Sampling-Free Estimation of Deep ReLU Activations: A Learned Cumulant-Transport Propagator and Its Measured Ceiling

Keenan Pepper
AE Studio
keenan@ae.studio

Claude Opus 5*
Anthropic

ARC White-Box Estimation Challenge, Phase 1
Algorithmic contribution entry — submission ID **327838**
August 16, 2026

Abstract

The challenge asks, verbatim: *given the weights of a neural network, can you predict its expected per-neuron activations more accurately than running it many times?* Quasi-Monte Carlo is running it many times, so we built the other thing: a fully sampling-free propagator, deterministic in the weights, that carries a mean/second-moment state, an $O(N^2)$ third-cumulant *plane* state, and a recurrent per-neuron latent through all 32 layers, with a learned closure trained end-to-end by backpropagation through the recursion. It is 4.8× under its own analytic Gaussian-closure backbone and scores raw final-layer MSE 1.50×10^{-5}, adjusted 1.95×10^{-6}, on the graded public suite. Our answer to the challenge question is nevertheless *no, not yet*, and the contribution we think is worth more than the artifact is a measurement of why. Four results: (i) the exact $O(N^3)$-state cumulant chain bills 1.4–3.4× the *entire* compute budget, and its $O(N^2)$ plane surrogate reaches only $R^2 \approx 0.50$ deep against the true tensor's 0.93; (ii) the conversion wall is a *conjunction* of correction density and input fidelity, and one leg is removable for free — a dense analytic Edgeworth pair term recovers ∼85% (log-scale) of what a $0.26B$ dense learned head buys at ∼0 cost; (iii) no learned corrector family we tried manufactures chain fidelity, and R^2 dissociates from value in both directions, including a matched-RMS noise arm that destroys a healthy stack 28× at inference time; (iv) with *true* means and covariance injected at every layer the family bottoms out at 8.1×10^{-7} raw — which *would* beat sampling, by 4.7× at equal arithmetic, but is worth only 1.4–2.0× on the challenge's adjusted metric, because both methods already sit near the compute-multiplier floor where only accuracy separates them. The prize for completing this research program is a factor of two, and the walls above say it is not reachable. We also report, as a negative result about our own submission, that the learned attention closure the architecture was designed around is numerically *inert* at this operating point while consuming 61% of the billed cost; deleting it is bit-identical and would have scored 1.50×10^{-6}.

*__Contributions.__ The research program and the estimator are K.P.'s, developed over roughly 110 numbered experiments in collaboration with Claude Opus 5 and Claude Fable 5. This manuscript's text, and the mechanism ablation, cost attribution and lean rebuild of §4–§5, were produced by Claude Opus 5 from the shipped artifact. The challenge rules explicitly endorse the use of language models.

1 Introduction

The task is to predict, for a random bias-free ReLU MLP of width $N = 256$ and depth $L = 32$ with He initialization $W \sim \mathcal{N}(0, 2/N)$ and Gaussian input $x \sim \mathcal{N}(0, I)$, the per-neuron expected post-ReLU activations of the final layer. Predictions are scored

$$s = \underbrace{\tfrac{1}{N}\textstyle\sum_i (\hat{\mu}_i - \mu_i)^2}_{\text{raw MSE}} \times \max(0.1,\ C/B)\,, \qquad C = F + \lambda R, \tag{1}$$

where F is analytically counted FLOPs, R is residual wall-clock time, $\lambda = 10^{11}\,\mathrm{s}^{-1}$, and $B = 2.72 \times 10^{11}$. Spending at most 10% of the budget earns a full 10× discount; exceeding the budget zeroes the prediction. Lower is better.

Two things make this a clean model problem rather than a leaderboard curiosity. First, the multiplier floor at 0.1 means a *biased but cheap* deterministic method competes on even terms with an *unbiased but expensive* sampler — exactly the trade the infinite-width and perturbative-finite-width literatures leave unpriced. Second, the ensemble is the one for which analytic activation statistics are best understood [3–7], so a failure to predict them cheaply is informative about the theory and not just about an implementation.

Analytic moment propagation for ReLU networks is well developed [1, 2, 8], and Wu et al. [9] recently showed that transporting *cumulants* with Hermite expansions estimates the expected output of *sufficiently wide* random MLPs using substantially fewer FLOPs than Monte Carlo, with experiments to depth 12. This paper measures that program at the challenge's own operating point — width 256, **depth 32** — where we find it loses to sampling, and we localise the reason.

Contributions.

1. **The propagator** (submission 327838, §3): a learned–analytic moment-transport recursion carrying $(\mu, M_2, \kappa_{21}\text{-plane}, z)$ and trained end-to-end through all 32 layers. Raw 1.50e−5 / adjusted 1.95e−6 graded; 4.8× under its own analytic backbone.
2. **The κ_{21} plane chain** (§3.3): an $O(N^2)$ state with a factorized ∼4-matmul-per-layer transport, standing in for the $O(N^3)$ third-cumulant tensor whose exact transport bills 1.4–3.4B.
3. **Free dense analytic consumption beats expensive sparse learned consumption** (§3.4).
4. **Low-rank state writes are the only corrector family that converts**, and the converting write is a co-adapted *compensator* rather than a state corrector (§3.5).
5. **The ceiling** (§7): an oracle state would beat sampling at equal arithmetic but is worth only ∼2× on the scored metric, which prices the entire remaining research program.

Everything reported here is measured. Section 4 ablates the shipped artifact and Appendix F maps every number to a script.

2 Setup

Evaluation. Each phase uses a fixed private suite of 100 MLPs, 50 public and 50 sealed, with per-MLP seeds identical across all submissions, and the prize is decided on a fresh re-evaluation with unseen seeds. Only distribution-level accuracy can therefore count. We report four tiers: the

