Bias–Variance Tradeoff Across Control Layers

Matt Liston — Submission #327723

Official Phase 1 public result. Final-layer MSE 1.218×10^{-6}, measured utilization 10.95%, adjusted score 1.334×10^{-7}, with all 50 public networks completed. Private-split evaluations have not been released.

LLM disclosure. Claude Code and OpenAI Codex assisted with ideation, implementation, experiments, debugging, analysis, and drafting. This work has not been peer reviewed, and all mistakes are our own.

The central idea

Submission #327723 keeps Monte Carlo as the final estimator. At an intermediate layer, it compares the sampled activation mean with an estimate of the true population mean. That difference reveals part of the error caused by the current finite batch. The estimator carries this error signal through the rest of the network and subtracts part of it from the layer-32 sample mean.

The layer at which we form this control creates the central tradeoff. Moving later makes the observed sampling error more predictive of the final error, so the control can remove more variance. But the analytic population mean used to center the control becomes less accurate with depth, so the same correction can import more bias. An effectively exact-center experiment exposed both effects. That result led us to learn a correction to the analytic center and use the control at layer 22 with strength 0.6.

1 The analytic-anchor tradeoff

1.1 The task and the score

Fix one width-256, depth-32 MLP. Its weight matrix at layer ℓ is W_ℓ; its Gaussian input is x; and its preactivation and post-ReLU activation are z_ℓ and h_ℓ. The quantity we must estimate is μ, the 256-component expected activation at layer 32.

For evaluation network j, the submitted estimate is $\widehat{\mu}_j$ and the true answer is μ_j. Raw error is the average squared coordinate error. The benchmark then multiplies that error by a utilization factor. In that factor, F_j is counted work, R_j is residual wall time, and B_j is the allowed budget. Below 10% utilization the multiplier is fixed at 0.10; above it, an accuracy gain must pay for its compute.

$$h_0 = x, \qquad z_\ell = h_{\ell-1} W_\ell, \qquad h_\ell = \mathrm{ReLU}(z_\ell), \tag{1}$$

$$x \sim \mathcal{N}(0, I_{256}), \qquad \ell = 1, \ldots, 32, \tag{2}$$

$$\mu = \mathbb{E}_x[h_{32}(x)], \tag{3}$$

$$\mathrm{MSE}_j = \frac{\|\widehat{\mu}_j - \mu_j\|_2^2}{256}, \tag{4}$$

$$s_j = \mathrm{MSE}_j \max\left(0.10, \frac{F_j + 10^{11} R_j}{B_j}\right). \tag{5}$$

1.2 Bias and variance as sample count changes

Fix one MLP and imagine rerunning the estimator with fresh random input batches. Let N be the number of sampled inputs and let $\widehat{\mu}_N$ be the 256-component estimate returned by one run. Every expectation in this subsection is over those hypothetical fresh batches.

The estimator can fail in two different ways. Its average answer can repeatedly miss the target; that repeatable error is the bias vector b_N. It can also fluctuate around its own average from batch to batch. We write the average squared size of that fluctuation as V_N and use the usual approximation $V_N \approx v/N$, where v is the sampling-variance constant.

Because the benchmark scores squared error, it charges the squared length of the bias vector. We write that width-averaged squared bias as β^2; equivalently, β is the root-mean-square repeatable miss per output coordinate. The cross term between bias and fluctuation is zero because the fluctuation around the estimator's own mean averages to zero. Sampling can reduce V_N; it cannot reduce b_N.

$$b_N = \mathbb{E}[\widehat{\mu}_N] - \mu, \qquad \beta^2 = \frac{\|b_N\|_2^2}{256}, \tag{6}$$

$$V_N = \frac{\mathbb{E}\|\widehat{\mu}_N - \mathbb{E}[\widehat{\mu}_N]\|_2^2}{256} \approx \frac{v}{N}, \tag{7}$$

$$\mathrm{MSE}(N) = V_N + \beta^2 \approx \frac{v}{N} + \beta^2. \tag{8}$$

1.3 Why plain Monte Carlo plateaus

Separate compute into fixed work F_0, paid once for this MLP, and marginal work c for each additional sampled input. Let B be this MLP's budget. Above the utilization floor, the modeled score is MSE times total work divided by the budget.

Idealized plain Monte Carlo has neither a fixed analytic pass nor a bias floor, so $F_0 = 0$ and $\beta = 0$. Increasing N then cannot improve the score: twice as many inputs roughly halve sampling variance and double sample work. The sample count cancels, leaving only the variance per input, the cost per input, and the fixed network budget.

$$\mathrm{score}(N) \approx \left(\frac{v}{N} + \beta^2\right)\frac{F_0 + cN}{B}. \tag{9}$$

$$F_0 = 0,\ \beta = 0 \quad \Longrightarrow \quad \mathrm{score}(N) \approx \frac{vc}{B}. \tag{10}$$

1.4 The finite optimum with an analytic control

An analytic control can lower the variance constant v, but it adds fixed work F_0 and an imperfect analytic center can leave $\beta > 0$. Once both effects are present, there is a finite best sample count. More samples dilute the setup penalty vF_0/N. At the same time, they increase the compute spent below the fixed bias floor, represented by $\beta^2 cN$.

