price
```

The first row is worth `+3.2%` in score. The second says that a *perfect* two-moment closure lands around 5th place, `×7.7` short of the 2nd-placed entry — so the class we chose does not contain a winning method, and we can say that with a number rather than with a hunch.

Our own submission is not competitive (`1.855e-07` adjusted), and the second row above is the reason: we spent Phase 1 inside a class whose ceiling we eventually measured and found too low. We are reporting the map rather than the estimator, because the map is the part that transfers — and because the same protocol that produced it also caught us being wrong 51 times, which is the other half of what we have to offer.

The report is organised around the protocol (§2), the ceilings it produced (§3), the mechanisms that explain them (§4), the small set of things that actually worked (§5), and what we got wrong (§6) — 51 of our 396 recorded findings are refutations of an earlier result of our own, and §6 sorts them into the six mechanisms that generated them.

Four results are separable from this competition entirely, and are the parts we would most like read by someone working on a different problem.

§3.3 — the residual is not complicated structure, it is the kinks, and the kinks are the medium. The high-degree Hermite tail of the target collapses `3–7×` when ReLU is replaced by `softplus₄`, while the function is still 55–66% nonlinear. And the median distance from a sample to the nearest switching surface is `1.1e-04`, with `99.1%` of the probability mass within `10⁻³` of one — forced by a count of `~8000` hyperplanes and confirmed by a layer-0 control that reproduces `|N(0,1)|` to four digits. So every smooth approximation buys the same first three Hermite degrees and stops at any price, *and* the opposite family — cells, Gegenbauer, poles, sampling aimed at the singularities — is equally dead, because there is nothing to localise. Separately, six theories of approximation reduce to one number: in 256 dimensions only ridge functions integrate exactly, and the ceiling of *any* additive one-dimensional class is `×1.89`, equal to what polynomials already give. **The bottleneck is additivity, not the basis.**

§3.5–3.6 — two statements about the scoring rule rather than about us. Under this rule the score of a pure Monte-Carlo estimator is **independent of the budget** (seal: two runs differing tenfold return `6.741e-07` both times), and a *better* deterministic branch makes the optimal budget floor *stricter*. Empirically, the entire analytic advantage is a small-sample phenomenon: honestly re-fitted, it falls `4.24× → 1.47× → 1.09×` as the sample budget grows `×1 → ×8 → ×64`, and at `×64` the optimal weight on the analytics is exactly zero.

§3.7 and §4.7 — two impossibility results, not difficulty results. The learned-surrogate family is closed at six independent corners, with the counter-intuitive centre that **adding capacity makes the estimate 200× worse** — least squares gives an unbiased mean for free and SGD does not — and an offline-trained hypernetwork fails with a diagnosis: the cheap analytic base is *the same number for every network*, so no cheap descriptor of an instance exists from which its answer could be recovered. Per-instance personalisation is closed by *unobservability*: the optimal constant depends on a linear functional of the realised noise while every statistic computable inside the estimator is quadratic in it, so the correlation is `+0.714` across networks and `−0.830` within one.

§4.6 — twelve structural facts about deep random ReLU networks themselves. 68% of the final error lives in a six-dimensional subspace computable from the weights with no sampling; the activation combinatorics collapses with depth but describes precisely the part that does not matter; the non-Gaussian residual is nearly rank 2 for one network and nearly orthogonal across networks; the closure is exact at infinite width; any "analytic anchor at layer ℓ plus sampling" scheme is capped by one measured profile; the answer arrives in the first forward pass; a homogeneity theorem explains why per-layer scalar corrections are always absorbed; the level statistics of the layer Jacobian sit on the GOE value and stay there across depth, so there is no hidden conserved structure to exploit — tested the way physics tests it; low rank of the values is not low entropy of the gates, and the difference is `2⁶⁵`; and the network performs its

own dimension reduction, which is why plain sampling is so hard to beat. It ends with the advanced machinery we tried and found inapplicable, with the reason in each case.

Companion volume: **The Compendium** — the full registry, in the form *claim* → *number* → *seal* → *verdict*, for all 54 closed classes and all 396 findings.

Notation. Seven labelled remarks recur throughout. They classify a statement rather than decorate it, and the classification is the point: most of what follows is a negative result, and a negative result is only as good as the check behind it.

> **Seal.** — an independent check that a number is what we think it is: a known value reproduced, an exact special case, a scaling exponent, or a control that must break and does. Twenty-eight appear below; the dagger † marks a sealed line inside a table.
>
> **Mechanism.** — why a number is what it is, as opposed to that it is. **Caveat.** — a limit on the claim it is attached to; these sit beside our own results, not only beside other people's. **Rule.** — a working rule adopted afterwards, usually bought with a wrong verdict of ours. **Lesson.** — something understood only after being wrong about it first. **Closed.** — a branch closed, with the ceiling and the pre-registered threshold that closed it. **Warning.** — a failure mode that can silently invalidate a measurement.

Numbers without a seal are marked as derived. Where a measurement contradicted an earlier claim of our own, both appear, and §6.2 lists the ones that changed a conclusion.

Contents

1. The problem, and the regime that governs everything

Given a bias-free He-initialised MLP (width `n = 256`, depth 32, ReLU) and `z ~ N(0,I)`, predict the final layer's mean activation analytically, under a budget metered in FLOPs.

The natural approach propagates the mean `m` and covariance `C` layer by layer, assuming the preactivation `p = Wᵀh` is jointly Gaussian at each step:

```
t = m_p/σ_p ,  ψ(t) = φ(t) + tΦ(t)
E[ReLU(p)]            = σ_p·ψ(t)
E[ReLU(p)²]           = σ_p²·[(1+t²)Φ(t) + tφ(t)]
E[ReLU(p_i)ReLU(p_j)] = Mehler series in ρ_ij
```

Exact at layer 0 — the input is Gaussian by construction — and approximate afterwards, because preactivations are sums of `n` weakly dependent terms and Gaussianity fails at order `1/n`.

The exact object exists, and is the wall. Writing the network as a sum over paths through the activation pattern gives

```
m_i = Σ_paths (Π weights) · P(p₁ > 0, …, p₃₁ > 0)
```

an orthant probability whose covariance is the Gram matrix of the partial products `u_k = W₁…W_k`. This is exact, finite, and useless: `256³¹ ≈ 10⁷⁴` paths, each orthant probability a #P-hard 31-dimensional integral, and for a *fixed* network there is no symmetry to fold them with. **Seal.** An independent count agrees — every one of 4096 samples lands in its own linear region, the largest chamber holding `0.02%` of the mass. Both descriptions of the wall — "the cumulant tower does not close" from the estimator side, and "a sum over exponentially many chambers with no symmetry" from the mathematics side — are the same statement, and we arrived at them separately.

One measured fact explains most of this report. The network *freezes* with depth:

	L0	L1	L8	L16	L24	L31
median |t|	0.003	0.466	1.588	2.191	2.593	3.274
neurons that switch	100%	99.9%	66.5%	49.7%	40.5%	30.9%
frozen = ‖m‖²/(‖m‖²+trC)	–	0.318	0.834	0.882	0.923	0.943
effective rank of Cov(h)	–	165.6	–	9.5	–	2.8

Mixing has an effective-rank half-life of 4.83 layers, which drives `frozen` to `0.94`, which drives `t` to ≈ `3.3`. There, ReLU is nearly linear over the bulk, `E[ReLU(p)] ≈ m_p`, and **the shape of the distribution barely enters the answer.** That is why a hundred attempts at "compute the higher moments more accurately" returned `×1.0x`.

Caveat. Every link in that chain is depth-dependent, and an earlier draft of this report quoted the deep value as if it held everywhere — a factor of 7 error at layer 1. The argument survives, and is strengthened: the backward operator from layer `l` to the output has effective rank `2.74` at `l = 0` and `100.5` at `l = 30` (§4.2), so shallow layers barely reach the answer at all. The regime that governs the result is the deep, frozen one.

1.1 One property underneath all of it: the network sits exactly at a critical point

The weights are symmetric (`N(0, 2/n)`, so `50.003%` are negative) and there are no biases, so every preactivation has zero mean and ReLU switches off `50.2%` of neurons at **every** layer — the maximum-activity point of the filter.

```
fraction of negative weights                 50.003%   (symmetry of N(0,2/n))
fraction of gates off, layer by layer         50.2%    constant across depth
"off" fraction, layer 1 → layer 32         50.0 → 52.1%
mean |corr| between gates                   0.03 → 0.01   weak
effective rank of the gate graph             180 → 74     HIGH (≫3)
```

There is **no separation of scales**, so correlations are long-ranged and fluctuations accumulate with depth. That is the signature of criticality, and it immediately yields a negative result: the gate correlations are weak but collectively high-rank — **there is no skeleton, only fog** — so tree-structured methods and belief propagation have nothing to attach to.

The contrast that makes this a statement rather than a metaphor. Take the same weights, all made positive. Then every preactivation `W·a ≥ 0`, ReLU is the identity from layer 2 onward, the network is **linear**, and a closed-form answer gives relative MSE `1.7e-6` (seal on the linearity itself: `9.165e-08`). Sweeping the fraction of negative weights and tracking the participation-ratio rank of the state down the depth:

```
minuses      L1     L2     L5    L16    L32     samples where a whole layer is silent
    0%      3.0    1.0    1.0    1.0    1.0     2.1% at every layer
   25%     44.5    1.3    1.0    1.0    1.0     0
   50%    159.0   99.4   46.0    8.0    3.8     0            ← the benchmark
  100%      2.9    0.0    0.0    0.0    0.0     100% from layer 2 onward
```

Both pure signs are trivial, but for opposite reasons. All-positive is linear and the state collapses to a *ray* — rank `1.0`, not merely small. All-negative is identically dead: with `a ≥ 0` and every weight negative, layer 2 has all gates closed, and the network outputs zero on every input. Only the mixture carries dimension, and the benchmark sits at its maximum. **Seal.** As a by-product this sweep independently reproduces the depth profile of §1 on a different network and sample — and we quote it with its disagreement rather than as a match: the shallow end agrees to `4%` (`159.0` against `165.6`), the deep end only to `36%` (`3.8` against `2.8`), a collapse of `×42` here against `×59` there. What replicates is the shape and the order of magnitude; the deep endpoint is the noisier of the two quantities and we do not claim it to better than tens of percent. **Caveat.** The figures `2.4 / 30.1 / 125.8 / 2.4` recorded in our own earlier notes are the *shallow* cut of this same sweep, and quoting them beside the deep profile — as an earlier draft of this section did — invites a contradiction that is not there. ⟹ **the minus signs do not complicate the problem, they create its dimensionality** — which is why nothing can be built on the assumption "the state is really low-dimensional". The entire difficulty comes from exactly two design decisions: **symmetric weights and no biases.**

One picture that merged three previously unrelated negatives. With no biases, *every* separating hyperplane passes through the origin, so the partition of input space is **central** — cones, not cells. From that, `out(r·u) = r·out(u)` follows immediately, and three old failures become one statement: the radial variance is `1/(2d) = 0.2%`, which is why radial schemes returned zero three times; spherical harmonics on a central partition coincide with the Hermite basis, which is why that family saturates exactly where we measured it saturating; and every observed gate pattern is **unique**, which is why state hashing could not work. **Caveat.** The same fact kills "invert the problem": central hyperplanes are **invariant** under `x → x/‖x‖²`. The degeneracy at the origin is not a coordinate artefact; it is the nature of a homogeneous function.

Two independent derivations of this regime meet the measurement, which is the best check we have that we understand the object. First: under He initialisation without biases `‖Wh‖² = 2‖h‖²`, so norm preservation *requires* exactly half the neurons to be off — the measured fraction with `t > 0` is `0.491` at **every** layer. Second: the angle between the images of two distinct inputs contracts as `θ_{ℓ+1} ≈ θ_ℓ −`

θ_ℓ²/(3π), giving θ_ℓ ≈ 3π/ℓ; at ℓ = 31 that predicts cos = 0.954 and t* ≈ 4.5, against measured cos = 0.9741 and t* = 4.20. ⟹ **the freezing is not an empirical observation — it follows from two lines of algebra**, which is why it is identical across every network in the ensemble.

And one detail of the regime without which all the averages above read wrongly. A median |t| ≈ 3 does not mean the typical neuron sits three sigma from zero on one side — the distribution is **bimodal**: roughly half the neurons are almost always open (t ≈ +3) and half almost always closed (t ≈ −3). The direct signature is Var(a)/Var(p) = 0.48, i.e. exactly half the preactivation variance survives to the activation. ⟹ any estimate "at t = 3" refers to **two different populations**, and the term describing their shared tail turns out to be not a 10^{-4} correction but the leading one.

And this is a peak, not a slope — the problem stands on a singularity. Add a bias b shifting the off-fraction away from 50%:

```
b                     0 (the task)     0.15        → 1.5
relative error          9.4e-05      ×6 smaller   monotonically falling
|γ₃| of preactivations    0.41          0.17
†  extrapolating from the range [0.15, 1.5] back to b = 0 UNDERSTATES the error by ×7.1
   and the skew by ×2.2  ⟹  this is a SINGULARITY, not a smooth continuation
```

A bias of 0.15 would make the problem six times easier. The value b = 0 is a cusp: every quantity peaks there, and no expansion around a neighbouring point extrapolates into it. **Lesson.** This is the shortest explanation of why experience with real networks does not transfer here — LayerNorm and biases in real models **detune** them away from criticality, and this benchmark is stuck on it.

The obvious objection, refuted directly: if the network is critical, isn't the answer fragile? No, and this is measured. Two networks whose weights differ by 1.175e-38 (the smallest normal float32) give **bitwise identical** E[a]. For contrast, a layer-0 perturbation large enough to survive in float32: relative 1e-06 gives 0.000%, 1e-04 gives 0.000%, and even **1% gives only 0.018%**. ⟹ **individual trajectories are chaotic, as they must be at criticality, but the average over the input cloud is stable — it self-averages.** The wall is not sensitivity to the weights; the anchor is more durable than the weights themselves. **The wall is the correlation structure.**

And direct proof that the culprit is the *switching*, not something else in the approximation. We built a differentiable replica of our own estimator (reproducing the original to 0.03% in MSE) and used gradient descent on the weights to **switch the switching off** from layer 8 downward. On **fresh** ground truth the error fell from 1.65e-06 to 7.12e-09 — a factor of **232**. ⟹ the gate ⟨relu'⟩ in our closure is a "frozen mask" approximation, and the mask is not frozen; that, and not the shape of the distribution, costs almost the entire error. **Warning.** Inside the search itself the error fell to 2.6e-14 — pure overfitting to a fixed truth sample, visible **only** on fresh data.

A practical consequence we met from the other side: a critical point is a marginally stable map, so any approximation of the transport operator is amplified exponentially over 32 layers. We tested this directly by replacing a matmul with a stochastic row subsample (half the rows, ×2 rescaling per layer): the mean |crude| came out 1.8e7 where the truth is 0.46, with correlation 0.013 against the exact pass. Rescaling destroys the critical norm 2/n, and 2^{32} does the rest. ⟹ **the mechanistic wall, the variance-reduction wall and the cost wall are one wall, and its name is criticality.**

The instrument. A single knob in our estimator isolates the two branches, and running the official grader at both settings decomposes the score exactly:

```
                         final-layer MSE (50 MLPs)
Monte-Carlo branch alone       1.9227e-06     (V)
analytic closure alone         2.3385e-05     (E²)
submitted blend                1.8115e-06
```

Seal. For two independent unbiased estimators the optimal blend gives `MSE = V·E²/(V+E²)`. Predicted `1.7767e-06`, measured `1.8115e-06` — **2% agreement, on two different MLP subsets.** This model converts any change in analytic precision into a score change and is used throughout.

2. The protocol

2.1 The move

Replace one internal quantity by its true value; leave the rest of the estimator running; measure. That number bounds every idea mediated by that quantity.

The asymmetry is the whole argument: **a ceiling is cheap and permanent, an implementation is expensive and answers only about itself.** We spent three sessions grading approximations of a step correction — Mehler, quadrature, symbolic search, low-rank, tabulated and per-neuron Edgeworth — and never once asked what a *perfect* correction was worth. One run answered it and closed all six.

2.2 Seven rules, each bought with a wrong verdict of ours

1 — Inject the exact quantity, not its best approximation. Rating approximations tells you which is best; only injecting truth tells you whether the best one matters.

2 — Seal the stand at both ends of its axis. With zero substitution it must reproduce your known baseline; with full substitution, a known ceiling or a tautology. One of our stands passed neither (`0.4178%` where the base is `0.507%`; `0.0237%` where the ceiling is `0.0913%`) and its score column was wrong by ×11 — it was re-injecting at *every* step, which suppresses error accumulation, and accumulation is 84% of the error. No real method can do that. Another stand read `1.89%` where a tautology demanded exactly `0`, catching an indexing bug that would have made the verdict on an entire class wrong.

3 — Check that channels are independent before multiplying their ceilings. We were about to report a class ceiling as the product of three separately measured channels. It is invalid here: our recursion starts from `(m, C) = (0, I)`, which is exactly true by construction, so injecting exact corrections into two coupled channels propagates by induction and returns exactly zero.

4 — Re-fit the remaining constants jointly with the substitution. A tuned estimator is a balanced system (§4.1). Substituting truth into one channel while holding the others at values calibrated for the old one measures their staleness, not the substitution. We reported a `×1.031` this way; joint re-fitting left `×1.011…1.024`.

5 — Run the foreign control, and know its blind spot. Same correction shape, content computed from a *different* problem instance. Half our surviving gains died here. But it does not catch a scale-only effect — and neither does leave-one-out, nor A/B cross-validation. All three passed on a branch whose true gain was `×0.9999`.

6 — Put the oracle and the delivery in the same table, as separate columns. A sweep of anchor depth without an oracle column reads "depth does not work". With it:

```
anchor at layer ℓ      0      2      4      8     16     24     30
†  ORACLE anchor     1.24×  1.68×  2.05×  2.90×  5.11×  11.1×  76.3×
   our closure       1.25×  1.29×  1.16×  1.13×  1.10×  1.10×  1.10×
   fitted weight λ*  1.19   0.60   0.26   0.16   0.10   0.09   0.08
†  controls: exact-at-0 ≡ oracle-at-0 (1.242 vs 1.244) · shuffled within layer → λ* = −0.000
```

Depth works spectacularly and **the closure cannot deliver it** — a completely different conclusion, and the one that told us where to work. Since `λ* ≈ Var(signal)/(Var(signal) + bias²)`, the fitted weight also *measures* the closure's bias: at layer 24 it is `3.2×` larger than the sampling error the anchor was supposed to capture.

7 — A class ceiling constrains only methods inside that class, and exceeding it is the test. We compared our `×3.4` against the `×2.67` ceiling of exact third-order polynomial integration and concluded we were at 95% of our own limit. But `×3.4 > ×2.67` — which proves our estimator is *not* in that class (it transports one exactly known anchor; it does not integrate polynomials). **If your method exceeds a class ceiling, your method is outside the class — the ceiling is not wrong.** The ceiling still binds the class it was measured for, which is cubature, sparse grids and QMC.

2.3 A second instrument: separating a block's value from its cost

Under a budget, every block competes with sampling, so "is this block worth its price?" is the only question that matters. Deleting it answers a *different* question, because deletion also returns its budget.

```
run A   remove the block entirely             → (value − cost), mixed
run B   keep it, force its coefficient to zero → value alone
A − B                                          → cost alone
```

Applied to our control-variate block: value `×1.1369` (sign test 36/50, `p = 0.0026`), cost `0.1324%` of the budget, **payback ×10.9**. We had it queued for deletion on the strength of a "leaner is better" argument; three lines of configuration inverted the decision.

When a component both consumes and produces a shared resource, never measure it by removal. Neutralise it in place, and take the cost from the difference.

2.4 A third: sort ideas before building them

Late in the project we tabulated **thirty-nine measured verdicts** — methods actually built and scored, not ceilings — and labelled each by what it structurally does:

```
label                                  n    median   share > ×1.05
REMOVES error where truth is known     8    1.043        38%
PREDICTS how error will propagate     18    1.000         0%
converts cost into samples             5    1.009        20%
requires an unknown quantity           8    1.000         0%
```

Eighteen of eighteen "predicting" methods returned nothing. Everything that ever crossed `×1.05` in this project is four items: whitening (`×1.98`), cubature (`×1.48`), dead-neuron pruning (`×1.20`), one control-variate family (`×1.196`). Three remove error where truth is known; the fourth converts saved cost into samples.

The taxonomy was written after the first thirty rows, then applied to three branches not yet run: it predicted `≈1.000` for two (measured `1.000`, `1.002`) and a catastrophe for the third (measured `0.003`).

Caveat. This is not a random sample of methods — we measured what looked promising, and the "cost" row has one catastrophic outlier. The remove/predict contrast holds on medians, on shares, and row by row; that contrast is the finding.

3. What the protocol found

The whole of it on one page. Each row is one run of the protocol: an internal quantity is replaced by its exact value, everything else keeps running, and the result is read as `×precision` (cosine error of the closure) and as `×score` on the official grader through the blend model of §1. Our unmodified submission is `1.0`. A row is a **ceiling**: no idea acting through that quantity can do better, however it is implemented.

what is made PERFECT	×precision	×score	verdict
the step formula (exact per-step error)	1.19	1.032	ceiling of a whole class
linear correction in the closure's own basis	1.027	1.001	ceiling of a whole class
2-D quadrature instead of truncated Mehler	1.001	1.000	the series is not the problem
genetic-programming search over expressions	1.002	1.000	converged to a hand-found constant
rank-r non-Gaussian step kernel (r = 4)	1.04	1.001	–
per-neuron Edgeworth on OUR moments (γ_3, γ_4)	1.02	1.000	needs exact moments to switch on
true OFF-DIAGONAL of C	1.149	1.012	the covariance channel is nearly empty
true DIAGONAL of C	0.118	0.930	← makes it 8.5× WORSE
true C entirely	1.098	1.008	← worse than the off-diagonal alone
true m AND true C (allm)	30.86	3.363	⇒ 5.5e-08, about 5th place
+ true per-neuron γ_3, γ_4 (edgm)	~160	22.78	⇒ 8.16e-09, between #2 and #3
per-network blend weight W*_i	–	1.029	realisable version: ×0.967
per-network output scale a_i	–	1.25	realised on the grader: ×1.003
true (m, C) injected after layer K = 1	–	1.00	shallow layers are worthless
true (m, C) injected after layer K = 24	–	3.7	the prize is at depth
exactness through 3rd Hermite degree	3.4	–	§3.3, ±0.6; we sit on it
exactness through ANY Hermite degree	7–9	–	§3.3, ceiling of SMOOTHNESS itself
any additive one-dimensional class	1.89	–	§3.3, the bottleneck is additivity
weights read at b < 10 bits	–	–	§3.8, closes coarsened-weight schemes

Three readings carry most of the report. `allm` says a *perfect* two-moment closure reaches about 5th place — the two-moment class is exhausted before the podium. The true **diagonal** of `C` making the answer `8.5×` *worse* says the tuned closure is a balanced system of cancelling errors (§4.1), which is why a hundred single-channel corrections returned `×1.0x`. And `edgm` — exact moments *plus* exact per-neuron third and fourth cumulants — is the only row that reaches the neighbourhood of the podium, and even it stays `×5.4` short of `#1` on our stronger instrument (§3.1). That is how we know the leading entry is not propagating moments (§7); against `#2` our two instruments straddle parity, so we make no claim there.

The full registry behind this table — all 54 closed classes in the form *claim* → *number* → *seal* → *verdict*, with the stand, the sample size and the controls for each — is the companion volume, **The Compendium**. This report carries the argument; the companion carries the evidence.

3.1 The ceiling of the class we chose

Substituting ground-truth moments into exactly one channel, recalibrating the scalar constants separately for each mode, 48 networks, two independent replicas:

mode	rel. error	×closure	×score	what was substituted
base	0.5072%	1.000	1.000	nothing † = grader's own 0.5070%
off	0.4732%	1.149	1.012	true off-diagonal of C
diag	1.4745%	0.118	0.930	true DIAGONAL of C ← breaks it 8.5×
full	0.4841%	1.098	1.008	true C entirely ← worse than off
allm	0.0913%	30.857	3.363	true C AND true m

`allm` — the ideal two-moment closure — scores `5.5e-08`, about 5th place.

```
#1 on the board   1.50e-09   ⇒ ×37 above us even at zero moment error
#2                7.20e-09   ⇒ ×7.7
#3                3.68e-08   ⇒ ×1.5
US WITH ORACLE    5.50e-08
```

Going further, to a full moment oracle, we have two measurements and they disagree by `×1.8` — so we quote both. Direct injection of exact per-neuron γ_3, γ_4 on top of exact `(m, C)` gives `8.16e-09` (12 networks × 524288, convergence ratio `0.0005`). An independent sieve, which fits the score against moment accuracy and extrapolates to perfect accuracy, gives `1.48e-08`. By our own rule 1 the direct injection is the stronger instrument; by conservatism about our own thesis it is also the number that makes the thesis hardest to defend, so it is the one we argue against:

```
                          vs #1 (1.50e-09)   vs #2 (7.20e-09)
direct injection  8.16e-09       ×5.4              ×1.13
sieve             1.48e-08       ×9.9              ×2.1
```

⇒ a full moment oracle — exact `m`, exact `C`, exact third and fourth per-neuron cumulants — lands level with second place at best and does not reach first. The weaker form of this claim is the one we rely on: no moment-based state reaches `#1`, by a factor of `5.4` even on the most favourable of our two measurements.

Seals: `base` reproduces the grader's number to four digits; `allm` landed at `0.0913%` against a pre-registered prediction of `0.10–0.20%`; at layer 0, increasing `N` by ×64 dropped both figures by exactly `√64`.

Internal consistency of the table itself. The two right-hand columns are not independent measurements: `×closure` is $(x_{base}/x)^2$ and `×score` follows from `x` through the blend model of §1, so recomputing one from the other is a check on the whole table. Predicted against tabulated: `off 1.011/1.012`, `full 1.007/1.008`, `diag 0.933/0.930`, `allm 3.267/3.363`; and $(0.5072/0.0913)^2 = 30.861$ against the tabulated `30.857`. The three near-base rows agree to `0.3%`; `allm` is out by `2.9%`, which is expected — at small E^2 the blend is dominated by the branch we know only to `±26%` (§1), and this is the size of that uncertainty showing up where it should.

By our own rule 2 this stand is sealed at only one end, and we say so rather than let it pass. The zero end is solid — `base` reproduces the grader. The far end has no tautology available: `allm` *is* the exhaustion of the two-moment class, so there is nothing further to make exact within it that would not simply be the answer. All that carries the far end is the pre-registered prediction quoted above, and a pre-registered interval is weaker evidence than a tautology. So the headline number of this section rests on a one-ended stand — a reader who wants the far end confirmed should re-run the protocol rather than take our word for it.

Caveat. Read the ranks carefully, because we got this wrong ourselves: all Phase-1 entries were re-scored on an updated cost model shortly before the deadline, and one entry's *adjusted* score moved by ×92 at unchanged MSE. Comparing an adjusted score against someone else's raw MSE — which an earlier draft of this document did, in four places — overstates the gap by more than an order of magnitude. All five numbers above are adjusted scores.

3.2 The ceiling of "a better step formula"

Four independent attacks on the per-step formula returned `×1.001` (exact 2-D quadrature instead of the truncated Mehler series), `×1.027` (best linear correction in the closure's own basis, with an oracle), `×1.002` (symbolic search over expression trees — it converged to a constant we had already found by

hand), and ×1.04 (rank-4 non-Gaussian kernel).

Rather than continue rating approximations, we injected the *exact* step error into the real recursion, b_l = step(true m_l, true C_l) − true m_{l+1}, with the remaining constants re-scanned jointly:

```
mode      rel. error   ×precision   ×score    what was injected
base        0.4796%      1.0000      1.0000   nothing
inj         0.4033%      1.1893      1.0315   exact per-step error, c* = 0.50
injp        0.4004%      1.1977      1.0331   the same, transported to our own state
†  foreign  0.5450%      optimum at c = 0.00, then monotone decay
†  injm     0.0000%      tautology: exact step on exact inputs reproduces truth
sign test inj vs base at the same grid point: 15/16
```

inj is an upper bound for every idea of the form "a better step formula" — all γ schemes, quadratures, discovered expressions, rank-r kernels and learned corrections *together*. The whole class is worth ×1.19 in precision and +3.2% in score. An Edgeworth term with exact per-neuron γ claims 11% of that ceiling.

Seal. One seal caught the instrument: injm must be exactly zero and read 1.89% in the smoke run — an indexing error (the input of the *next* step is (m_{l+1}, C_{l+1}), not C_l). Without it the verdict would have been wrong.

Corollary that saved us from a whole research direction. The step error *is* Edgeworth, to R² = 0.983 (cos = −0.9915), if you use per-neuron γ — and exactly zero (cos = +0.0003) if you use the per-layer tabulated γ we had been using, because those describe the γ of one-dimensional *projections*, a different object with mean 0.000 and standard deviation 0.42. So the mechanism is understood at 98%, the channel still does not exist, and it would not matter if it did: a perfect step is worth ×1.19.

A null correlation is a reason to audit the instrument, not to declare orthogonality.

3.3 The ceiling on the sampling side belongs to ReLU, not to us

The sampling branch admits exact bookkeeping, which we recommend to anyone estimating a Gaussian expectation because it says precisely what is still available:

```
antithetic pairs [z, −z]   cancel ALL odd orders in z        exactly, bitwise
whitening (sample cov = I) cancels ALL quadratic forms       †  relative Δ 1.2e-08
⇒ the first uncontrolled order is the FOURTH
```

A 4th-order control variate, constructed to have exactly zero mean by Isserlis' theorem, gives R² = 0.0035 at 0.734% of the budget. And the total left is bounded: measuring the Hermite degree spectrum directly through the Mehler identity Cov(F(z), F(ρz+√(1−ρ²)z')) = Σ_k ρ^k Var_k(F), which avoids any expansion in 256 dimensions:

```
degree k     1       2       3       4       5       6
a_k       0.338   0.294   0.108   0.058   0.032   0.010    monotone, stable at ×4 samples
†  pure He₁…He₅ recovered as 1.000 / 1.001 / 0.994 / 0.962 / 0.945
†  a mixture He₄+He₅ recovered as 0.487 / 0.466 ⇒ the instrument separates 4 from 5
```

Degrees 1–3 hold 0.74 of the variance ⇒ a method exact through third order has a ceiling of ×3.4, and our estimator sits on it. **Caveat.** Honestly: ≤3 ≈ 0.71 ± 0.05, so the ceiling is 3.4 ± 0.6 and the right statement is that we are *within the error bar* of it, not exactly on it.

And the ceiling of the whole class is a second, larger number, which is the one that matters. 10–16% of the variance sits at degrees ≥ 7, and no finite-degree method removes it, so a method exact through *arbitrarily* high degree still saturates at 1/(0.10…0.16) ≈ ×7–9. The two numbers answer different

questions: ×3.4 is what exactness through third order buys, ×7–9 is what exactness through *any* order buys. Everything below is about the second.

And the residual is not "complicated structure" — it is the kinks. Replacing ReLU by softplus_β and re-measuring:

```
β        tail (degrees ≥ 7)                  a₁ for context
relu     0.190 / 0.16 / 0.15  (3 networks)   0.336
16       0.154 / 0.123 / 0.130               0.28–0.38
4        0.021 / 0.044 / 0.059               0.34–0.45
1        0.000 / 0.000 / 0.005               0.62–0.73
```

At β = 4 the function is still 55–66% nonlinear and the high-degree tail has already collapsed 3–7×. A kinked function has a *power-law* Hermite tail, not an exponential one — Gibbs' phenomenon in other coordinates — and no polynomial takes it. ⟹ **that ×7–9 is therefore a ceiling of smoothness itself, not of any particular scheme.** Every smooth approximation — quadrature, cubature, QMC, polynomial control variates — buys the same first three degrees and stops, at any price.

And then a third measurement changed what the kinks *are*. If the tail is kinks, the natural next move is to work *with* the singularities — cells, Gegenbauer reconstruction, poles near the discontinuity, sampling aimed at the switching hyperplanes. We measured the distance from a sample to the nearest switching surface in the input metric, d_ℓj = |p_ℓj| / ‖∇_z p_ℓj‖, with exact Jacobians:

```
†  layer-0 control (there ∇p = w, so d must be exactly |N(0,1)|)
quantile     0.05     0.25     0.50     0.75     0.95
measured    0.0621   0.3177   0.6737   1.1505   1.9577
theory      0.0627   0.3186   0.6745   1.1503   1.9600

median distance to the NEAREST kink of the whole network:  0.00011
fraction of probability mass within ε of a kink:  ε = 0.001 → 99.1% · ε ≥ 0.003 → 100.0%
†  forced by counting: ~8000 hyperplanes × φ(0) ⟹ 1/(8000·0.8) ≈ 1.6e-04, measured 1.1e-04
   within a single layer: predicted 0.005, measured 0.0033
```

The kinks are not a rare feature contributing 10% — they are a dense medium with a spacing of 10⁻⁴. Every sample sits essentially *on* a kink. ⟹ **this inverts the recommendation**: the entire class of methods that *localise* the singularity is precisely the wrong one, because there is nothing to localise; what works is the class that treats singularities *on average*, which is plain averaging. It also closes exact cell integration without a stand — a cell has diameter ~10⁻⁴ in 256 dimensions, so no per-cell cost can matter — and it explains the rational-approximation result below.

Six theories of approximation, one number. Rational approximation (Newman, Padé, AAA, lightning solvers), ridgelets and the Radon transform, nonlinear approximation in Besov spaces, Gegenbauer reconstruction, Gaussian measure of polytopes, Euler–Maclaurin with jump corrections — all six reduce to one question, because in 256 dimensions the only thing that integrates exactly against a Gaussian without sampling is a *ridge* function g(w·z) (w·z ~ N(0,1), a one-dimensional quadrature). So the shared ceiling is the variance share taken by c + Σ_{i≤m} g_i(u_i·z) with arbitrary g_i:

```
m              1      4     16     32     64    128
share taken  0.314  0.359  0.425  0.455  0.471  0.466   ⟹ ceiling ×1.46 … ×1.89
†  linear m=1 → 0.999 · Σ₈relu m=8 → 0.998 ·    random directions m=32 → 0.029
```

×1.89 equals the ×1.93 measured for a purely polynomial control variate ⟹ an arbitrary one-dimensional class buys nothing over polynomials. **The bottleneck is additivity, not the basis.** And Newman's theorem, tested at *our* parameter rather than its canonical example: along a Gaussian line the

network has 1384–1945 kinks, and the advantage of rational over polynomial approximation has already vanished by K = 16.

3.4 Six ways to aim the sample, six different mechanisms, all dead

Conditioning does not commute with composition. Split z = √(1−θ)z_A + √θ z_B and integrate z_B analytically, giving a smoothed ReLU. Variance can only fall and the mean is preserved. Predicted ×1.12 / ×1.34 / ×1.89; measured, smoothing *only layer 0*: ×0.043 / ×0.011 / ×0.002. **Seal.** The unbiasedness of that layer is intact (4.2e-04). The mechanism is the s·φ(t) term: conditioning preserves m but reduces s, and the next layer reads s. Rao–Blackwell is unbiased for a fixed f; the output is a composition.

There is nothing to gain inside a linear region. Inside one activation cell the network is exactly linear, so conditioning on the cell is tempting. Measuring the inner term by walking outward to the first pattern change:

```
object                    hyperplanes   INNER share of variance, %
one layer, 4 neurons            4              42.65     (saturated)
one layer, 16 neurons          16               4.35
one layer, 64 neurons          64               0.348
two real layers               512               0.0078
REAL NETWORK, 32 layers      ~8000              0.0005
†  control monotone across four orders of magnitude
```

All the variance is in *which* cell you land in ⟹ the whole Rao–Blackwell family has a ceiling **below ×1.001** — ×1.000005 reading the last row as the percentage it is labelled, and ×1.0005 even under the most generous misreading of that unit. We quote the bound that survives both readings, because the verdict does not depend on which is right.

Importance sampling moves the difficulty into the normalisation, which is our problem. Self-normalised, even with the ideal proposal q ∝ f: ×0.980. Unnormalised, needing the exact E[g]: ×4.2e+32. The reason is exact — for q ∝ p·g the self-normalised variance is ∝ E_p[(f−I)²/ĝ], which division by ĝ does not reduce.

And "find the hot region and sample it" fails on measure, not on search. The fraction of the sphere inside a cone of half-angle θ in 256 dimensions is I_{sin²θ}((n−1)/2, ½): at 0.60 rad that is 3.0e-65. Points bred there carry importance weights of order 1e-65.

Underneath all four is one observation, and it is narrower than the one we first wrote down. An estimator that re-expresses the *same* approximation of the *same* functional from the *same* draw is the same estimator wearing different notation. Sharpest form: an Edgeworth estimator built from the sample moments gives ×0.996 against the flat sample mean, and the correlation between the two estimators' errors is **+0.9964**.

Caveat. Stated generally — "no function of the same samples can reduce their variance" — this would be false, and our own §5 is the counterexample: whitening is a function of the same draw and is worth ×2.06, and the λ control variate is built from the same draw and is worth ×1.14 at a payback of ×10.9. The distinction is what the function is allowed to *use*: both of those inject an **exactly known** quantity from outside the sample — E[zzᵀ] = I and E[h₁] = ‖w‖/√(2π) — and correct the draw against it. Re-weighting that introduces no exact external anchor buys nothing, and the +0.9964 is what that looks like when measured. **Rule.** The operative question for any proposed variance reduction is therefore not "is it a different estimator" but **"what exactly known quantity does it pin the sample to"** — and §5 inventories every one that exists in this problem.

And a fifth way, closed by an argument rather than a search. If some sample sets are better than others, one could find them offline once and reuse them — the budget allows sets far larger than the Delsarte bound. Measured across 96 sets × 8 networks: the split-half correlation of a set's quality *between networks* is **`−0.010`** against a noise level of `±0.204`, and the choice of set explains `1.5%` of the spread. **There are no universally good sample sets**; the error is an interaction between set and network.

Mechanism. The argument that makes it a theorem rather than a measurement: Monte-Carlo is unbiased, so the expectation of its error over sample sets is zero for *every* network. Therefore **no function of the weights alone can predict the sampling error** — the object simply has no mean to predict. (This is not in tension with our later finding that a deterministic feature of `W` predicts the *closure* error at LOO-$R^2 \approx 0.35$: that error is a bias, not a fluctuation.) And a related test with a clean sign: the network's own weight columns are already `N(0, 2/n)` vectors, so they are free candidate sample points — using *foreign* weights gives `1.34e-06`, the same as fresh Gaussians (`1.13e-06`), while using the network's *own* weights gives `1.80e-05`, 13× worse, because $x = c \cdot W_0[:,j]$ guarantees neuron `j` fires.

Seal. One more free lunch we checked and did not get. The network depends on `z` only through $z^1 = W_0^\top z$, a bijection — so any feature of z^1 has an exactly known Gaussian mean and is a free control variate requiring no extra forward pass. The first-moment feature gives `0.989×` (noise, because antithetic pairs already removed it) and the He_4 feature gives `2.31×` **worse** — even with an oracle coefficient it is only `1.22×`, because estimating the coefficient injects more variance than the control removes. **Thirty-two layers of criticality decorrelate the first layer from the answer**, which is the same wall as the rank-`2.74` backward operator of §4.2, met from the input side.

And one structural measurement that closes low-dimensional methods in advance: knowing `k` coordinates of `z` explains about $0.3 \cdot (k/256)$ of the output variance at small `k` — **worse than uniform.** Eight coordinates give 1.0%, matching that line; *half* of them give 20%, slightly above it and still far below the 50% a uniform split would hand you.

3.5 A ceiling on the competition itself: the flat-budget theorem

One ceiling is not about our method at all. Write $u = C/B$ for the fraction of budget used, `P` for the cost that does *not* scale with the sample count, $N = (uB - P)/c$ for the samples you can afford, and $K = V \cdot N$. Then

```
score(u) = u · V·E²/(V+E²),        V = Kc/(uB − P)
d(score)/du = MSE·[1 − (uB/(uB−P))·(1−w)]   ⇒  leaving the floor ⇔ w < P/(uB)
```

with $w = V/(V+E^2)$ the blend weight. Setting $P \to 0$ gives $\text{score} = (Kc/B)/(1 + Kc/(uB \cdot E^2))$, which increases monotonically in `u` and tends to `Kc/B`. ⟹ **the score of a pure Monte-Carlo estimator does not depend on the budget at all.** Spending twice as much buys exactly nothing.

Seal on our own numbers: one sample costs $32 \cdot 2 \cdot 256^2 = 4.194\text{e}6$ FLOPs, so $65000 \cdot 4.194\text{e}6 \approx B$. Running $u = 1$ with $N = 65000$ and $u = 0.1$ with $N = 6500$ gives `6.741e-07` **both times.**

Two consequences we did not expect:

- **The better your deterministic branch, the stricter the optimal budget floor.** Improving the closure raises `w`, and the crossing condition is $w < P/(uB)$, so a better analytic component pushes you *further* from wanting more budget. Our margin is `0.0883` against `0.0261` — a factor of 3.38 inside the floor.

- **The roles of the two branches change in the weight, never in the budget.** At our current analytic error the blend weight is `0.088`; at an analytic error of `0.034%` it would be `0.951`, i.e. Monte-Carlo would contribute 5% and could be dropped entirely. The estimator would become deterministic, and the variance of a private re-evaluation would go to zero.

⟹ **cost is not the gate; accuracy is.** We spent 91 sessions looking for cheap corrections while sitting on `×16` of unused room: the entire closure costs `1.66e9` FLOPs, which is `6.1%` of the budget floor.

3.6 The analytic advantage is a small-sample phenomenon, and frozen constants fake a floor

The flat-budget theorem has an empirical twin that is worth stating separately, because it tells you *where* on the budget axis analytic work is worth anything at all. We re-fitted every coefficient honestly (split-half) at several sample budgets and measured the advantage of the full estimator over plain Monte-Carlo on the same draw:

```
sample budget      ×1       ×8       ×64
advantage        4.2408   1.4723   1.0946
optimal analytic weight g(λ)   1.5      0.5      0.00
†  cross-check: 3.9668 here against 3.4 elsewhere in the corpus, a ratio of 0.86;
   truth noise is known to compress this figure by about 0.81, so the two instruments
   agree to 6% — close, but quoted as a cross-check rather than as a seal
```

At `×64` the optimal weight on the analytic branch is **exactly zero**: the method degenerates into Monte-Carlo, on its own optimum. ⟹ **our whole apparatus buys `×2.19` over raw sampling and only in the small-sample regime — and the competition is set exactly there.** A scoring rule that granted more samples would erase the difference between every entrant's analytics, which is a design property of the benchmark, not of any method.

The measurement law this produced, and it caught us: *coefficients are curves, not constants — re-fit them before declaring a floor.* At `×16`, switching off the shrinkage and blend terms with our shipped constants costs `+66.5%`; at `×64` it costs `+225%`. We had read those numbers as a bias floor of the method. They were the staleness of frozen constants. The honest curves all descend: `g(λ) 1.5 → 0.5 → 0`, blend weight `0.045 → 0.005`, the closure-depth switch `8 → 4`. **Seal.** The control that makes this safe to report: at `×1` — the shipped operating point — an oracle re-fit gains `−2.2%`, i.e. there was nothing to tune, and the submission was left unchanged.

Caveat. A related trap, same shape. We once measured a catastrophic collapse of our estimator at width 1024 and were preparing to branch the code by width. The benchmark scales budget as `4×/4× depth-and-budget`, so the invariant is the *number of samples the budget buys* (`∝ depth·width²`), which is `×16` for width 1024, not `×4`. At the correct scaling the unmodified estimator gives `2.2167×` and the "fix" gives `1.5112×`. **When you change a parameter, check what is tied to it** — the break sits between `×4` and `×8`, and we had been solving a regime that does not exist.

3.7 The family everyone tries first: learn it

Before the closure existed we spent an entire arc on the obvious modern move — *train something*. Every variant failed, and the failures are structured enough to be worth more than the successes. We report them because "just fit a model" is the first thing a reader will propose, and because two of the mechanisms are properties of the problem rather than of our training.

A. The student map: six students, all ending below plain sampling (the companion adds a seventh, layer-folding). Train a surrogate from an intermediate activation a^{16} to the output a^{32} — the second half of the network is nearly linear (R^2 of a linear head rises `0.44 → 0.88 → 0.99` at layers 0/16/30), so this

should work:

```
student                                bias²        why it fails
per-neuron affine  γ·z+β              ×1.00        provably a SUBSET of antithetic sampling
per-neuron nonlinear 1-6-6-1          600–2400×    pointwise fit error 1.9e-04 amplifies
wide linear (OLS) 256→256             1e-07 ✓      but needs E[a¹⁶] — the cost wall
wide nonlinear 256-128-64-256         2e-05        200× worse than the LINEAR student
residual/skip 256-32-256              6–40×        tail blow-up: pred |max| = 320 at signal 0.5
bounded activation (tanh)             2e-05        stable, but 24× worse when cost-matched
```

Three mechanisms, each transferable:

- **The task penalises capacity.** The *linear* wide student has bias `1e-07`; adding nonlinearity makes it `2e-05` — **two hundred times worse.** Ordinary least squares guarantees a zero-mean residual for free; a network trained by SGD has no such guarantee, so flexibility bends exactly the quantity we are estimating. **Linearity is not a weakness here, it is protection.** Every added capacity — depth, bottleneck, skip — made the mean worse, in that order.
- **Unbounded students explode on the tails.** Predicted `|max|` reaches `2.7e5 … 7e11` across ReLU/LeakyReLU/GELU/SiLU/SwiGLU, because `a¹⁶` has heavy tails after sixteen critical layers and a linear read-out extrapolates. A *bounded* activation cures the explosion (`tanh` holds `max = 5.6`) — and then loses on bias instead. The failure was never the architecture; it was unboundedness meeting heavy tails.
- **Knowing `E[control]` costs as much as the answer.** The best student by bias needs the mean of its own input. That is Monte-Carlo of the same kind, and the regression slope amplifies its noise. Cost-matched at every budget from 4k to 200k samples, the surrogate scheme is `1.07–2.41×` *worse* than plain sampling and **never crosses** — more budget makes it worse, not better.

B. Approximating ReLU itself is actively harmful. Replacing ReLU by a learned `1-6-6-1` micro-network with pointwise error `1.9e-04` gives a result `600–2400×` worse. The same shape appears whenever anything inside the network is approximated: a Gaussian anchor mid-stream `+2e9%`; truncating the cumulant recursion at layer 24, `7×`; averaging or summing mid-stream, `400×`; folding pairs of early layers into learned blocks, `255×` (linear) and `10⁶×` (MLP). **The network is a pencil balanced on its point.** The exactness of ReLU — the kink at zero and `g(0) = 0` — is load-bearing for the criticality that makes the answer what it is.

And the weights have no compressibility either, which we checked separately because it is the cheapest possible saving: replacing each `256×256` weight matrix by its rank-`r` truncation gives MSE `0.52 / 0.40 / 0.05 / 0.0071 / 6.8e-07` at `r = 16 / 128 / 192 / 224 / 256`. Discarding 12% of the spectrum is still `1700×` worse. iid Gaussian weights are full-rank **by structure, not by redundancy**, and any perturbation flips gates that then amplify through 32 layers. (The irony worth recording: the factored form `U·S·Vᵀ` costs *more* than the plain matmul until `r` is small enough that the answer is garbage.)

C. Amortisation removes the cost but not the wall. The one machine-learning route that escapes the cost argument entirely is a hypernetwork trained *offline* on many networks with ground truth — training is not charged per instance. We built the strongest version: learn not the whole mean but the *residual* that a cheap analytic base misses (60 networks, 48 train / 12 test, per-neuron features).

```
base only                     test MSE  5.64e-01
base + learned residual                 5.88e-01   ← worse; it fit noise
†  corr(predicted residual, true residual)  0.0012
```

The mechanism is the interesting part, and it is a statement about the object. The ground-truth scale varies across networks (`rms 0.50 → 1.36`) while the cheap base is essentially the *same number for every network* (`rms ≈ 0.55`): under criticality the mean-field variance collapses to a fixed point over 32 layers. **The cheap base carries zero instance-specific information** — all of the between-network signal is exactly the correlation structure it discarded, and cheap per-neuron features correlate with it at `0.001`.

⟹ there is no cheap descriptor of a network from which its mean can be recovered. Computing an instance-specific descriptor *is* the work. Amortisation moves the cost offline, and finds nothing there to amortise.

3.8 A different axis: how much of the weights the answer actually consumes

Every ceiling above has the same shape — *substitute the exact value of an internal quantity*. One more is orthogonal to all of them: **limit how much information about `W` the method is allowed to read at all.** Quantise the weights to `b` bits and measure how far the answer moves:

```
D(b) = ‖E[out(W_b)] − E[out(W)]‖ / ‖E[out(W)]‖
```

Pairing is essential — both networks are run on the *same* `z`, so the Monte-Carlo noise cancels and `D` is readable far below the `0.29%` noise floor of a single run. Measured on 12 networks with symmetric round-to-nearest per column:

```
b        2      3      4      5      6      8      10      12      24
D(b)  1450%   309%  46.2%  18.2%  8.95%  2.11%  0.498%  0.139%  3.3e-05%
†  se 4.05%  1.04% 0.333% 0.212% 0.132% 0.039% 0.011%  0.003%  0.000%
Δw           80.7%  29.5%  12.6%  5.89%  2.85%  0.696% 0.173%  0.043%
```

Seal. `b = 24` returns `3.3e-05%`, i.e. zero. **Seal.** The weight error falls by `×18.1` between `b = 4` and `b = 8` against a theoretical `×16`. **Seal.** Every `D` sits at least `45×` above its own resolution. **The transfer is constant**: `D / Δw = 3.05` for all `b ≥ 5`, across a factor of 100 in perturbation size, so the table reads as a single law rather than nine points.

Our own cosine closure error is `0.53%`, which lands between `b = 9` and `b = 10`.

What this does and does not say, stated carefully because we first stated it wrongly. This measures **one** coding scheme. It therefore establishes that `b` bits *suffice*, not that they are *required*: the true rate–distortion function is an infimum over all codes and lies below this curve. It closes the family "coarsen the weights and compute from the coarse version" for `b < 10`, and nothing wider. In particular it says nothing about a method that computes an arbitrary *statistic* of `W` rather than quantising it — a descriptor of 256 numbers trivially exists, since the answer itself is one. Our earlier claim that this bounds learned descriptors was a logical inversion, and the empirical failure of the hypernetwork (§3.7) stands on its own evidence, not on this curve.

Mechanism. The by-product is worth more than the headline: sensitivity to the weights grows by `×52.9` with depth.

```
layer    0      1      2      4      8     16     24     28     30     31
D_L%  0.0132 0.0769 0.0659 0.1170 0.1886 0.3583 0.5240 0.5488 0.6879 0.6994
```

Seal. At layer 31 the transfer is `0.6994/0.696 = 1.005` — exactly unity, as it must be: a relative perturbation of the last matrix passes into the output one-for-one. **Seal.** At layer 0 the transfer is `0.0190` against `0.018` recorded from a *different* stand in a different session with a *random* rather than

quantisation perturbation — 5% agreement across implementations. **Seal.** `√(ΣD_L²) = 2.187%` against `2.089%` for all layers perturbed at once, a ratio of `0.96`, so **layers corrupt the answer independently**, without systematic accumulation.

Caveat. We record one prediction failure here because its mechanism is reusable: we forecast `D(8) ≈ 0.04%` and measured `2.11%`, a miss of `×50`. The forecast came from transporting the layer-0 transfer to all layers — and that transfer grows `×53` with depth. **A number obtained by dividing two numbers from different stands is not a measurement**, which is why the per-layer profile had to be measured separately, and it then produced the two best seals of the run.

Together with §4.2 this gives the geometry of the problem in three lines:

```
the answer is born early     (layers 0–7 carry 109% of the projection, §4.6 fact 11,
                              or 50.1% of the answer by the identity of §4.2)
the error is made late       (84% is accumulation; backward operator rank 2.74)
weight sensitivity is late   (the ×52.9 profile above)
```

Caveat. Those two shares — `109%` and `50.1%` — are not the same measurement quoted twice. They decompose different objects: the first projects the answer onto the ensemble of realised linear maps, where later layers subtract and a share may exceed `100%`; the second sums the per-layer covariance terms `c_ℓ` of the exact identity, which are non-negative and must total `100%`. Both put the mass early, and that is the only claim we make from them.

⟹ a shallow layer can be coarsened almost with impunity; the last one cannot.

4. Why the ceilings are where they are

4.1 A tuned closure is a balanced system of cancelling errors

The most robust qualitative finding in the project, and the one we would most like others to test on their own closures.

We trained a model to predict the per-step error of σ from local quantities (196 608 rows: 24 networks × 32 layers × 256 neurons), then applied the prediction as a correction with a scanned coefficient:

```
channel      R² (held-out networks)   optimal correction coefficient
   m                 0.353                    0.50   →  ×1.011…1.024
   σ                 0.836                    0.00   →  ×1.0000
†  shuffled features 0.003
```

The model can say, with `R² = 0.84`, how wrong σ is at each step — and correcting it yields exactly nothing. The optimum did not move by a single grid step. The σ error is not a parasite; it is load-bearing. Corroborating, all independent: the true covariance diagonal makes the answer `8.5×` worse; sharpening `C` toward its true spectrum makes it `125×` worse; supplying the true `C` entirely is worse than supplying only its off-diagonal; and the global scalar κ on `C` has its optimum at `0.9985 < 1` while the true `C` at layer 31 is ~25% *larger* than ours.

A decomposition that isolates the culprit exactly. For the whole family of linear filters and shrinkage rules we split the requirement in two: give the method an oracle for the *signal* and its own estimate of the *noise*, then the reverse. Oracle signal + own noise gives `×1.2064` (better on 100% of networks); own signal + oracle noise gives `×0.9957`. **Knowing the noise is worth exactly nothing; the entire wall is the signal estimate.** That single measurement closed a class we had approached from four directions.

⟹ **substituting truth into one channel of a tuned closure is expected to hurt.** Any evaluation of a "more accurate submodel" must re-fit the system's remaining constants jointly, or it measures their staleness instead.

The same wall, reached from the opposite side, with a mechanism. Independently we built a correction that genuinely works: exploiting the fact that the non-Gaussian residual of a single network is nearly rank 2 and its top directions lie in the span of six quantities the closure already computes, we compressed it into **21 numbers per layer**. It does what it should — the σ chain becomes `1.9–2.7×` more accurate (oracle `4.2×`), with leave-one-out transfer across networks holding at `0.58–0.78` and every control behaving (`Ĝ = 0` reproduces the Gaussian closure to `1.9e-05`; a random `Ĝ` gives `−0.6…−0.9`; a `Ĝ` from a *different layer* degrades).

On 6 networks it gave `+7.9%`. On 32 independent networks with frozen constants it gave **`×1.125` against `×1.134` for the uncorrected branch — i.e. nothing.** The progression `8 → 16 → 24 → 32` networks reads `1.305 → 1.161 → 1.110 → 1.125`.

The mechanism, which is the transferable part: a more accurate closure justifies a larger shrinkage coefficient — but closure accuracy *varies across networks*, and a larger coefficient triples the variance of that variation. Keeping the improved closure with the *old, conservative* coefficients lands exactly on the baseline. **Making the analytic component twice as accurate bought zero**, and that is a property of the coupled system, not of the improvement.

Caveat. Two instrument lessons were paid for here. First: **measure the chain you are fixing.** The correction enters the *covariance*, and we measured the error of the *mean* at layer 31 and declared failure; a per-layer cut showed it doing exactly what it should. Second: **if a control beats the oracle, suspect the metric.** Our first capture metric rewarded an oversized correction — a `Ĝ` taken from the wrong layer scored `0.945` against the oracle's `0.641`. With the correct metric `1 − ‖R−Ê‖²/‖R‖²`, the controls went negative, as they must.

4.2 The answer is born early; the error is made late

The complementary question to "where is the error made" is "which part of a perturbation survives to the answer". That is the spectrum of the backward operator `B_l = (Φ_{l+1} ⊙ W_{l+1}) … (Φ_31 ⊙ W_31)`, where `Φ_i = Φ(t_i)` is already computed by the closure, so the measurement is free:

```
l → 31   effective rank   σ₁/σ₂₅₆    OUR ERROR inside top-16 directions
   0          2.74        1.4e+31              5.52%
   8          5.78        3.0e+29              6.95%
  16         11.20        1.0e+31              6.06%
  24         23.85        6.6e+28              6.14%
  30        100.49        3.6e+14              5.11%
†  isotropic control: 16/256 = 6.25%
```

Condition number `10³¹`. Of the 256 numbers describing a shallow layer, roughly three reach the output. The information is destroyed, irreversibly, by the network. Three consequences follow without further runs: state reconstruction from the output is impossible at any precision one could actually compute in — the operator is invertible in exact arithmetic, but `10³¹` against float64's `10¹⁶` means the inverse is noise, so this is a numerical impossibility rather than an information-theoretic one, and the distinction matters only in that no amount of care rescues it; a separate stand that injected *true* moments after layer `K` found shallow injections nearly worthless (`K=1`: `×1.00–1.05`; `K=24`: `×3.7`), which the rank

curve reproduces; and targeting "the important part of the state" is dead, since our error inside the operator's own top-16 directions is `5.1–7.0%` against an isotropic `6.25%` — **exactly isotropic in the basis that matters.**

The answer, meanwhile, is born early. An exact identity splits every layer's contribution: `E[ReLU(p)] = E[p]·ḡ + c` with `c = Cov(p, gate)` and `ḡ = 0.500` at every layer, so the final mean is a transported sum of the per-layer covariance terms, `m₃₂ = Σ_ℓ transport(c_ℓ)` (seal `5.35e-09`; a linear network has all `c = 0` and the identity returns exactly zero). Measured, layers 0–7 carry **50.1%** of the answer and layers 24–31 only **16.0%**.

⟹ **the answer is born early and the error is made late.** The two measurements are independent — one is a forward decomposition of the target, the other the spectrum of the backward operator — and together they say that the quantity you want is assembled in the shallow layers and then carried through thirty applications of an approximate transport that destroys all but three of its dimensions. That is the shape of the problem, and it explains why "compute the shallow layers more accurately" and "reconstruct the state from the output" both fail, for opposite reasons.

Closed. It also kills the tempting expansion around the linear network. He-initialisation doubles the variance per layer without ReLU, so `2³²` — the expansion point is off by `1.03e5` and the first-order term misses by `2.03e5`. **Rule.** Check the scale of your expansion point before trusting the expansion.

4.3 The error recursion closes exactly, and its budget is known

Writing the per-layer error vector `b_ℓ` and measuring each term in the coherent norm the answer actually uses:

```
b_ℓ = Φ(t_ℓ)·(b_{ℓ−1} @ W_ℓ) + φ(t_ℓ)·δs_ℓ + e_local,ℓ

band                            1..30     1..8    24..30
†  non-closure of the identity 0.00005  0.00002  0.00008
†  linearity-of-transport seal 0.00000  0.00000  0.00000
transport   Φ·δm_p              0.939    0.754    0.969
variance    φ·δs                0.014    0.016    0.016
LOCAL FORMULA error e_local     0.022    0.067    0.017
```

Seal. The seal that validated it: at layer 1 the local share must be **exactly 1** — after whitening the input is exactly Gaussian, so there is nothing yet to accumulate. Measured `1.010`. ⟹ the closure formula accounts for 2.3% of the error, which §3.2 reaches independently by direct injection.

The propagation operator is known from the weights alone. Taking `G_L = diag(Φ_L)·W_Lᵀ` and predicting the next layer's error vector from the current one: `cos` rises `0.735 → 0.976` and `R²` `0.537 → 0.953` with depth; scale factor `1.003`, so no calibration is needed; **Closed.** without the gate `diag(Φ)`, `cos` falls to `0.693`; **Closed.** with a foreign network's operator, `cos = 0.083`.

At depth, **95% of a step's error is the previous step's error, transported.** The closure error is *one object travelling on one shaft*, not 31 independent numbers — which condemns a parameterisation we used for months: four correction families × eight layers, 32 independently fitted scalars, was structurally the wrong shape.

No hot layer exists. The step error *decreases* with depth while transport amplifies later contributions, and the two cancel into a flat profile — the shortest explanation of why the error **saturates** instead of accumulating: `|Δm|/m` is `0.0046%` (L0), `0.526%` (L4), `0.793%` (L12), `0.907%` (L30).

And one corollary is an impossibility result we did not expect. If the closure error is a single trajectory along a *known* linear operator, observed at 30 layers, then it is the textbook setting for a Kalman smoother along depth — thirty noisy views of one path, with a √k gain at zero cost. We measured the assumption before building it:

```
corr( transported sampling noise of layer ℓ , sampling noise of layer ℓ+1 ) = +0.943
                                                  (the filter assumes 0)
realised: Kalman ×1.044–1.051 · symmetric cross-fit ×1.017–1.020
          †  cost of splitting the sample in half alone: ×1.062 ⇒ the split costs more
             than every estimator built on it returns
```

Mechanism. The mechanism: the "noise" at layer `ℓ` is the sampling error of **the same draw**, and it propagates forward through `Φ(t)⊙(Wᵀ·)` in exactly the same way the bias does. **Signal and noise cannot be separated when one operator carries both.** ⇒ the thirty layers are not thirty independent views of the trajectory; they are one view, transported thirty times, and no √k exists. **Lesson.** This also re-reads the observation that started the idea — full recursion `+0.246` against transport alone `+0.242` does not show "the bias is transported", it shows **the operator dominates everything, noise included.**

Rule. The general form, which retired several branches at once: **a correction derived from the sample cannot beat the sample.** The sampling estimate is unbiased and the closure is biased, so a closure correction estimated from the same draw is capped by the draw itself. The closure earns its place for one reason only — it is *independent of the draw* — and that independence is precisely what one pays for.

4.4 The error is a finite-width `1/n` effect — measured three ways

Evaluating the same network at reduced width tests the textbook claim directly. **Warning.** The first measurement lied: truncating `W[:n,:n]` keeps `Var(W) = 2/256` while summing only `n` terms, which does not reduce the width — it changes the **gate regime**. Slopes came out at `−17.5 … −7.2`. After renormalising by `√(256/n)`:

```
n           32     64     96    128    160    192    224    256
rel. error  8.45   4.54   2.52   2.22   1.67   1.38   0.93   0.958 %
local slope    −0.897 −1.100 −0.964 −1.007 −1.011 −1.135 −1.047
†  the two rows are different aggregations and do not recompute from each other
```

Caveat. Read those two rows as what they are, because a reader who divides one into the other will get something else. The error row is the aggregate over the six networks; the slope row is each network's *own* local slope, aggregated afterwards. Taking pairwise slopes of the aggregate row instead gives `−2.6 … +0.2` — the last pair even changes sign, since `0.93` and `0.958` differ by 3% against a between-network scatter an order of magnitude larger. The `1/n` conclusion rests on the per-network statistic, where that scatter divides out; it would not survive on the aggregate row, and we would rather say so than let the table imply otherwise.

Seal. An independent replication on a different construction: *genuinely* narrow networks (`n = 8…32`, depth 32, exact truth) rather than sub-matrices. The object has a lower bound — at `n = 2` the network dies with probability 1, at `n = 8` it survives 97%, and only from `n = 12` is survival certain, so "test it on a small network" has no object below `n = 12`. Above it, the arms `16→24`, `24→32` and `32→256` give `−1.152`, `−1.096`, `−1.132`, with whiskers `[−1.362, −0.735]` on the long arm: covers `−1.0`, excludes `−1.5`.

Caveat. That verdict inverted three times in one evening, always for the same reason — too small a sample at the decisive point. A single-network anchor gave `n^(−1.46)` with an out-of-sample agreement of `×1.005` that looked conclusive; it happened to be the lowest of nine networks. **Rule.** We had written

the caveat ourselves and published the conclusion anyway. **A caveat that is not executed before the conclusion is decoration.** And the constructive half: the question was settled without the new widths we had planned — the power came from **more networks on the arm we already had.**

A third construction — synthetic networks swept across widths `48 … 768` — gives the same answer with a sharper seal, and is reported in §4.6 with the structural facts it belongs to.

⟹ the closure error is not a missing term in a moment expansion; it is the finite-width fluctuation of *this particular* `W`. That retro-predicts most of our dead branches: why a higher-order Mehler series changed nothing, why no residual basis is shared across networks, and why ten low-rank schemes died at the same `×1.0×`.

4.5 Bulk and tail: the gate we now apply to every idea

One piece of exact algebra in the project is not a closure approximation. A bias-free ReLU network is positively homogeneous of degree 1, so Euler gives `out(z) = z·∇out`; for `z ~ N(0,I)` Stein gives `E[z_i g] = E[∂_i g]`. Setting `g = ∂_i out`:

```
E[out] = Σ_i E[z_i ∂_i out] = Σ_i E[∂_i∂_i out] = E[Δ out]
```

`out` is piecewise linear, so `Δout` is a measure supported **on the kinks** — the expectation becomes an integral over the `32 · 256` decision hyperplanes, with no Gaussian assumption anywhere.

```
†  analytically at L=1:  Δ = ‖w‖²δ(w·z) ⟹ E = ‖w‖/√(2π) — exactly the truth
†  numerically (mollified δ):  L=2 ×1.004 · L=4 ×0.996 · L=32 ×1.030, monotone to 1
†  statistically at L=32:  mean(out − Δout)/se = −0.19, −1.90, −0.27, +0.24 ⟹ zero
```

And it is dead in both directions, for a structural reason. Evaluating it needs the **density of the preactivation at zero**. Measuring both functionals from the *same* exact `(m, s)`, so that only sensitivity to non-Gaussianity is compared:

```
layer    error of Gaussian E[ReLU]    error of Gaussian ρ(0)
  0            0.0596%                     0.6531%   ← p exactly Gaussian here
 31            0.1068%                     7.2726%   ⟹ ×68
```

Lesson. At depth `t ≈ 3.3`, so the zero sits 3.3σ out. `E[ReLU(p)] ≈ m_p` is a **bulk** functional and barely sees shape; `ρ(0) = φ(t)/s` is a **tail** functional and is nothing but shape. The identity is exact and converts an insensitive functional into one 68× more sensitive.

Mechanism. What it bought instead is an explanation we had been missing. ReLU activation masks carry real, long-range, non-Gaussian structure — inter-layer co-activation deviates from the orthant-probability prediction by `4.88%` against a `0.28%` noise floor, with five seals including a ×32 sample increase that moved the noise by ×5.7 and the signal by `0.02%`. It correlates with our error at `+0.02`. Masks are `P(p>0)`: a **tail** object. Our error is a **bulk** object. They are orthogonal not through noise but because they are different parts of the same distribution.

Gate: is your idea about the bulk or the tail? Everything near zero — masks, orthant probabilities, densities, gate thresholds — pays the full non-Gaussianity price. Our functional almost does not. That question retro-predicts four dead branches.

4.6 The object itself: twelve structural facts, and the mathematics that does not apply

These are properties of deep random ReLU networks, not of our estimator, and they are the part of this work that outlives the competition.

1 — The error direction is computable from the weights alone, and it is six-dimensional. The 32-layer transport is a product of matrices, so Furstenberg–Oseledets applies. Measured: the participation ratio of the transport operator is `2.0 ± 0.7`, the spectral gap $\sigma_1/\sigma_2 = 2.72$, and $\ln(\sigma_1/\sigma_{256}) = 48.4$ — a dynamic range of $e^{48} \approx 10^{21}$.

```
share of the final MSE inside the top-6 subspace
   Lyapunov subspace (from weights, no samples)   0.682
   closure's own subspace                         0.699
†  random subspace                                0.023
overlap of the top-6 of M_k with that of M_1:  k=5 0.914 · 10 0.823 · 20 0.628 · 30 0.161
```

⟹ **68% of the final error lives in a 6-dimensional subspace computable with no sampling at all**, and error born before layer ~20 arrives along the same direction. And the gap is *born by multiplication*: each individual factor has $\sigma_1/\sigma_2 = 1.02$, the product has `2.72`. There is no cheap local route to that direction — you must multiply 30 matrices.

2 — The combinatorics collapses with depth, and it describes the part that does not matter. The activation-pattern arrangement shrinks sharply:

layer	switching neurons	mask entropy	chambers	max chamber mass
1	251.9	213.7 bit	2e64	1.2e-04 (= 1/N)
8	134.9	106.7	1.3e32	6.5e-04
16	84.6	66.2	8.4e19	3.8e-03
31	58.0	46.1	7.3e13	7.4e-02

That looked like a lever — a 58^3 tensor instead of 256^3. It is not. The overlap of the error's top-6 subspace with the 32 most *frozen* neurons is `0.295`, with the 32 most *switching* neurons `0.052`, and with 32 random neurons `0.131`. **Lesson.** A frozen neuron passes signal *linearly* and does not damp transport; a switching one ($g \approx 0.5$) halves it, so the Lyapunov product survives through the frozen ones. And decomposing the non-Gaussianity by block: `switching×switching 0.025`, `switching×frozen 0.934`, `frozen×frozen 0.041` — it lives in the *cross* term, so the block tensor buys `2.5×` where `100×` was needed.

3 — Kolmogorov widths are fast for one network and slow for the class. The non-Gaussian residual of a *single* network is nearly rank 2 (rank 2 captures `0.917` of it at layer 31, rank 8 captures `0.975`). Across networks, the class modes take about `1/6` of the norm each — the residuals of different networks are nearly orthogonal.

That one distinction explains about ten of our failures at once. Tucker decompositions, low-rank triangulation, learned models over network groups, James–Stein shrinkage, transported coefficients, LOO profiles — every one of them assumed a *shared* basis, which does not exist by construction. **Rule.** Ask not "which basis" but **"are the widths of the object fast, or the widths of the class?"**

4 — The closure is exact at infinite width; everything we fight is an artefact of $n = 256$. A width sweep on synthetic networks ($n = 48 \ldots 768$, depth 16, truth from two independent halves) gives an exponent of `1.11` for the one-step error. **Seal.** The seal that validates it: at layer 1 the accumulated error is *width-independent* ($\alpha = 0.02$) and equals the noise of the truth itself — because the closure is exact there (arccosine kernel). **Seal.** The truth's noise is flat across widths, so what decays is the signal, not the instrument.

⟹ **a correction of order $O(1)$ does not exist.** Combined with §4.4's two other constructions, three independent experiments — synthetic width sweeps, genuinely narrow networks, and renormalised sub-matrices — agree on the same exponent.

5 — A hybrid scheme's ceiling is fixed by one profile. For any method of the form "analytic anchor at layer `ℓ`, sampling for the rest", the variance cannot fall below `1 − ρ²(ℓ)`, where `ρ²` is the explanatory power of a linear head on layer `ℓ`'s activations (head fitted on one half, measured on the other):