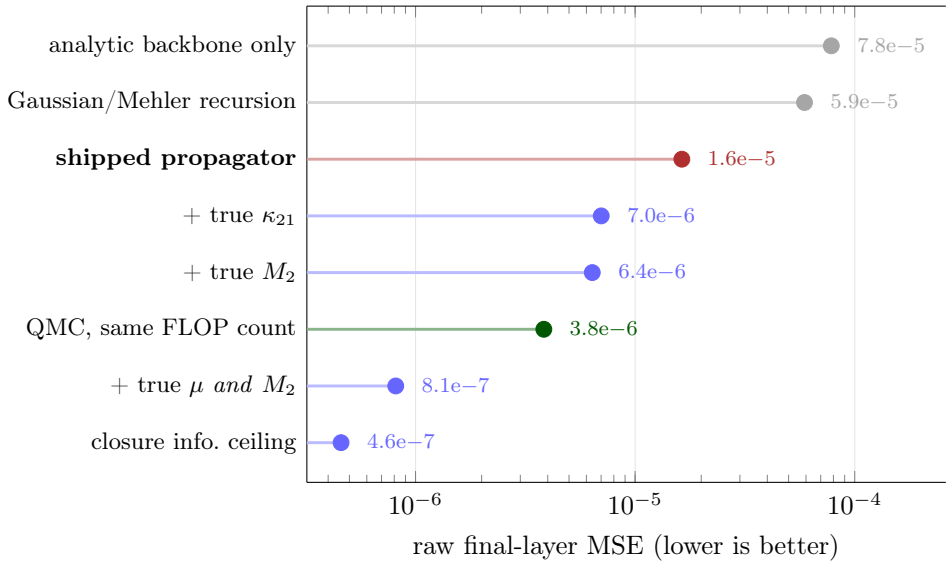

Figure 1: Where the sampling-free propagator sits. Grey: analytic baselines. Red: the shipped submission. Blue: oracle rungs of the *same* state family (rows marked "true ..."), obtained by injecting truth into the running recursion. Green: quasi-Monte Carlo given the same FLOP allowance the propagator spends. The propagator sits well above the sampling line; its own oracle sits well below it. Closing the gap is therefore worth doing and, per §6, out of reach — and per §7 worth only about 2× on the challenge's adjusted metric even if reached. Oracle rungs are measured on a sibling checkpoint of the same stack (Table 5); the analytic-backbone and shipped rows are this submission on `mini`-100.

graded public 50; `full`-split MLPs 0–39 ("bench-40", the tier used for checkpoint selection); `full` 40–139 ("freshval"); and the `mini` split's 100 MLPs, which are seed-disjoint from `full` and were never used for anything. Agreement across them is the generalization evidence:

tier	graded public 50	bench-40 (selection)	freshval 100	`mini` 100
raw MSE	1.50e−5	1.6972e−5	1.6994e−5	1.6325e−5

The selection tier and the two never-selected-on tiers agree to within 4%, so nothing here is a selection artifact.

The budget is the design constraint, and it binds on state size. Only one aspect of the cost model matters for the ideas in this paper, and it matters a great deal: a per-network budget of 2.72×10^{11} FLOPs at $N = 256$, $L = 32$ means an $O(N^2)$ state costs a few percent of budget while an $O(N^3)$ state costs multiples of it. Concretely (§3.3), the faithful third-cumulant tensor's transport bills 1.4–3.4 times the *entire* budget, so the whole architecture is organized around finding an $O(N^2)$ object that stands in for it. Every accuracy number in this paper should be read against that constraint: we are not asking how well cumulant transport can do, but how well it can do inside a budget comparable to a few thousand forward passes. Implementation-level cost engineering — how the challenge's wall-clock term interacts with operation count, and what that rules out — is not part of the argument and is confined to Appendix B.

Baseline. The pure analytic Gaussian/Mehler covariance recursion with no learned component scores 5.91e−5 raw at $\sim 0.008B$. Our shipped backbone, evaluated with every learned mechanism

disabled, scores 7.82e−5 (§4); it is trained to be corrected, not to stand alone.

3 The propagator

3.1 State and one layer step

The estimator runs a single fused recursion over layers $\ell = 0 \dots L-1$, carrying

$$\underbrace{\mu \in \mathbb{R}^N}_{\text{post-ReLU mean}} \quad , \quad \underbrace{M_2 \in \mathbb{R}^{N\times N}}_{\text{post-ReLU 2nd moment}} \quad , \quad \underbrace{K_{21} \in \mathbb{R}^{N\times N}}_{\kappa_3 \text{ plane}}, \quad \underbrace{z \in \mathbb{R}^{N\times 32}}_{\text{recurrent latent}} , \quad \underbrace{k_3 \in \mathbb{R}^N}_{\text{marginal } \kappa_3} . \tag{2}$$

Given the layer weights W, pre-activation moments follow exactly: $\mu' = W^\top \mu$, $\mathbb{E}[zz^\top] = W^\top M_2 W$, whence per-neuron σ_i and correlations ρ_{ij}. Everything after that is a ReLU push, and the ReLU push is where all the difficulty lives.

3.2 The mean map is solved *given* the marginal cumulants

Writing $b_k = \mathbb{E}[\mathrm{ReLU}(v)\, He_k(v)]/k!$ for the Hermite coefficients of the ReLU at the current (μ_i, σ_i), computed on a 481-point Gauss-density trapezoid grid, the Edgeworth mean map is

$$\hat{\mu}_i = b_0^{(i)} + \frac{\gamma_3^{(i)}}{6}\,\mathbb{E}[\mathrm{ReLU}\cdot He_3] + \frac{\gamma_4^{(i)}}{24}\,\mathbb{E}[\mathrm{ReLU}\cdot He_4]. \tag{3}$$

Teacher-forcing *true* marginal γ_3, γ_4 into Eq. 3 while the covariance state free-runs takes the final-layer MSE from 1.055e−4 (Gaussian, $\gamma = 0$) to 2.11e−7 — it recovers 101% of the gap to the free-running floor, and the measured deep marginals are moderate ($|\gamma_3|$ median 0.43, γ_4 median +0.37 at layer 31), comfortably inside the perturbation-friendly regime. So the functional form is not the problem. *Obtaining γ_3, γ_4 at depth is the entire problem*, and it is what forces a third-cumulant state.

3.3 Why the third-cumulant tensor cannot be shipped, and the plane surrogate

The faithful object is $\kappa_3 Z \in \mathbb{R}^{N\times N\times N}$, transported per layer by three dense mode contractions. We costed this exactly. At $N = 256$, $L = 32$: $3.42B$ dense; $1.56B$ using `flopscope`'s verified symmetric-tensor einsum discount (0.405×, probed directly); $1.37B$ with a speculative orbit-restricted source build. No configuration comes within 14× of the budget we are allowed to spend. The obvious structural shortcut also fails: restricting the transport to "live" neurons prunes nothing useful, because the live fraction plateaus at ≥ 0.55 at depth, capping the f^4 lever at ∼5× where 10–45× is needed, and even the mildest pruning costs +109% MSE (Appendix B).

What the closure actually consumes from that tensor is low-dimensional. We therefore carry only the **plane**

$$K_{21}(a,b) \;=\; \kappa_3 Z[a,a,b] \;=\; \mathrm{Cov}(z_a^2, z_b) - 2\,\mathbb{E}[z_a]\,\mathrm{Cov}(z_a, z_b), \tag{4}$$

an (N, N) state, transported by a factorized all-distinct-truncated map that reduces to ∼4 plain N^3 matmuls per layer, with its source terms sliced out of the same Mehler accumulators the covariance step already computes (Appendix A). This is the cost-viable rung: $0.0154B$ instead of 1.4–$3.4B$.

The price is fidelity. Centered off-diagonal R^2 against the true plane, averaged over layers 17–31 and 20 MLPs: full $O(N^3)$ tensor chain **0.930**; plane chain **0.499**; a Gaussian-minted distinct-bulk

Table 1: The conversion wall is a conjunction (raw MSE, bench-40, matched retrains). Columns are plane-feature fidelity; rows are how densely the correction is applied. Only the bottom-right and the free analytic row convert.

correction density	no plane features	plane chain ($R^2 \approx 0.5$)	oracle tensor ($R^2 \approx 0.93$)
learned head, stride 8	2.127e−5	2.125e−5 (1.00×)	1.99e−5 (1.07×)
learned head, stride 1 ($0.26B$)	1.777e−5 (1.20×)	1.710e−5 (1.04×)	**5.95**e−**6** (3.0×)
analytic term, all pairs ($\sim 0B$)	—	2.072e−5 (1.03×)	**7.01**e−**6** (3.04×)

term adds only +0.027 (so it can be skipped, removing the last N^4-class contraction); source-only 0.064. The plane is largely self-transporting — dropping the all-distinct response costs $\sim$0.4 R^2 at depth, decaying linearly rather than collapsing — and critically, its fidelity inside the deployed free-running forward is *identical* to its fidelity when fed true marginals (0.440 vs 0.444 at L=31), so nothing is lost to state drift.

3.4 Two consumers, and a lesson about which one to pay for

Given a plane estimate, something must turn it into a covariance correction. We shipped both of the following, and only one earns its keep.

(a) A learned pair head ("CtxClosure"): 17 per-neuron features → encoder → two attention blocks over the 256 neuron tokens with a per-head $\alpha\rho$ attention bias → a pair MLP over 14 pair features plus symmetric token embeddings, emitting a tanh-bounded, $\sigma_i\sigma_j$-scaled patch scattered into M_2. Evaluating all 32 640 upper-triangular pairs costs $0.26B$, so we evaluate a 1/8 stride with the offset rotating as ℓ mod 8.

(b) A dense analytic Edgeworth term: the leading linear-in-κ_3 covariance correction, applied to *every* pair at *every* layer,

$$e_{21}(i,j) = g \cdot b_2^{(i)} b_1^{(j)} \frac{K_{21}(i,j)}{\mathrm{var}_i\, \sigma_j} \quad \text{(symmetrized)}, \qquad \delta b_0^{(i)} = g \cdot b_3^{(i)} \frac{K_{21}(i,i)}{\sigma_i^3}, \tag{5}$$

behind a single zero-initialized scalar gate g. Cost: $\sim N^2$ per layer, $0.0013B$, essentially free.

The reason both exist is a *conjunction* we had to measure our way out of. Neither correction density nor input fidelity pays alone (Table 1): at stride 8, even oracle-quality plane features are worth 1.07×; at stride 1 with the low-fidelity chain, 1.04×; but stride 1 *and* oracle features together are worth 3.0×. The learned head is starved by sparsity, and the sparsity is forced by its cost. The analytic term dissolves the sparsity leg for free: with oracle features it converts 2.9× *at stride 8*, recovering $\sim$85% on a log scale of what the $0.26B$ dense learned head buys, at 1/200th of its cost.

3.5 Two learned low-rank state writes

The last two mechanisms are learned rank-16 *writes* into the state itself, not into its consumers: a symmetric correction to M_2 (layers $1..L-2$) and a non-symmetric correction to K_{21} (layers ≥ 1), each produced by a small per-neuron MLP over six standardized scalars plus the latent, clamped, and scaled by the local σ's. Each costs $\leq$2.3% of C.

Their motivation is an injection oracle. Free-running the stack and injecting, at every layer, the best rank-r symmetric approximation of the true state residual $E_\ell = M_2^{\text{true}} - M_2^{\text{model}}$ gives a *cliff*, not a ramp (Table 5): $r = 2$ already captures 51% of the full-injection gain, $r = 16$ captures 86%. A mode audit explains it: E is essentially rank 2–5 at depth (top-2 Frobenius share > 0.90

Table 2: Where the budget goes. Measured by source-injected markers gated bit-identical and FLOP-identical to the shipped forward (`paper/attribute.py`); "% of C" includes the challenge's wall-clock term, apportioned as described in Appendix B. Read this table against Table 3: the ordering by cost is almost the reverse of the ordering by value.

mechanism	FLOPs/B	% FLOPs	% of C
CtxClosure pair head (§3.4a)	0.0641	67.8	**61.2**
κ_{21} plane transport (§3.3)	0.0154	16.3	13.9
μ/M_2 recursion	0.0080	8.5	7.0
recurrent latent	0.0027	2.8	5.3
ReLU quadrature (481-pt grid)	0.0013	1.4	2.3
M_2 write (§3.5)	0.0008	0.9	2.3
Mehler-8 off-diagonal	0.0011	1.2	2.0
K_{21} write (§3.5)	0.0006	0.6	1.5
analytic edge term (§3.4b)	0.0001	0.1	1.3
plane-derived features	0.0001	0.1	1.0
marginal κ_3 chain	0.0001	0.1	0.8
Edgeworth mean map	0.0001	0.1	0.6
M_2 assembly, misc.	0.0001	0.1	0.6
total	**0.0944**	100	100

from $\ell \approx 15$), with a paired $\pm\lambda$ spectrum ($\lambda_2 \approx -0.7\lambda_1$) and a top mode substantially $\hat{\mu}$-aligned ($\cos^2 \approx 0.55$–0.64). The model carries roughly the right covariance energy in a slightly wrong direction — a correction that per-neuron factors *can* span.

That predicted, and delivered, the only converting corrector family in the whole program (§6). It also predicted the failures: the κ_{21}-*chain* residual is not low-rank in this way, and every corrector we aimed at the chain converted nothing. **Write site is not input span**: correct the state where its residual is low-rank; do not correct the chain with factors that cannot reach its residual.

3.6 Training and implementation

All learned parts are trained jointly by backpropagation through the free-running 32-layer recursion against final-layer mean targets, batch 8 MLPs, 24k steps, over a 499 000-MLP corpus baked for this purpose (Appendix D). The shipped checkpoint is trained **from scratch** — the long greedy warm-start lineage that produced the architecture turned out to be unnecessary, and a from-scratch run matches its endpoint within 3k steps and then beats it. From-scratch runs are a stability lottery, though: at full learning rate 1 of 5 seeds completed the anneal without a non-finite forward; at 0.6×, 3 of 4 did. Appendix C records the protocol and the divergence anatomy.

4 What each mechanism actually buys

We ablated the shipped weights mechanism by mechanism on the `mini` split's 100 MLPs against exact $N = 10^9$ targets (Table 3). Each arm is an exact no-op edit of the loaded artifact — a gate set to zero, or a write-MLP's output layer zeroed so its rank-16 outer product vanishes identically — so no retraining is involved and nothing is approximated.

Three readings, in increasing order of how much they cost us to learn.