```
ℓ        1      2      4      8     12     16     20     24     28     30
ρ²     0.496  0.589  0.690  0.792  0.861  0.912  0.956  0.981  0.996  0.999
cost   0.043  0.086  0.171  0.327  0.455  0.575  0.686  0.792  0.893  0.943
```

⟹ an anchor at layer 8 is capped at `4.8×`, at layer 16 at `11×`, at layer 24 at `54×`. Our own submissions anchor at layers 1–8, i.e. inside a `4.8×` ceiling while already achieving `3.4×`. **Seal.** The full Giles telescope was measured too, with both controls landing exactly where they must (a single level gives `1.0000×`, a foreign network `0.26×`): a pre-registered ladder gives `0.961× ± 0.117`, and the best ladder found by search gives `1.480×` — chosen on the same data, so an overestimate. **MLMC over depth is below our plain estimator.**

6 — The answer arrives in the first forward pass, and iteration destroys what is left. Feeding the output back into the network — a natural "is there more structure?" probe — converges immediately:

```
passes    cos(output_k , E[output])      foreign network   cos between two samples
0 (input z)      0.00003 †  seal            0.001              −0.0001
1                0.98852                    0.323               0.977
40 (plateau)     0.98952                    0.324               1.00000
```

One pass reaches `0.9885`; forty more add `0.001`. **All samples already look almost the same way — which is exactly why the common direction carries no information.** The answer lives in the 2.3% by which they differ, and iteration is the operation that removes it.

Seal. The best self-consistency check in the project falls out of this: `cos = 0.9885` means a transverse component of `15.11%`, and pure averaging over our sample count gives `15.11%/√8192 ≈ 0.17%` — against our measured working error of `0.177%`. The estimator is doing exactly what the geometry says it should, and nothing more.

7 — And a theorem that explains why per-layer scalars are always absorbed. The closure map is strictly positively 1-homogeneous (seal `3.1e-13`), so a scale perturbation introduced at *any* layer collapses into a single multiplier at the output. The global scale constant therefore sees it entirely, and **any correction expressed in the dimensionless variables `(t, ρ)` cannot compete with that constant by construction.** This is why 32 per-layer multipliers commute into one, why an oracle per-layer scalar on `m` gives `×3.26` before the scale fit and `×1.053` after it, and why we should have stopped proposing per-layer scalars after the first measurement rather than the fourth.

8 — There is no hidden conserved structure, and we tested that the way physics does. Before building any apparatus that presumes latent structure, we measured the *level statistics* of the layer-to-layer Jacobian spectrum — the adjacent-gap ratio `⟨r⟩`, which distinguishes universality classes and is insensitive to everything else:

```
depth        1 layer  →  32 layers          reference values
⟨r⟩           0.5294  →   0.5357           GOE 0.5359 · GUE 0.6027 · Poisson 0.386
log σ range      6.8  →     42.1
```

Flat across depth, and on the GOE value to three digits. A hidden conserved quantity — anything that would let the system be block-diagonalised in an unknown basis — would drive `⟨r⟩` toward the Poisson value `0.386`. It does not move. Depth changes the *scale* of the spectrum by six orders of magnitude and leaves the local statistics exactly where random matrix theory puts them. **Lesson.** This also refuted our

own hypothesis that "depth crystallises structure", and it is the second independent confirmation, by a completely different instrument, that the function is *worse* than a uniformly high-dimensional one for the purpose of finding exploitable structure.

Rule. The transferable part: **level statistics is a cheap, assumption-light detector for "is there hidden structure to exploit here?"** — and it can return "no" before you build the machine that would have exploited it.

9 — Low rank of the values is not low entropy of the gates, and the difference is exponential. Two of our measurements looked contradictory for a long time. The participation ratio of the activation covariance collapses with depth (`19.6 → 5.9 → 3.2 → 2.3` at layers 8/16/24/28), which invites a low-dimensional method. The chamber count does not collapse anywhere near as far. Both are true, and they are about different objects:

Caveat. This report quotes that participation ratio from three stands — `19.6 → 2.3` here, `165.6 → 2.8` in §1, `159.0 → 3.8` in §1.1 — differing by up to `×1.7` at matched depth, because they use different networks, sample sizes and layer grids. Nothing below turns on the value; the claim is the *collapse*, which all three show, and we flag the spread rather than silently quoting whichever number suits the sentence.

```
oracle projection of a^k onto its top-r directions, then run the rest of the network
   r = 3   (any layer)      2e-04 … 3.7e-03      50–1000× worse than plain sampling
   r = 20  (layer 28)       9.4e-06              still 2× worse
†  cumulative variance in the top 3: 0.288 / 0.535 / 0.686 / 0.768 — the PR is a mirage
gate patterns among 20 000 samples: 20 000 distinct — at EVERY layer, including layer 31
   (layer 31 still has 65 neurons with 0.02 < p_on < 0.98 ⇒ 2⁶⁵ cells)
```

The tail of small eigenvalues, carrying 25–30% of the variance, is what drives the answer — because it passes through nonlinear layers, where a direction's importance is not its variance. And a ~20-dimensional manifold cut by 65+ hyperplanes still yields 2^{65} cells. **The continuous structure collapses exactly where the discrete structure becomes exponentially rich.** Low-rank methods and gate-caching methods die from the same root, approached from opposite sides.

The mechanism, derived rather than guessed, and it explains every low-rank failure we had. Multilinearity of cumulants gives the third cumulant of a preactivation exactly: $\kappa_3(h_i) = \Sigma_{rst}\, M_{rst} \cdot G_{ir}\, G_{is}\, G_{it}$ with $G = W^{\top}B$. **Seal.** At full numerical rank the formula reproduces the measured cumulant with correlation and scale both `1.00000000`. Truncating `R` then prices the entire low-rank family:

```
R           3      8     32     truth
gain      ×1.19  ×1.51  ×3.08   ×10.46      ⇒ slope ×1.364 per doubling of R
(at R = 32 the cumulant itself already correlates 0.9978 with the truth)
```

An effective rank of ~3 is measuring where the *variance* is, and the non-Gaussianity lives in the *tail* of the spectrum — the dominant components are sums of many contributions and are therefore nearly Gaussian by the central limit theorem, so exactly the directions a low-rank method keeps are the ones with nothing to say. Reproducing the cumulant to four digits still leaves two-thirds of the available gain on the table.

10 — Why plain sampling is so hard to beat: the network does the dimension reduction for you. The 256-dimensional input is nowhere near covered by any affordable sample — nearest- neighbour spacing ratio `1.14` at layer 1, effective dimension `165 → 46 → 7 → 2` with depth. But the estimator never needs the input covered; it needs the *scalar* projection `h = w·a` covered, and that is covered to **99.5%**, with the uncovered tail worth `0.0 – 0.66%` of the mean.

⟹ **there is no under-sampled region to exploit**, which is why every importance-sampling, stratification and resonance scheme we tried returned zero: stratifying the dominant direction `+0.1%`, importance sampling along a weight direction `+0%` (32 ReLUs smear the receptive field; a resonant direction exists only at layer 1), conditioning on the sign `−4%` at the oracle ceiling. And the exact identity behind the last one: the fraction of negative preactivations equals the fraction of zeros after ReLU, `≈50%` per layer, exactly — true, and therefore empty. **The uncertainty is in the magnitude of the tail, not in the sign.**

11 — The answer is a pure covariance between the realised map and the input, and that identity kills the family it suggests. For a fixed `z` the network *is* a linear map: `out(z) = z @ M_{D(z)}` with `M_D = W₀D₀W₁D₁⋯W₃₁D₃₁` and `D_i = diag(1[p_i > 0])`. Since `E[z] = 0`,

```
E[out] = M̄·E[z] + E[(M_D − M̄)z] = 0 + E[(M_D − M̄)z]
```

⟹ **the entire answer is the covariance between which network you got and what the input was.** The mean map `M̄` contributes exactly nothing — which is precisely why the mean-field operator `Φ∘W` works as an error-transport operator and never worked as an answer. Seals: reconstructing `out` from the masks matches the direct forward pass **bitwise**; shuffling the gates against the inputs drives the estimate to `0.96` of its own noise floor; a calibration with all gates open returns rank `1.000`.

The construction this suggests — represent the ensemble by `k` virtual linear networks — is dead by measurement. The ensemble has effective rank `≥343` and does not saturate, and a *linear* rank-`k` truncation with the modes fitted on one half of the sample and evaluated on the other reaches `48.4%` of the answer at `k = 64` against `1.6%` for a random subspace (a real concentration, `×30`) — but on the cosine error, which is the metric that decides, `k = 512` still leaves `8.4%` against the `0.53%` we need. **Seal.** The exact ensemble rank without sketching is `1034`.

Mechanism. And the structurally important by-product: the state collapses and the ensemble does not. They are different objects.

	layer 4	8	16	24	31	
ensemble rank of M	782	664	405	277	201	×3.9 over 27 layers
state rank Cov(h)	–	165.6	9.5	–	2.8	×59

⟹ roughly ten low-rank schemes in this project were justified by the rank of the **state** (`2.8`), when what has to be represented is the **ensemble of maps**. The state collapses because every input is driven into the same direction; *which map took you there* varies as much as ever. **And criticality generates both ranks, which is the cleanest evidence that one cause is under both.** A separate sweep of the *ensemble* rank against the negative-weight fraction drops it from `200.3` to `9.6` when a quarter of the minus signs are flipped; the sweep of the *state* rank in §1.1 falls the same way on the same axis (`159.0 → 44.5` at layer 1). Two different objects, two different stands, one driver. **Caveat.** We keep them labelled throughout, because conflating them is precisely the error this fact exists to correct — and an earlier draft of this very paragraph cited the state sweep as though it were the ensemble one.

12 — Our closure propagates a product of means; the network delivers a mean of products, and the 36% gap between them is entirely inter-layer gate correlation. The two-moment closure transports state with `B = W₀·diag(Φ₀)·W₁·diag(Φ₁)⋯`, replacing each gate by its marginal firing probability. The ensemble gives `M̄ = E[W₀D₀W₁D₁⋯]`. The difference is exactly the accumulated correlation between gates at different layers:

```
layer                    0      1      2      4      8     16     24     31
‖M − M̄‖/‖M̄‖          1.00   1.52   1.99   2.89   4.49   7.12   9.55  10.26
gap, M̄ vs product of means
                       0.0000 0.0554 0.0850 0.1340 0.2127 0.3117 0.3785 0.4077
extrapolated N → ∞       −   0.0503 0.0778 0.1225 0.1982 0.2836 0.3393 0.3606
```

The gap **saturates** (`0.339` at layer 24 against `0.361` at 31), like everything else here, and is zero at layer 0 *analytically* — with one factor, the mean of the product and the product of means coincide by definition.

The seal that decides the interpretation is a scaling exponent, not a value. The control shuffles the masks layer by layer: same marginals, inter-layer correlation destroyed. But at finite `N` independent permutations leave a residual correlation of order `1/√N`, so the shuffled gap *cannot* reach zero, and the verdict must be carried by how it scales:

```
N 2048 → 8192 (×4)          exponent     reading
shuffled gap  0.2713 → 0.1456   −0.449    layers 1–4 give exactly −0.50 ⇒ pure noise
real gap      0.4077 → 0.3729   −0.064    holds ⇒ the control decays, the object does not
```

⇒ the true shuffled gap is **zero**, so the gap is entirely inter-layer gate correlation. **Seal.** An unplanned seal from the same run: at **layer 0** the gap computed through the closure's own `Φ` also decays at `−0.507`, i.e. is pure noise — confirming that the closure is exact at layer 0 and that the instrument sees what it should. **Seal.** The empirical `Φ` at layer 0 is `0.50005` against a theoretical `0.5`.

The number worth carrying: `96.1%` of the gap is correlation, not error in the marginals. At layer 31 the gap computed with the closure's `Φ` is `0.3805` against `0.3729` with the empirical `Φ`, so the error in our own firing probabilities contributes only `0.0754` in quadrature. Our marginal gate probabilities are essentially right; what is wrong is the assumption that gates at different layers are **independent**. This parallels — as a theme, not as a numerical coincidence — the finding that `96%` of the non-Gaussianity comes from dependence between neurons (§4.4): both times the bottleneck is the coupling, not the marginal.

The limit of this fact, which must be quoted with it: `E[out]` does not depend on `M̄` at all, by the identity above. The `36%` is therefore a diagnostic of ensemble structure and **not a lever**; reading it as "our closure is 36% wrong" would be incorrect.

And the machinery that has no hook here. We checked each against the object rather than against intuition, and the reasons differ:

```
Gutzwiller trace formula   needs PERIODIC orbits; we have 32 steps and no return
Maslov / Morse index       bookkeeping over a Lagrangian structure the map lacks
scattering / S-matrix      needs unitarity; ReLU discards half the space, no poles
shadowing lemma            closure error is PER-NEURON; a per-layer constant fails
renormalisation group      would only NAME a measured number: the decay rate of
                           irrelevant directions IS the spectral gap 2.72
Stein / Berry–Esseen       bounds the error at N^(−1/2); reality is N^(−1)
                           ⇒ our case is BETTER than the theorem, via a cancellation
                             the bound cannot see
free probability           its known result — dynamical isometry is impossible for
                           ReLU with Gaussian weights — we measured directly:
                           ln(σ₁/σ₂₅₆) = 48
```

Rule. The pattern worth keeping: a framework earns its place by predicting a number we have not yet measured. Three of the seven above reproduce measurements; none predicted one.

4.7 Why per-instance personalisation is unobservable, not merely hard

The ×1.03 ceiling in the abstract's table is the one we fought longest, because the ceiling above it is genuinely large and the signal is genuinely real. Two knobs can be set per network — the blend weight w and a global output scale c — and their oracle values would be worth ×1.02 and ×1.10, on independent axes.

The signal exists and we can see it: the correlation between our computable proxy and the oracle weight is **+0.714 across networks.** And it is worth nothing:

	across networks	within a network	realised
blend weight w	+0.714	−0.830	×1.0002
output scale c	+0.438	−1.000	×0.9977

The correlation reverses sign between the two regimes, and the regime that matters is the one inside a single network — that is the direction in which the constant would have to move. **Mechanism.** The mechanism, and it is a statement about observability rather than about effort: the optimal per-network constant depends on ⟨T, e⟩, a *linear* functional of the realised noise projected onto the truth, while every statistic computable from inside the estimator is *quadratic* in that noise. A quadratic form cannot see the sign of a linear one. **The quantity is unobservable from inside the estimator**, which is why a better predictor, a better model class, or a Bayesian shrinkage rule cannot rescue it — the missing ingredient is the direction T, not the modelling. **Caveat.** The claim is scoped to statistics available to the estimator at prediction time; it is not a statement that the quantity is unknowable in principle, since with T in hand it is immediate. And indeed split-half estimation does not rescue it (correlation −0.002: it estimates the *expected* noise level, not the *realised* one), and a Bayesian shrinkage form, which is the correct functional form and cannot do worse than the constant, captures $\rho^2 = 0.079$ of the ceiling and lands on +0.11% — the same as simply retuning the constant.

Lesson. Two instrument laws were bought here, both expensive:

- **Measure the reliability of your reference before reading a correlation off it.** Our oracle target had reliability 0.774, which caps any attainable ρ at 0.880. Without that number a measured ρ cannot be interpreted at all.
- **A cosmetic parameter dominates the difference of two close quantities.** An early version of this measurement read $\rho = 0.28$ and we closed the branch. The feature and the target had different denominators, and since our prediction and the closure agree to within 0.7%, the $\alpha = -0.0075$ scale constant changed the norm of their difference by 1.6–3.1×. The real ρ was +0.714. The verdict happened to survive, for a different reason — but the number that produced it was wrong, and we would not have known.

5. What actually worked

Six things, in order of contribution. Everything else in this report is a negative result.

whitening the sample batch	×2.06–2.34 in the Monte-Carlo branch
control variate on the one exact expectation	×1.1369 value / 0.1324% cost ⇒ ×10.9
dead-neuron pruning, introduced	3.62e-07 → 2.81e-07 on the grader
κ = 0.9985, one scalar on C	×1.0039 on the grader, at zero cost
per-network scale a_i	×1.003–1.010 on the grader
all_layer_means, previously discarded	×41345 on a secondary score, cost zero

Caveat. Two readings of that table to set straight before it is used. The pruning pair is from an **earlier baseline** — our final submission scores `1.855e-07`, better than both — and it is quoted to size the step, not to locate us. And pruning appears again in §6.2 as a *failure*: introducing the cut (keep a neuron that fired at least once in 256 pilot samples) is the win above, while every attempt to tighten it lost, monotonically (`K=2 ×0.955`, `K=4 ×0.801`, `K=8 ×0.365`), and the one tightening our local bank endorsed lost `×0.929` on the grader. The threshold is already past the true dead fraction — it cuts `17.4%` where `11.2%` are genuinely dead — so there is **no slack in either direction**, and the two entries describe opposite ends of one knob rather than a contradiction.

The reframing that produced the estimator, and it took us twenty-seven sessions. For most of the project the analytic engine and the sampler were *rivals*: both were asked to carry the answer all the way to layer 32, and the analytic one always lost, because its error accumulates while sampling stays unbiased. Every comparison was "engine versus sampler", and the engine kept losing by a factor that no amount of better mathematics closed.

They are not rivals; they are **partners with disjoint competences**. The forward pass has already computed the intermediate activations `a^ℓ` — at zero additional cost — and what is missing is only `E[a^ℓ]`. So use the engine where it is *accurate* (shallow), let the sampler measure how much *this particular draw* deviates there, and transport that correction to the output:

```
μ̂= mean(a^L) − t · βᵀ( mean(a^ℓ) − Ê[a^ℓ] ),     β = Cov(a^ℓ)⁻¹ Cov(a^ℓ, a^L)
anchor layer ℓ     0      4      8     12     16     20     24     28
†  oracle ceiling  0.98   0.80   0.68  0.46   0.35   0.25   0.13   0.06   (variance ratio)
realised, engine   −     0.954   −    0.779  1.334   −      −     2.035
```

The realised curve is a **bell**: shallow anchors have nothing to correct, deep ones have an engine error larger than the sampling error they are meant to measure. And the seal that justified an entire line of work: a Gaussian anchor (`k = 2`) gives `δ = 4.7e-05` at layer 12 while a third-cumulant anchor gives `3.3e-06` — a factor of `14` — so the Gaussian version of this scheme is *dead* and the cumulant machinery is **a condition of the scheme working, not a refinement of it**.

Rule. The generalisable form: **when an exact method and a sampling method both fail at the same task, check whether they fail at different parts of it.** Ours did, and the whole submitted estimator is built on that split.

The control variate is the algorithmic idea worth stating. Layer 1 is the only place in the network where an expectation is known in closed form (`E[h₁] = ‖w‖/√(2π)`, the input being Gaussian by construction). Subtracting its sampled version and adding back the exact value removes the correlated component of the sampling noise — and because every later layer is a deterministic function of `h₁`, that correlation survives all 31 remaining layers.

This is worth generalising: we enumerated **every** place in this problem where a quantity is known without estimation, and the list is finite and short.

```
exact fact                  used as             worth
E[z zᵀ] = I                 whitening           ×2.06–2.34   ✓
E[h₁] = ‖w‖/√(2π)            control variate     ×1.14, ×10.9 payback   ✓
E‖z‖ = 15.984382667         sphericalisation    ×0.74
E[(u·z)⁴] = 3‖u‖⁴           4th-moment fix      ×1.00
out(cz) = c·out(z)          †  seal 7.45e-15, exact factorisation
E[out] = E[Δout]            †  exact (§4.5), dead in the tail
```

Two of the first four fail the same way. Sphericalisation is exact — `out(z) = ‖z‖·out(ẑ)` with `‖z‖ ⊥ ẑ` — and worth `×1.13` alone, but `×0.74` on top of whitening, which already fixes all `32 896` second moments including the radial one. Fourth moments must, by elimination, govern the residual sampling error; the excess is real and twice the statistical level, and carries no information: `corr = +0.017` against a **foreign-sample control at `+0.055`** — the control exceeds the signal.

A new point generator competes with whitening, not with plain Monte-Carlo. Three ideas (RQMC, sphericalisation, fourth-moment correction) each looked strong against naive sampling and were each already contained in the whitening step.

And the test that closed the whole family at once. Rather than keep proposing sample features, we asked whether a *shared* signal exists: ten networks through sixty **identical** sample batches. If a batch is intrinsically bad, it must be bad for everyone.

```
signed error, correlation across networks, shared batches   +0.0120 ± 0.1459
|error|,      correlation across networks, shared batches   −0.0004 ± 0.1044
†  control: independent batches                             +0.0103 ± 0.1367  ← identical
```

⟹ **there is no universal sample quality.** One measurement replaced an unbounded search and retired three branches.

Two things about κ are worth reporting even though it is one number. First, it broke our own taxonomy: we had classified every modification as a *channel correction* (never above `×1.13` in a hundred attempts) or a *global replacement* (never below `×3.1`), and `κ` is neither — a global scalar applied inside the recursion, i.e. a knob on the dynamics. Second, **its motivation was wrong and the result still holds**: the closure understates `σ` monotonically with depth (`−0.08%` at L1, `−10.56%` at L31), so we expected a multiplier above 1; the optimum is below 1. What is repaired is not `σ` but the balance (§4.1) — the fifth time damping inverted the expected sign.

Seal. Its calibration was checked at the largest scale in the project: a 5×5 grid over `(κ, γ)` on **all 1000 networks** of the public split with exact truth. `κ = 0.9985` is the optimum; the optimum chosen on the first 50 networks transfers to the remaining 950 at **`×1.00000`**; and across 20 disjoint blocks of 50, none moves more than one grid step. Our constants are not overfitted to the visible half — measured, not argued.

One observation we should have made in session 1 and made in session 100. For a prediction `a·m` against truth `T`, `min_a ‖a·m − T‖² = ‖T‖²(1 − cos²(m,T))`. **The scored quantity, after the optimal scale, is literally a cosine loss:** magnitude is free, only direction is paid for. **Rule.** Measure every correction to `m` on the cosine error, not the raw error — the gap between them is `×1.9`, and inside it sits exactly what the global scale constant already takes for free. Three branches passed LOO, A/B cross-validation *and* the foreign control while delivering nothing, because none of the three catches a scale-only effect.

6. What we got wrong

Our corpus holds 396 recorded findings. **113 carry a correction to something we had concluded earlier, and 51 record an outright refutation of one of our own results.** That is the rate this protocol produces when every claim must name the stand that measured it and every stand must reproduce a known number first. A project of this length reporting no retractions has not been checking.

6.1 Six ways a measurement lies, with counts

The individual corrections matter less than the fact that they cluster into six mechanisms, each of which produced several independent "findings" before being caught.

1 — A statistic whose null distribution equals the result you are claiming. We reported "the gates are open with probability 0.47, so they behave like coins". The distribution is *bimodal*: 75–81% of neurons at layer 31 are pinned (`p < 0.02` or `p > 0.98`). The mean-gate statistic has a null distribution of exactly `0.5` and **zero resolving power** between "coins" and "symmetrically pinned". Three conclusions rested on it; one downstream quantity was wrong by ×40, and an expansion had been anchored where its parameter is *maximal* rather than small.

2 — A tautology re-discovered as four different insights. `E[z] = 0` implies anything linear in `z` has zero mean, and antithetic pairs take that bitwise. We wrote it up four times in four vocabularies. **Seal.** The control that settles it: `tanh` gives `2.3e-34` and `sin` gives `1.8e-34` — strongly nonlinear networks produce the same bitwise zero, so the statement carries no information about ReLU.

3 — Regressing a target on a feature that is identically part of it. We reported `R² = 0.444` for activation gates as predictors of the output, but `a_j = h_j·g_j` identically. **Seal.** A run over 256 statistically independent neurons — zero cross-information by construction — reproduces `R² = +0.389`; with each neuron's own gate excluded, `R² = −0.003`.

4 — Reading the norm column instead of the error column. A substitution test printed `‖E_sub‖/‖E‖ = 0.9283`, read as "nothing happened". The error column read `‖ΔE‖/‖E‖ = 0.4394` — the answer had *rotated* by 25.4° while keeping its length. The tolerance is `0.044%`, so the perturbation was 1004× over budget.

5 — An instrument that printed its own noise floor, unread. One stand included layer 0, where the analytic answer is exact by construction and the true bias is identically zero. It printed `1.98e-07`. That is the stand's resolution at 400k samples — and three downstream stands operated in the range `1e-6 … 1e-5` without ever subtracting it. The fix was one index in a tuple. **The refutation of several weeks of work was already sitting in our own logs, formatted and printed.**

6 — A calibration constant obtained by dividing two other numbers. Four occurrences, each surviving weeks because the result was plausible. One made every MSE we quoted too small by `×4.61`; one inflated a cost estimate by `×7`; one invented a variance-reduction factor of `×3.7` where the direct measurement gives `×1.28`.

Five of the six are caught by one discipline: make every stand reproduce a known number, at both ends of its range, before it is allowed to produce a new one. The sixth is caught by never deriving a constant you can measure.

6.2 The retractions that changed a conclusion

claim	what killed it
"γ_4 only switches on at exact moments"	Intermediate points show it turns on **gradually** (`×1.10 → ×2.39 → ×4.50`). Our claim came from two endpoints with nothing between them.
"Non-uniform sample allocation gives +7% free"	In-sample artefact. Four checks: the seal we built was empty by construction; the aggregate is separable, so the optimum is uniform *structurally*; 67% of the spread is realisation noise; cross-seed the branch is worse than uniform.
"Richardson extrapolation in width works (`×1.22`)"	The gain was **pure scale**, which the global constant already absorbs. Cosine error `×0.9999`. LOO, A/B and the foreign control **all passed** while the true gain was zero.
"RQMC beats Monte Carlo by 14%"	24 networks × 8 shifts, paired: `×1.0156`, sign test **12/24**, median `×0.9897`. The original number was noise.
"A learned correction gives `×1.031`"	Measured against a base using *stale* constants. Joint re-fitting leaves `×1.011…1.024`. Our own rule, violated by us.
"`cos = 0.93` after transport implies a large gain"	The backward operator has effective rank `2.74`, so any two transported vectors look nearly collinear. `cos` after a low-rank projection measures nothing.
"The step error is orthogonal to Edgeworth"	The Edgeworth term had been built from **projection** γ, not per-neuron γ. With the right object, $R^2 = 0.983$.
"A per-layer anchor family is worth `+6.2%`"	The stand reported the **ratio of summed MSE**; the score is a mean over networks, but the MSE distribution is heavy-tailed. Median across 40 networks: `×1.0022`. Half the networks got *worse*, and the in-sample networks — where the coefficients were fitted — read `×0.97`, a sign that only appears when everything rests on a few tails. Dropping the three largest networks turns `×1.08` into `×0.9855`. **Mechanism.** This also resolved a standing tenfold disagreement with the grader, which had measured `+0.6%`: there was never a disagreement, only two different statistics. The submission was cancelled.
"Blending our analytic engine with sampling wins by 17–43%"	Two proxies lying in opposite directions at once. The gain was real on the *all-layers* metric, which is reported but **not scored**; on the final layer — where the engine is weakest — it vanished. And the local cost meter ignored residual wall time, which the engine consumed heavily (`0.36 s` against `0.02 s`), inflating the multiplier from `0.10` to `0.166`. On the real grader the blend was `3.2×` **worse**. The analysis was internally correct and aimed at the wrong target.
"The estimator collapses at width 1024, so branch the code by width"	The benchmark scales budget as `4×/4×` depth-and-budget, so the invariant is the *sample count* ($\propto \mathtt{depth}\cdot\mathtt{width}^2$), i.e. `×16` and not `×4`. At the correct scaling the **unmodified** estimator gives `2.2167×` and the fix gives `1.5112×`. We had solved a regime that does not exist.
"Per-network blend weight has no usable signal (ρ = `0.28`)"	The feature and the target were normalised by different denominators, and since our prediction and the closure agree to `0.7%`, a `0.0075` scale constant changed the norm of their difference by `1.6–3.1×`. Real $\rho = +0.714$. The verdict survived for a different reason (§4.7) — but the number behind it was wrong.
"The apparatus has a bias floor of `≈3e-07`"	Frozen constants, not bias. Re-fitting honestly at each budget removes the floor entirely: at `×64` the optimal analytic weight is `0` and the method is simply Monte-Carlo (§3.6).

claim	what killed it
"Even a perfect two-moment oracle is ×53 from the podium"	Two errors compounding: we compared our *adjusted score* against another entry's *final-layer MSE*, and used board figures from before the re-scoring. The correct figure is ×7.7 from #2. The qualitative conclusion survives; the magnitude was wrong by an order of magnitude and had propagated into three planning documents.

And one retraction we are making without a replacement measurement, because the honest form of it is an open question rather than a result. We closed the spherical branch — using the exact radial factorisation `E[out] = E‖z‖ · E_{u~S}[out(u)]`, which holds exactly by positive homogeneity (seal on the identity: `6.6e-17`; `E‖z‖ = 15.984382667`) — on the grounds that it does not survive contact with the rest of our sampler. The measurement was correct and we stand by the numbers: plain Monte Carlo `7.1988e-06`, sphericalisation alone `6.3906e-06` (`×1.13`), whitening alone `3.4933e-06` (`×2.06`), **both together `4.7092e-06` — i.e. `×0.74`, actively harmful**; on the grader, `×0.8081` with a sign test of `15/50`.

The verdict we drew from it — "a new point generator competes not against plain Monte Carlo but against whitening" — was then applied as a general law, and that is the part we retract. What the experiment actually shows is that *sphericalisation composed with whitening* destroys itself: whitening rewrites the sample's second moments, and normalising onto the sphere afterwards breaks what whitening just fixed. It says nothing about a point set that is spherical **by construction** rather than by post-hoc normalisation of Gaussian draws. Whitening corrects one degree per direction pair — the second moment. A construction that fixed all spherical harmonics through some degree `k > 2` exactly is a different object, and we never tested one. ⟹ **we do not claim this class is closed**, and we flag it as the weakest closure in this report. **Rule.** The general lesson stands and is now sharper: a law extracted from one pairing of two components describes the *pairing*, not either component.

And one pattern behind several: a branch closed "because it is too expensive" reopens whenever cost estimates move. One correction was resurrected three times purely because its death had been recorded as a circumstance. **Close branches with a ceiling and a pre-registered threshold instead** — cost is a moving target, a ceiling is not. When the cost of a tensor we had dismissed fell by six orders of magnitude, the ceiling was still `×1.02` and the branch stayed dead.

7. What the numbers exclude about the top of the leaderboard

From the public submission panel of a leading entry and our own measurements:

```
1. NOT moment-based    a full (m,C,γ₃,γ₄) oracle gives 8.16e-09 by direct injection
                       or 1.48e-08 by the sieve (§3.1) ⟹ ×5.4…×9.9 above #1.
                       Against #2 the two instruments straddle parity (×1.13 / ×2.1),
                       so this row excludes #1 only — we do not claim it excludes #2
2. NOT layer-wise      their per-layer[0] error equals the ground-truth sampling floor
                       (6.77e-10), while layers 1..30 are not computed at all
3. NOT pair-coordinate the information ceiling in (t_i, t_j, ρ, depth) is 0.795; 0.992
                       is required
4. DETERMINISTIC       vs-sampling ratio 1782×, per-network spread ×9.6 (ours: 88.5%)
5. CAN AFFORD ALGEBRA  2.61e10 effective FLOPs = 15.7× our closure, 10× headroom unused
```

Point 2 is measured rather than inferred, and it turns on keeping **two different noise floors** apart. The dataset's own truth is Monte-Carlo with `n_samples = 10⁹`, so every layer has a floor equal to that layer's variance over `10⁹` — and the network freezes with depth, so the floors differ by more than an order of magnitude:

```
layer 0   Var(ReLU(W₀ᵀz))/10⁹ = 6.83e-10      †  measured directly on the dataset: 6.87e-10
layer 31  avg_variance/10⁹    = 4.95e-11
```

Their published layer-0 error is `6.77e-10` — `×0.99` of the layer-0 floor. **It is not their error; it is the truth's own noise**, which means their computation returns an exact layer 0 without traversing the network at all. Against the *final*-layer floor, the separate number, the best entry sits `3.628e-9 / 4.95e-11 = ×73` above it, so the competition has not saturated the problem either. **Caveat.** An earlier draft quoted the final-layer floor and asserted their layer-0 figure equalled it — a mismatch of `×13.7` presented as an identity. The argument is stronger once the floors are separated, because the agreement is then `×0.99` rather than approximate.

We also note, for the record, that the Phase-1 board ranks budget tuning at least as much as method quality: the entry immediately below ours had a final-layer MSE **3.4× better** and a worse adjusted score, purely through the cost multiplier.

8. What remains open

We separate branches closed by a **ceiling** from branches closed by a **circumstance**, because our own history says the second kind comes back.

Closed by a ceiling: the two-moment class (`×3.36`), any better step formula (`×1.19`), any better covariance (`×1.10`), per-network personalisation (`×1.03`, and unobservable — §4.7), linear correction in the closure's own basis (`×1.027`), smooth approximation of any kind (`×7–9` at any order, `×3.4` at third), any additive one-dimensional class whatever the basis (`×1.89`, and equal to what polynomials already give), localising the singularities (they are a dense medium with spacing 10^{-4}), state reconstruction from the output (rank `2.74`), sample selection and correction (no common signal exists, and none can — §3.4), multilevel Monte-Carlo over depth (`0.961×` with a pre-registered ladder), finite-size AMP corrections (`×1.25` even with a perfect closure on layers 1–8), the learned-surrogate family (six students, §3.7), the amortised meta-model (no cheap descriptor of an instance exists), and every idea mediated by the density near zero.

Mechanism. One class is closed for a reason worth separating out, because it is about *conditioning* rather than about information: a full fourth-cumulant moment closure **diverges** at criticality. Per-layer measurement shows it more accurate than the third-cumulant version for the first three layers and then exploding — mean `83`, covariance 10^4, $\kappa_4 \sim 10^8$ — via the positive feedback $\kappa_4 \to \mathrm{Cov} \to \kappa_4$. Both natural clamps failed: they destroyed the accurate early layers without stopping the blow-up, which is sudden rather than gradual. A low-rank, trace-extracted variant is stable everywhere precisely because it is self-bounding. And the true kurtosis is quiet — it saturates at $\gamma_4 \approx 0.48$ on this stand, with the worst neuron at `1.2–1.8` — so **the divergence is a badly conditioned method, not physics. Caveat.** Two other stands put the saturating per-neuron γ_4 at `0.385` and `0.395` (the companion tabulates the second), a `23%` spread from the figure quoted here. Nothing in the paragraph turns on which is right — all three are `O(0.4)` against a κ_4 that runs to 10^8 — but we flag the spread rather than quote whichever suits the sentence. The usable form of the idea is the hybrid: fourth cumulant on the first two layers, dropped before the bifurcation, worth `−28%` in that setting.

Closed by a circumstance — a stated reason to revisit:

```
end-to-end learned correction   verdict taken on 12 networks (×1.0004 held-out against
                                ×1.0575 in-sample). The full split carries 1000.
spectral (Wiener) blend         cross-validated ×1.013 against our scalar blend, killed
                                by an eigendecomposition at 0.59% of the budget.
low-rank carrier for C          never measured. If C admits a rank-r carrier where its
                                effective rank is 2.8, ~5% of the budget frees up.
```

And the honest frontier. Our measurements say a great deal about what the top of the board is *not*, and nothing about what it *is*. The one structural hint we have is the factorisation of §5: writing `out(z) = ‖z‖·out(ẑ)`, the radial factor is exactly known, and an oracle on the *angular* factor alone reaches `0.2778%` — **1.83× better than our entire closure** — while an oracle that fixes the direction and keeps only the radius reaches `2.7545%`. The two-moment class spends part of its representational budget on a component that factorises for free. That is not a method; it is a statement that if a better class exists, its state is probably angular rather than moment-based. We did not find it. We can say precisely how much room it would need.

9. If you are starting a problem like this

1. **Measure the ceiling of your class before optimising inside it** (§2), and never measure a component by deleting it when it competes for a shared budget (§2.3).
2. **Evaluate an analytic branch on its own error**, never on a score dominated by a sampling branch. The leverage `d log(score)/d log(precision)` is `0.22` at ×1.5 and `1.07` at ×15 — small improvements are suppressed fivefold exactly where you are deciding whether to pursue them. We discarded ideas for sessions through this.
3. **Before measuring an oracle ceiling, name the channel** that will deliver the finding into the answer. Four of ours had none, and it was visible in advance: an oracle demonstrably helped, every deterministic proxy was orthogonal to it, and the only sample-based source destroyed the mechanism it fed.
4. **Measure the *variation* of a phenomenon across instances, not its magnitude.** We found real long-range structure in the activation masks — 4.88% deviation against a 0.28% noise floor, five seals — and it delivers nothing, because a per-instance correction needs the part that *differs*, and that part was uncorrelated with our error.
5. **A proxy must correlate with the target, and the target is often a ratio.** Our best proxy for the optimal per-network blend weight correlated with the numerator at 0.589 and the denominator at 0.616 — and was useless *for exactly that reason*: in the ratio it cancels to 0.0499.
6. **When several features return zero, stop testing features and test whether a common signal exists** (§5). One measurement replaced an unbounded search.
7. **Size your parameter family against the number of instances you can validate on.** We measured the overfitting penalty directly: `11.76%·k/N`, stable at 9.6–15.4% across two independent runs. At `N = 50` a single fitted scalar already overfits measurably; the workable size was 2–4 parameters, and a proposal for 2209 was dead arithmetically.
8. **Ask whether your functional lives in the bulk or the tail — and at which depth** (§4.5).
9. **When a test lacks power, look for less variance at the point you have** before paying for a new point (§4.4).
10. **Measure noise by redrawing, never by deriving it from `σ/√N`.** That formula is the noise *in the absence of variance-reduction channels*. **Seal.** Without channels we measured `measured/naive = 0.9818`; with them the formula **overstates** the noise by `√g = 1.3–1.65`, so every "signal versus

noise" threshold computed under it looks more generous than it is. Correcting this moved one of our crossover points from layer 14 to layer 5 and closed a branch we had been extending.

11. **Before calling a number precise, measure its noise and its spread separately.** For one central quantity our instrument noise was `2.8e-06` — five significant digits — while the spread *between networks* was `8.1e-03`, a factor of **2861** larger. There was no single number to report, only a distribution, and two earlier "disagreements" between our own sessions turned out to be different slices of it.
12. **When an idea sounds like "disable X so that Y can work", price X and Y in the same units first.** Ours was "disable whitening so the control variates revive": whitening and antithetic pairs are worth `×2.86` together, the entire control-variate ceiling is `×1.50`, so the trade is `×0.52` — dead on inspection, no measurement required.
13. **A lever that needs two accuracies at once costs their product, not their sum.** Our gate channel with an oracle was worth `×1.3140`; with realistic inputs it gave `×1.0025`, *below* its own foreign-feature control at `×1.0165`. The reason is visible only when both are varied: with the true argument an Edgeworth term cuts the error `2.01×`, with the closure's argument only `1.21×`, because **the errors of the shape and of the argument had been cancelling each other.** Price both axes before building.
14. **Two necessary conditions at opposite ends of one axis mean the lever is closed.** We derived a zero-cost formula for the per-layer shrinkage weight whose correlation with the oracle *rises* with depth (`+0.52` at layer 10, `+0.80` at 25 — against `≤0.18` for every hand-designed predictor we had tried). It gains nothing, because the *value* of the channel *falls* with depth. Neither curve is wrong; they simply never overlap.
15. **Verify the ground truth is the truth of the object you are scored on.** For a while our bank computed the truth of the *pruned* network our estimator builds — a target fitted to the method — while the grader measures the full one. The offset was `1.5e-04…6.1e-04`, larger than either noise floor, and it was hiding up to 20% of the MSE. And a small consolation that recurs: **the correct reference was also 30× cheaper** (48 networks in 50 s instead of 25 min). A reference that is expensive *and* subtly wrong usually means the wrong object is being computed.
16. **If the score is a mean, a median improvement does not count.** A deeper analytic anchor won the median on 10 of 16 networks and still finished in a dead heat (`2.429e-06` vs `2.433e-06`), because it also produced fatter tails (`±1.63` against `±1.19`) on the networks where its structure assumption missed. Aggregation rule first, then read the experiment — it decides which statistic you are allowed to improve.
17. **Profile before you optimise mathematics.** One of our engines cost `1.58 s` of unbilled wall time and we spent a session looking for a cheaper algorithm. It was not mathematics: `21 397` tiny array operations paying a fixed per-call tax, with `matmul` and `SVD` not in the top eighteen. **Seal.** The proof is that the cost is *rank-independent* (`1.46 / 1.50 / 1.74 s` at rank `4 / 12 / 24`) — it scales with the *number* of operations, not their size.
18. **Ask not "which basis is better" but "what in this class can be integrated exactly".** Six theories of approximation collapsed into one measurable quantity in five minutes of thinking, and one stand replaced six (§3.3). The reduction is usually available, and it is always cheaper than the six implementations.
19. **Test a theorem at your own parameter, not at its canonical example.** Newman's rational-approximation result is about *one* kink; our function has ~2000 on a single line, and a sweep at `K = 4/16/64/256` shows exactly where the advantage dies.

20. **Before running anything, ask which degree of freedom the idea acts on and who is already standing there.** Applied as a filter to ten proposals — moment problems, randomised linear algebra, juntas, determinantal point processes, tensor trains, QTT, convex-body volume, optimal recovery, hypercontractivity, AMP — it closed nine of them by argument or arithmetic and priced the tenth at `×1.25` with one stand. **Lesson.** The expensive lesson underneath: an importance-sampling tilt we developed over four stands was worth `×1.08` alone and `×0.84` on top of whitening, because both act on the *second moment* and whitening already sets it exactly. **A mechanism that touches an already-corrected degree of freedom does not add — it competes.**

10. Reproducibility

This report is one of two documents. It argues a thesis. The evidence base is the companion volume **The Compendium**: a registry rather than a narrative — the full map of ceilings, all 54 closed classes grouped by what they were trying to repair, the instruments and the seals that caught an instrument rather than confirming it, the methodology laws, all retractions in full, and an index of the 396 findings. Where this report says "measured", the companion says by which stand, on how many networks, with which seal, and what the control returned. Both derive from the same corpus: 109 sessions, 600 measurement stands.

The submitted estimator is documented separately — flow, constants with provenance, FLOP budget, known defects — in a method specification we are glad to supply on request. Every measurement above was produced either by the official grader — with the estimator's blend knob set to isolate a branch — or by stands run against the public `arc-whestbench-public-2026` dataset, whose `final_means` are exact at $N = 10^9$. Ground-truth stands use disjoint seeds for oracle and target where both are sampled.

Numbers we could not seal are marked as derived. Where a measurement contradicted an earlier claim of ours, both appear.