Table 3: Mechanism ablation of the *shipped* weights (`mini`-100, exact targets). "×" is the factor by which raw MSE degrades when the mechanism is removed. "% of C" is from Table 2.

arm	raw MSE	×	% of C
shipped, unmodified	1.6325e−5	1.000	—
analytic backbone only	7.8244e−5	4.793	—
− M_2 write	5.7606e−5	3.529	2.3
− both writes	3.7503e−5	2.297	3.8
− K_{21} write	2.8113e−5	1.722	1.5
− all plane consumers	2.3261e−5	1.425	3.5
− analytic edge term	2.0796e−5	1.274	1.3
− learned γ_3, γ_4	2.0320e−5	1.245	0.6
− latent plane feature t_n	1.8673e−5	1.144	∼1
− marginal κ_3 chain	1.6546e−5	1.014	0.8
− CtxClosure pair head	1.6329e−5	**1.000**	**61.2**
− learned plane pair features (pf)	1.6325e−5	1.000	∼1
− learned plane row features (nf)	1.6325e−5	1.000	(in head)
− ρ attention bias	1.6325e−5	1.000	(in head)

1. The value is in the cheap mechanisms. Everything that matters — two rank-16 state writes, one analytic pair term, learned per-neuron γ's — costs at most 2.3% of the bill each. The propagator's accuracy is not bought with compute.

2. The expensive learned closure is inert (§5). Removing the attention closure entirely changes the MSE by 0.02% while freeing 61% of C.

3. Single-mechanism ablation of a jointly trained closed-loop recursion measures entanglement, not contribution. Read the table against the history and the two disagree violently. The M_2 write, added greedily to a warm stack, bought −12%; *removed* from the from-scratch model it costs 3.53×. And the table is non-monotone in ways no contribution accounting can produce:

- removing *both* writes (2.297×) is far better than removing only the M_2 write (3.529×) — the K_{21} write is actively harmful on an uncorrected covariance state;
- making the whole κ_{21} subsystem inert (1.425×) is better than keeping its consumers but deleting its write (1.722×) — a mis-fed chain is worse than no chain.

Both echo, from the opposite direction, the finding that the converting M_2 write is a co-adapted *compensator*: after training, injecting *true* M_2 into the written stack makes it worse (§6). Ablation tables for recursions like this need both columns — what adding it bought, and what removing it costs — and the gap between them is the interesting quantity.

5 The inert closure, and what we would ship next

The CtxClosure pair head was the centerpiece of the architecture. It is the mechanism that the expressivity gates were run for, that the capacity and data-scaling ladders were run on, and that carries almost all of the learnable parameters. Three independent measurements say it does nothing in the shipped checkpoint:

1. **The trained weights.** The scalar gates that admit plane features into it converged to 0.0039 (pair features) and 0.0133 (row features); the ρ attention bias converged to 4×10^{-5}; the x_{1n} attention bias to exactly 0. The trained output scale is 1.06×10^{-2}, and the patch is tanh-bounded, so its magnitude is at most a few percent of $\sigma_i \sigma_j$.
2. **The ablation.** Zeroing the output scale moves the MSE from 1.63252e−5 to 1.63286e−5.
3. **A bit-identical rebuild.** Deleting the entire subtree — the encoder, both attention blocks, the projection, the pair MLP, and the pf/nf/x_1 features that only fed it — yields predictions *bit-identical* to the zeroed-scale ablation on all 32 layers, since the deleted subtree's only output was a patch that is exactly zero.

Because the rebuild is bit-identical, its accuracy is known exactly and only the bill changes (Table 4). It falls from $C/B = 0.1355$ to 0.0552, which is *under the multiplier floor*. The floor claim does not depend on our residual-wall model: FLOPs alone are $0.0326B$, so even if the residual wall did not improve at all, $C/B = 0.0738 < 0.1$. At the graded raw MSE this would have scored **1.50**e−**6** adjusted rather than 1.95e−6, a 1.30× improvement obtained by deleting code.

Table 4: The lean rebuild. Bit-identical predictions to the shipped model with its closure output scale zeroed; measured cost.

	shipped (327838)	lean	
FLOPs per MLP	$0.0944B$	**0.0326B**	−65%
op events per MLP	14 189	**7803**	−45%
C/B	0.1355	**0.0552**	below the 0.1 floor
raw MSE (`mini`-100)	1.6325e−5	1.6329e−5	+0.02%
adjusted at the graded raw	1.95e−6	**1.50**e−**6**	−23%

Why did it go inert? We think this is a genuine *regime-dependence* result rather than a bug, and we state it as the former with the evidence for and against. For: the same closure family demonstrably converts on the research-grade stack, where the exact $O(N^3)$ chain supplies its features and the error scale is ∼2e−6; there it is worth more than an order of magnitude over the plain recursion (5.91e−5 → 1.96e−6). At the 1.6e−5 error scale reached by a shippable, chain-free state, the stack is bound by other deficits, and Table 1 shows directly that a stride-8 head cannot convert transported pair information at *any* fidelity. Training from scratch then rediscovered exactly that: it drove the plane-feature gates to zero and routed the plane's entire contribution through the free analytic consumer and a single scalar latent feature. Against: we did not run the controlled experiment — retraining with the pair head as the *only* correction mechanism at matched cost — that would separate "inert because superseded by the writes" from "inert because unusable at this stride." We flag that as the missing ablation.

What we would ship next. The lean forward, at the floor, with the $0.045B$ of freed budget spent on accuracy rather than reclaimed: the propagator's error is deterministic in W while a sampler's is scramble noise, and we measured them orthogonal (correlation −0.016 to −0.030 pooled over 40×256 predictions), so an independent-error blend is available. Phase 1 closed before we ran the ablation that found this, so submission 327838 is the shipped artifact and this section is a forecast, not a result.

6 Why it stops: five measured walls

Wall 1 — the conjunction, and what is left of it. Dense consumption is free (§3.4), so the conjunction collapses to a single unknown: chain fidelity. On one stack, one stride, one consumer, R^2 0.5 → 0.93 separates 2.07e−5 from 7.01e−6 — a 2.95× prize for lifting plane fidelity at N^3 cost. The response is smooth in fidelity, with no threshold: mixing α of the oracle tensor into the plane features gives $2.07, 1.56, 1.03, 0.78, 0.70$ ($\times 10^{-5}$) at $\alpha = 0, \frac{1}{4}, \frac{1}{2}, \frac{3}{4}, 1$, near-linear in log MSE through $\alpha = 0.75$. Half-quality fidelity already converts 2×. Depth is uniform: correcting only the shallow half is worth 1.39×, only the deep half 1.49×, and they are sub-additive.

Wall 2 — no learned corrector manufactures fidelity. We built two corrector families to lift the plane chain toward the oracle, both zero-initialized and verified no-ops at init. Per-neuron message passing transported through $W^\top$ and $(W^2)^\top$ reached teacher-forced deep R^2 0.484 → 0.745; pair-native inputs (weight column Grams, the $(W^2h)^\top Wh$ cross-Gram, quadratic transport views) reached 0.735. Frozen-swapping either into the trained stack was *negative* (2.50e−5, 2.53e−5 against 2.09e−5), and value-aligned through-training — corrector trainable, end-to-end mean loss — landed at 1.98e−5 against a 1.56e−5 signal bar. Per-neuron and pair-level inputs land indistinguishably on both axes, so the plateau is **not input-span-limited**. The missing R^2, and all of the value, either is not learnable from any local function of (state, W) under these losses, or lives in components these losses do not seek.

Wall 3 — R^2 and value dissociate, in both directions. This is the result that most changed how we run experiments. A mixture arm with $R^2 = 0.71$ *by construction* converts to 1.56e−5; the learned corrector with $R^2 = 0.745$ converts to 1.98e−5. Same nominal fidelity, opposite verdicts. And an arm whose plane state is the oracle tensor plus fresh Gaussian noise matched per item and layer to the RMS of the plane chain's own error does not merely fail — it takes a healthy 7.0e−6 stack to 1.96e−4, a 28× collapse that is present at step 0 (so it is inference-time closed-loop amplification, not a training failure) and that 3k steps cannot repair, while the plane chain's error *at the same RMS* trains stably. Conversion is governed by error *structure*, not magnitude: deterministic, learnable error is co-adapted around; incoherent error is poison through a closed loop. Corollary for practice: teacher-forced R^2 is not a sufficient gate for any in-loop correction.

Wall 4 — corrections convert only where the residual is low-rank. Restated from §3.5: the M_2 residual is rank 2–5 and $\hat{\mu}$-adjacent, so rank-16 per-neuron-factor writes converted (−12%, the first corrector in the program to convert anything); the κ_{21}-chain residual is not, and identical machinery aimed at it converted nothing. Note also that the M_2 oracle's low ranks *hurt* when injected zero-shot (r=1: 4.06e−5 against a 2.09e−5 baseline) — partial fixes must be trained through, which is the sixth time in this program that a frozen swap understated a learned consumer.

Wall 5 — the converting write is a compensator, not a corrector. Re-running the injection oracle on the *written* stack, every truth injection now hurts, including full true M_2 (2.50e−5 against the 1.82e−5 baseline). Auditing the written state shows its error is *larger* than before the write and concentrated in a single $\hat{\mu}$-aligned mode with eigenvalue $\approx +5$ against $+0.4$ pre-write. The write improves final means by injecting a large, deliberately wrong, $\hat{\mu}$-aligned covariance component. This kills the obvious iteration (re-audit the residual, write against it) and it re-prices eigen-supervised writes downward: supervising toward truth would fight the mean objective. Structure priors belong in the architecture, not in the loss.

7 The ceiling

The five walls say where our error comes from. The ceiling says whether closing them would have been enough. We measured it by injecting truth into the free-running recursion at every layer (Table 5).

Table 5: Oracle rungs: truth injected at every layer of the free-running stack (bench-40, sibling checkpoint of the same architecture, baseline 2.087e−5). "×" is against that baseline.

injected	raw MSE	×	note
rank-1 M_2 residual	4.06e−5	0.51	*hurts*: partial fixes must be trained through
rank-2 M_2 residual	1.353e−5	1.54	51% of the full gain already
rank-16 M_2 residual	8.447e−6	2.47	86%
full true M_2	6.381e−6	3.27	the whole covariance channel
true κ_{21} (via the analytic term)	7.01e−6	2.98	a perfect plane chain
true means only	2.02e−5	1.03	*hurts*
true means and true M_2	**8.127e−7**	**22×**	the joint state-fidelity ceiling
closure information ceiling	4.58e−7	46×	oracle features, this closure family

Two things stand out. First, the interaction: true means alone *hurt*, true covariance alone gives 3.3×, and both together give 22×. Mean-side and covariance-side errors in this recursion are strongly interactive, which is why the program's many single-channel upgrades kept coming out non-additive or anti-useful.

Second, the arithmetic that prices the whole family. Take the challenge's cost model away first and just count arithmetic, since that comparison is metric-independent. Our forward's $0.0944B$ buys 6130 sample forwards (4.19e6 FLOPs each), and quasi-Monte Carlo at $K = 6130$ scores 3.8e−6 raw against our 1.50e−5: the propagator is worth about 1300 samples while costing 6130 samples' worth of FLOPs. **As built, it loses to sampling by 3.9× at equal arithmetic** — or by 1.35× in the lean configuration of §5, which is nearly parity. *With an oracle state it would win*, by 4.7× at the shipped FLOP count and 13.7× at the lean one. The analytic route is not intrinsically dominated; our instantiation of it is.

That accounting charges every sample the 4.19e6 FLOPs of a full-width forward, which is not what a serious sampler pays. Censusing neurons that never fire and forwarding the survivors through the pruned sub-network brings the rate to ≈2.2e6 billed FLOPs per sample — all-in, at deployed sample counts, and agreed to within 3% by two independently built pipelines of ours (Appendix E). By this paper's own taxonomy that is real arithmetic avoided rather than billing arbitrage: it is the one case in §8 where exact structure converts, so the propagator has to face it. Priced that way the same allowance buys 1.9× more samples, and interpolating our measured Sobol curve (local exponent 1.07 in K near these budgets) the comparison moves against us throughout: the shipped deficit widens to 7.9×, **the lean configuration's near-parity becomes a 2.5× loss**, and the oracle's margins move to 2.4× shipped and 7.3× lean. The oracle state still wins comfortably; nothing we actually built does. We quote the unpruned figures above because they are measured directly on both sides, but the pruned rate is the honest one, and it points the same way as the sampling-frontier caveat of Appendix E. Note that it is an *all-in* rate amortized over deployed K: at the lean comparison's much smaller budget the fixed probe and census share is proportionally larger, so this column is if anything generous to the sampler there.

Now put the cost model back, and the picture changes in an instructive way (Table 6). Because both methods can sit at or near the 0.1 multiplier floor, the metric rewards only accuracy, and the

oracle's 18× raw advantage over our shipped propagator converts into just 1.4–2.0× against our own quasi-Monte Carlo submission. **The entire remaining research program — closing all five walls perfectly — is worth about a factor of two on the scored metric.** That is the number we would want a successor to know before starting, and §6 is our evidence that it is not reachable with any corrector family we could construct.

Table 6: Pricing the family. Sampling column is quasi-Monte Carlo given the same FLOP allowance; "adjusted" applies Eq. 1 at each configuration's own multiplier. Our deployed quasi-Monte Carlo submission (Appendix E) scores ≈ 1.6e−7 adjusted. The sampling column is calibrated against *our* scrambled-Sobol estimator, which is not the frontier of that lane (Appendix E); every comparison here is therefore conservative in the propagator's favor.

state	cost point	raw MSE	vs. QMC at equal FLOPs	adjusted
shipped	$0.1355\,C/B$	1.50e−5	0.26×	1.95e−6
shipped	lean, at the floor	1.50e−5	0.74×	1.50e−6
oracle (μ, M_2)	$0.1355\,C/B$	8.13e−7	4.7×	1.10e−7
oracle (μ, M_2)	lean, at the floor	8.13e−7	13.7×	8.13e−8

8 Negative results that shaped the design

Each of the following was a route we costed, gated and closed; we report them because the closures are more reusable than the artifact.

Sampling as an input. Injecting per-layer sampled means and second moments into the recursion, sweeping $K \in \{512, \ldots, 32768\}$ against rank r: *no* (K, r) beats its own sample mean. The anatomy is the finding — at $K = 32$k, injecting sampled means and covariance gives 1.23e−6 against the truth-injected 8.13e−7, so sampling estimates the state well enough; the binding constraint is the recursion's own final-map error, which exceeds quasi-Monte Carlo noise at any K where the samples are good. Once you hold K good samples, the propagator is a *worse* readout of them than averaging. A learned fusion of the final sample mean with the propagator's prediction has an identity floor by construction and does have real margin below it at low K (74% at K=512), but the adjusted-score crossover goes to pure sampling for $K \geq 8192$: fusion pays only with a near-free side model, and a $0.12B$ propagator can never earn its toll.

Learned control variates. A first-layer $\{\mathrm{relu}, |\cdot|, \mathrm{sq}, \mathrm{lin}\}$ dictionary with exact Gaussian means and ridge-fitted coefficients achieves fit $R^2 = 0.22$–0.36 yet a variance-reduction ratio of 0.95–1.06 at every K. The ∼30% of variance a shallow surrogate captures is exactly the variance scrambled Sobol already integrates; the scramble-specific degree-≥ 3 residual that *is* the quasi-Monte Carlo error is orthogonal to closed-form families, which are smooth by construction.

Direct weights-to-means prediction. A metanetwork trained on $(W \to \mu)$ pairs, scanned over the canonical capacity×data quadrant: data helps at fixed capacity (1.4×), capacity helps only with data, best 9.28e−4 at 49M parameters and 15k MLPs. Scaling is alive and the pre-registered gate formally passes, but the level is 55× worse than the analytic stack and the measured slope extrapolates to ∼2e−4 at 10^6 MLPs. **The analytic prior is worth two to three orders of magnitude of data** on this task.

Only *exact* structure converts to billing arbitrage. Two attempts to trade approximate structure for compute both died on coherent bias. Rank-compressing the per-sample forward: retaining 99.7% of eigen-energy still needs rank 100–250 at depth (the participation ratio describes the bulk, not the ϵ-rank), the truncation-bias floor is already 4.7× the full-precision error, and rank contracts only in the last third of the layers so Amdahl caps the sample multiplier at ∼1.5×. Per-neuron James–Stein shrinkage on split scrambles: the *oracle* truth-optimal shrink equals the raw pooled MSE to three digits at every (K, M) — with exactly-dead neurons already pruned, the surviving error sits on neurons whose means dwarf their sampling noise. The pattern across both: approximate structure buys compute only by paying coherent bias, and at these error scales the admissible bias is far smaller than any compression worth having. The one structural exploit that did convert — pruning neurons that provably never fire — works precisely because it is *exact*, introducing no bias at all.

Global low-rank compression of the cumulant tensor. Tucker truncation of $\kappa_3 Z$ after every chain step, with a shared symmetric basis: $R = 64$ retains 95% of the Frobenius energy and is already 2.23× worse; $R = 16$ is 10× worse. The mean-relevant content lives in tail modes that no global decomposition finds — the same signature that killed a spectral-sufficiency probe and an adjoint-metric corrector. Note the contrast with Wall 4: the *residual* object is low-rank and $\hat{\mu}$-adjacent even though the *spread* object is not. Which object you decompose decides whether low rank helps.

Others, briefly. Mixture/cell conditioning of the closure is real zero-shot (+31% on a frozen stack, direction-selective and covariance-dependent) and converted nothing through-trained at this capacity; covariance write ports were null in every arm at every strength, with three architecturally distinct arms agreeing to 3–4 significant digits at all 60 evaluations, which is how we knew the channel and not the design was empty; per-neuron W-readers were null six times for six, though the stride result in Table 1 later showed those nulls were confounded with the consumer's sparsity and say less than we thought.

9 Discussion

What we believe is excluded, for this ensemble. At width 256 and depth 32, bias-free He-initialized ReLU with Gaussian input: cumulant states carried at $O(N^2)$ cost, any consumer of those states we could build, learned correctors of the chain from per-neuron or pair-level local functions of (state, W), global low-rank compressions of the cumulant tensor, per-neuron post-hoc shrinkage, function-space control variates from shallow dictionaries, and sample-assisted hybrids at competitive sample counts. And — the load-bearing one — the 18× of raw accuracy separating our propagator from an oracle version of its own state, which no corrector family we constructed recovered any material fraction of.

What is not excluded. These are empirical bounds, not theorems. The oracle constructions bound families rather than implementations, which is why we lean on them, but the sample sizes are tens of MLPs for the oracles and the fidelity-lifting question is closed only against the corrector architectures and losses we tried. The $O(N^3)$-state chain itself is not excluded on *accuracy* — it reaches R^2 0.93 and converts 3× — only on cost, and cost is a property of this challenge's billing rather than of the mathematics. Above all, the result is stated at one point in the width–depth

plane. Wu et al. [9] beat sampling at sufficiently large width and test to depth 12; we lose at width 256 and depth 32. Whether the crossover is a width effect, a depth effect, or a depth/width effect is the obvious next measurement, and the one we would most like to see: a width ladder at fixed depth would turn our single point into a phase boundary.

The honest answer to the question. Can you predict a deep network's expected per-neuron activations more accurately than running it many times? At this operating point, with the estimator we built: no — sampling wins by 3.9× at equal arithmetic, and by 1.35× even after we delete the 61% of our compute that turned out to be dead — or by 7.9× and 2.5× respectively once samples are priced with the dead-neuron pruning a real sampler uses. The cumulant hierarchy is nonetheless doing real work: true marginal cumulants recover 101% of the gap to the floor, our propagator is 4.8× under its own Gaussian backbone, and an oracle version of its state would beat sampling outright. The barrier is not that moments are the wrong idea. It is that the fidelity needed to exploit them at depth 32 lives in a part of the state we could carry cheaply but could not fill — and every route we found to fill it either costs more than the entire budget, or, as in the compensator result of Wall 5, gets absorbed by a closed loop that has already learned to be wrong in a compensating direction. The state you would need approaches the complexity of a sample cloud, at which point Monte Carlo has been re-derived.

10 Methodology notes

The practices that paid for themselves, offered as transferable practice. **Pre-register kill rules with numbers**; the arm-level early-signal calibration mined from 21 finished runs (every converter is $\geq$2% under its warm reference by step 2000) let us kill dead arms in 18 minutes rather than 3 hours. **Through-train before believing any negative**: frozen substitution understated a learned consumer six times out of six in this program. **Do not trust teacher-forced R^2** as a gate for in-loop corrections (Wall 3). **Use latent-blind and architecture-swapped controls**: three cov-port arms agreeing to four digits was worth more than any single arm's number. And **ablate the shipped artifact**, not the design — we would not otherwise have discovered that 61% of our compute was dead.

Artifacts

Submission **327838**; code and weights at `research/cumulant_transport/submission_v4/` (611 lines, 392k parameters, 1.6 MB). The ablation, cost-attribution and lean-rebuild harnesses used for §4–§5 are in `research/cumulant_transport/paper/` and run in minutes on a laptop against the public dataset. Two training corpora are released: `whestbench-relu-mlp-means-500k` (500k MLPs with per-layer post-ReLU mean targets and pre-activation marginals, seed-regenerable weights) and `whestbench-relu-mlp-moments-10k` (pairwise moments to fourth order at $N = 10^8$).

References

[1] Adel Bibi, Modar Alfadly, and Bernard Ghanem. Analytic expressions for probabilistic moments of PL-DNN with Gaussian input. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9099–9107, 2018.

[2] Jochen Gast and Stefan Roth. Lightweight probabilistic deep networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3369–3378, 2018.

[3] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as Gaussian processes. In *International Conference on Learning Representations*, 2018.

[4] Radford M Neal. Priors for infinite networks. In *Bayesian learning for neural networks*, pages 29–53. Springer, 1996.

[5] Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/148510031349642de5ca0c544f31b2ef-Paper.pdf.

[6] Daniel A. Roberts, Sho Yaida, and Boris Hanin. *The Principles of Deep Learning Theory*. Cambridge University Press, 2022. https://deeplearningtheory.com.

[7] Samuel S. Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL https://openreview.net/forum?id=H1W1UN9gg.

[8] Oren Wright, Yorie Nakahira, and José MF Moura. An analytic solution to covariance propagation in neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 4087–4095. PMLR, 2024.

[9] Wilson Wu, Victor Lecomte, Michael Winer, George Robinson, Jacob Hilton, and Paul Christiano. Estimating the expected output of wide random MLPs more efficiently than sampling. Preprint, Alignment Research Center, 2026.

A The plane transport

Write $h = \mathrm{ReLU}(z)$ for the post-activations at a layer, $g_1^{(i)}$ for the ReLU gain $b_1^{(i)}/\sigma_i$, and let C_{11} and S_{2m} be the Mehler accumulators the covariance step already forms,

$$C_{11}(i,j) = \sum_{k=1}^{8} k!\, b_k^{(i)} b_k^{(j)} \rho_{ij}^k, \qquad S_{2m}(i,j) = \sum_{k=1}^{8} k!\, \tilde{b}_k^{(i)} b_k^{(j)} \rho_{ij}^k, \tag{6}$$

with $\tilde{b}_k$ the Hermite coefficients of ReLU^2. The post-activation plane splits into a diagonal-source part and an all-distinct response part. Truncating the all-distinct bulk to its leading separable term — justified empirically, since a Gaussian mint of that bulk adds only $+0.027$ deep R^2 (§3.3) — gives

$$\tilde{A} = (S_{2m} - 2\, b_0\, C_{11}) + \psi_2\, g_1^\top \odot K_{21}, \qquad \tilde{A} \leftarrow \tilde{A} - \mathrm{diag}(\tilde{A}), \tag{7}$$

$$d = \left(\mathbb{E}[h^3] - 3\, b_0\, \mathbb{E}[h^2] + 2\, b_0^3\right) + g_1^3 \odot \mathrm{diag}(K_{21}), \tag{8}$$

where $\psi_2^{(i)} = (\tilde{b}_2^{(i)} - 2b_0^{(i)} b_2^{(i)})/\sigma_i^2$, and the transport to the next layer with weights W and $U = W \odot W$ is

$$K'_{21} \;=\; \big(U \odot d\big)^\top W \;+\; U^\top \big(\tilde{A} W\big) \;+\; 2\,\big(W \odot \tilde{A} W\big)^\top W. \tag{9}$$

Eq. 9 is four plain $N \times N$ matmuls; a direct implementation of the same map as a dense einsum agrees with it to 5×10^{-16}. The state is carried in fp32: it is consumed only through clamped, standardized features, the transport products are fp32-native, and parity against the fp64 reference was re-gated after the switch (final-layer maximum relative difference 1.5×10^{-4}, against a 10^{-3} gate).

B Cost engineering under the challenge's billing model

None of this is part of the paper's argument, but it determined what we were able to build, and the effects are large enough to mislead anyone who assumes the budget is a FLOP count.

The wall-clock term makes operation *count* a first-class cost. The challenge bills $C = F + \lambda R$ with $\lambda = 10^{11}\,\mathrm{s}^{-1}$, where R is residual wall time, i.e. the part not attributed to the counted arithmetic. Our forward issues 14 189 billed operations per network and spends 0.112 s of residual wall time on the grader, so *30% of our total billed cost is interpreter and dispatch overhead*, at

$$\frac{\lambda R}{\#\mathrm{ops}} = \frac{10^{11} \times 0.112}{14\,189} \approx 7.9 \times 10^5 \text{ FLOP-equivalents per operation.} \tag{10}$$

A 256×256 fp32 matmul costs 3.4×10^7 FLOPs, so the overhead is 2% of a large matmul — but a 256-element `clip` costs 512 FLOPs, where the overhead is 1500× the arithmetic. Because the overhead is roughly fixed per operation, we apportion the measured R across mechanisms by operation count in Table 2; the resulting total, $C/B = 0.1356$, reproduces the grader stack's measured 0.1355.

Fast matrix multiplication is a net loss here. Recursive Strassen multiplication cuts counted FLOPs by 22–27% at every large shape in this forward. It still loses: each call dispatches ∼46 sub-operations and adds ∼0.9 ms of wall time, about 9×10^7 FLOP-equivalents, against a largest-per-site saving of 2.7×10^7. Measured end to end, Strassen at every model-side site gives $C/B = 0.291$ against plain multiplication's 0.116. We use plain multiplication throughout the propagator, and retain Strassen only in the sampling baseline, where at $K \sim 10^5$ a single call saves $\sim 3 \times 10^9$ FLOPs and clears the bar comfortably.

Fusing loops into tensor passes. Reconstructing the Mehler off-diagonal as one batched $(8, N, N)$ pass instead of eight sequential Hermite orders, running the two N^3 products in fp32 (the native weight dtype, so only the product rounds), and skipping feature construction behind exactly-zero gates together took C/B from 0.167 to 0.1355 with no accuracy change beyond floating-point reassociation. Numerical parity against the fp64 reference was re-gated after the change.

Measure under the grader's own installation. The identical forward billed $0.0585B$ under a local copy of nominally the same accounting library and $0.0712B$ under the grader's. Every cost number in this paper is from the grader stack. Two further notes for reproducers: the grader hands

estimators *float32* weights rather than float64, so fp32 intermediates are billed at the base rate and there is no precision-downcast arbitrage; and symmetric-tensor contractions receive a verified $0.405\times$ discount, which is what makes the $O(N^3)$-state costing in §3.3 as favorable as $1.4B$ rather than $3.4B$.

Liveness pruning of the exact cumulant transport. Restricting the third-cumulant transport per layer to neurons with non-negligible firing probability, the fidelity and cost requirements are mutually exclusive by a wide margin. Thresholds 0.002/0.01/0.03/0.1/0.2 give live fractions 0.84/0.79/0.75/0.68/0.63 and MSE ratios $2.09/2.58/3.06/4.85/7.85\times$ against the unpruned reference, with implied total C/B of 0.98/0.82/0.70/0.53/0.40. Reaching a shippable cost needs a live fraction around 0.44, which never occurs at any threshold, and the fidelity has already collapsed long before that. Liveness in this regime is simply not sparse.

C Training protocol

Batch 8 MLPs, 24 000 steps, backpropagation through the full 32-layer free-running recursion against final-layer post-ReLU mean targets; ∼3.5 GPU-hours on one device. Closure learning rate 1.8e−4, latent 6e−4, 500-step warm-up, cosine decay, global multiplier 0.6. Weights come from a 499 000-MLP corpus disjoint from every evaluation tier; 1000 seeds are held out. The shipped checkpoint is the best bench-40 evaluation, which occurred at step 2999 — the remaining 21k steps improved the training loss ($1.29 \to 1.10\text{e}{-5}$) without improving held-out error, and the final-layer training MSE (1.70e−5) matches held-out error, so this is evaluation noise on a 40-MLP tier rather than overfitting. The `mini`-100 and freshval-100 checks in §2 confirm the selection bought nothing spurious.

Divergence anatomy. Failures in this training regime are non-finite *forwards*, never exploding gradients (zero gradient-guard hits across the program). A handful of divergent MLPs would veto an entire batch under batch-level skipping, causing a permanent stall; the fixes that worked were per-sample finite masking with bad-count telemetry, a non-finite-gradient guard, and `nan_to_num` on the write outputs. Learning-rate restarts on a converged basin fail hard: at $3\times$ and $8\times$ there is a deterministic forward-divergence cliff at the ramp (∼step 449) that per-sample masking cannot save. From-scratch stability is a seed lottery — 1 of 5 seeds completed the anneal at full learning rate, 3 of 4 at $0.6\times$ — and 3 of 5 reached within 5% of the final number before dying, so the architecture converges fast and the deep anneal is the fragile part.

D Released datasets

`whestbench-relu-mlp-means-500k`: 500 000 training and 2 000 low-noise validation MLPs of the challenge family, with per-layer post-ReLU mean targets and pre-activation marginal statistics. No weights are stored; every MLP is exactly regenerable from its seed by a spec-portable recipe (inverse-normal-CDF Gaussians from a frozen Cephes implementation, not a framework RNG, whose seed-to-tensor map is not portable), and every record carries a fingerprint for verification. Target noise is ∼1.2e−8 on the train tier ($K = 2^{21}$ quasi-Monte Carlo) and ∼7.5e−10 on the validation tier ($K = 2^{25}$), the latter matching the official evaluation targets. `whestbench-relu-mlp-moments-10k` additionally stores weights and full pairwise moments to fourth order at $N = 10^8$, which is what made the teacher-forced oracle measurements in this paper possible.

E The sampling baseline

For completeness, since it is the thing the challenge question contrasts itself with. A fixed 256-dimensional scrambled Sobol design, generated offline and shipped as a pickle-free array, is forwarded through the network and the per-layer sample means returned; a 2^k prefix of a Sobol net is itself a valid net, so sample count is a runtime knob. Two exact-structure optimizations pay: a short full-width probe prefix censuses neurons that never fire, after which the remaining samples are forwarded through the pruned sub-network (this is the exact-zeros arbitrage of §8), and Strassen recursion is worth using here because at $K \sim 10^5$ a call saves ∼3e9 FLOPs against ∼9e7 FLOP-equivalents of dispatch. This lands at raw ∼1.6e−6, adjusted ∼1.6e−7.

The per-sample rate used in §7. Because the equal-arithmetic comparison turns on what a sample actually costs, we take the rate from metered bills rather than from any scaling-law inference. Two independently built pipelines — different point sets, different probe and census details — agree:

pipeline	billed C/B	points	FLOPs/sample
spherical 5-design (Kerdock), current	0.525	66 048	2.16e6
scrambled Sobol, previous	0.267	32 768	2.22e6

So ≈2.2e6, against 4.19e6 for an unpruned full-width forward. Two caveats attach to this number. It is *all-in* at the deployed sample count: the fixed probe and census cost ($\approx 0.017B$) is amortized over the K above, so at much smaller budgets — such as the lean propagator's $0.0326B$ — the true marginal picture is less favorable to the sampler than the flat rate suggests, and §7's pruned column is correspondingly generous to it. And the rate is not a constant of nature: it tracks the dead-neuron threshold ε, whose optimum has moved down over the program ($0.006 \to 0.0045 \to 0.002$) as better designs lowered the variance floor and the prune bias stopped paying, so earlier and more aggressive settings gave lower per-sample costs at worse MSE. Quoting a single rate is only legitimate because it is read off the two pipelines we actually shipped.

We do not develop this further, for two reasons. Several teams independently found the same tricks, so none of it is a contribution. More to the point, scrambled Sobol is not even the right answer: the frontier of the sampling lane is not a low-discrepancy sequence at all but an exact *spherical 5-design* constructed from Kerdock codes, which integrates every polynomial of degree ≤ 5 on the sphere exactly and thereby exploits the rotational symmetry of the input distribution in a way no scrambled sequence does. Every "QMC" number in this paper is calibrated against our own Sobol estimator and is therefore **conservative in the propagator's favor**: the real sampling frontier sits below the green row of Figure 1, so §7's deficit is if anything understated and the oracle's margin overstated. Correcting for it would strengthen the paper's negative conclusion and weaken its optimistic one. We leave the sampling lane to the people working in it.

F Reproduction map

result	script / artifact
mechanism ablation, Tab. 3	`paper/ablate.py` `ablate_results.json`
cost attribution, Tab. 2	`paper/attribute.py` `attribute_results.json`
lean rebuild, Tab. 4	`paper/lean.py` `lean_results.json`
torch↔estimator parity	`ap_n107_parity_v4.py` `out_n107.log, out_n108b.log`
grader-stack cost & smoke	`submission_v4/replicate_smoke.py` `out_n107c.log`
plane-chain fidelity, §3.3	`ap_n87_planegate.py` `out_n87.log`
conjunction grid, Tab. 1	`ap_n87_planeport.py` `out_n87_planeport*.log`
fidelity ramp & noise arm, Walls 1/3	`ap_n89_cornet.py` `out_n88_*.log`
corrector families, Wall 2	`ap_n89_mpgate.py, ap_n90_paircorr.py` `out_n89_*.log`
injection oracles, Tab. 5	`ap_n91_rankceil.py, ap_n93_k21rank.py` `out_n91_*.log`
sampling-as-input	`ap_n96_sampgate.py, ap_n96b_fusegate.py` `out_n96*.log`
control variates	`ap_n98_cvgate.py` `out_n98.log`
metanetwork scaling	`ap_n97_neuralprop.py` `out_n97_*.log`
exact-structure arbitrage	`ap_n105_ranksamp.py, ap_n103_shrinkgate.py` `out_n10[35].log`
cumulant-tensor costing	— `COSTING_N74.md`
full experiment log (∼110 numbered runs)	— `RESULTS_M1.md`