Within this approximate model, only those two terms depend on N. Their sum is smallest when they are equal. The resulting optimum is continuous and assumes $F_0 > 0$, $\beta > 0$, approximately $1/N$ variance, roughly stable bias, linear marginal cost, and operation above the utilization floor. It is not specific to cumulant-aware moment closure or to this particular anchor.

$$B\,\mathrm{score}(N) \approx vc + \beta^2 F_0 + \underbrace{\frac{vF_0}{N}}_{\text{setup cost diluted by samples}} + \underbrace{\beta^2 cN}_{\text{samples spent below the bias floor}}, \tag{11}$$

$$\frac{vF_0}{N^\star} = \beta^2 cN^\star, \qquad N^\star = \sqrt{\frac{vF_0}{\beta^2 c}}, \tag{12}$$

$$\mathrm{score}^\star = \frac{\left(\sqrt{vc} + \beta\sqrt{F_0}\right)^2}{B}. \tag{13}$$

1.5 What the intermediate-layer control does

Choose a control layer k between 1 and 31. Its true population mean is μ_k. For one Monte Carlo batch, the measured layer-k mean is $\bar{h}_k$. The difference between them is the batch's sampling error, denoted ε_k.

The true mean μ_k is unavailable, so the estimator centers the batch using an analytic approximation a_k. The repeatable error in that center is δ_k. Subtracting a_k from $\bar{h}_k$ produces the control residual r_k: the useful sampling error minus the unwanted analytic-center error.

At the output, $\bar{h}_{32}$ is the final mean from the same batch and e_{32} is its ordinary Monte Carlo error. A linear map $T_k : \mathbb{R}^{256} \to \mathbb{R}^{256}$ carries a small perturbation at layer k through the remaining weights and ReLU gates. The scalar γ is the fraction of this transported residual that we subtract. The resulting error has three pieces: the original output error, the batch error removed by the control, and the analytic-center error imported by the same subtraction.

$$\bar{h}_k = \mu_k + \varepsilon_k, \qquad a_k = \mu_k + \delta_k, \tag{14}$$

$$r_k = \bar{h}_k - a_k = \varepsilon_k - \delta_k, \tag{15}$$

$$e_{32} = \bar{h}_{32} - \mu, \tag{16}$$

$$\widehat{\mu}_\gamma = \bar{h}_{32} - \gamma T_k r_k, \tag{17}$$

$$\widehat{\mu}_\gamma - \mu = e_{32} - \gamma T_k \varepsilon_k + \gamma T_k \delta_k. \tag{18}$$

1.6 How bias and variance set the control strength

To isolate the design tradeoff, imagine changing only the Monte Carlo batch while holding the analytic center and downstream transport fixed. Assume the output sampling error e_{32} and the control-layer sampling error ε_k both average to zero across batches.

Four quantities then determine the control's MSE. First, V_{out} is the output variance before control. Second, S_k measures how much the transported layer-k sampling error moves with the output error; this is the useful shared signal. Third, $V_{\mathrm{ctrl},k}$ is the total variance carried by that transported sampling error. Fourth, $B_{\mathrm{anchor},k}$ is the squared size of the transported analytic-center error. The last quantity is squared because MSE scores squared vector length.

Increasing γ earns a linear reward by removing shared sampling error, but pays two quadratic costs: variance carried by the control and bias imported from the analytic center. When the denominator is positive, the best unconstrained control strength is therefore the useful shared signal divided by the combined control-variance and anchor-bias penalty; with $B_{\mathrm{anchor},k} = 0$ this reduces to the classical

control-variate coefficient [1]. Moving the control later can increase the useful shared signal, because less of the network remains between the observed error and the output. The same move can increase analytic-center error because approximation error accumulates with depth. This is the tradeoff that motivated the learned correction.

$$V_{\text{out}} = \frac{\mathbb{E}\|e_{32}\|_2^2}{256}, \qquad S_k = \frac{\mathbb{E}[e_{32}^\top T_k \varepsilon_k]}{256}, \tag{19}$$

$$V_{\text{ctrl},k} = \frac{\mathbb{E}\|T_k \varepsilon_k\|_2^2}{256}, \qquad B_{\text{anchor},k} = \frac{\|T_k \delta_k\|_2^2}{256}. \tag{20}$$

$$\text{MSE}_k(\gamma) := \frac{\mathbb{E}\|\widehat{\mu}_\gamma - \mu\|_2^2}{256} \tag{21}$$

$$= \underbrace{V_{\text{out}} - 2\gamma S_k + \gamma^2 V_{\text{ctrl},k}}_{\text{sampling variance after control}} + \underbrace{\gamma^2 B_{\text{anchor},k}}_{\text{imported anchor bias squared}}, \tag{22}$$

$$\gamma_k^\star = \frac{S_k}{V_{\text{ctrl},k} + B_{\text{anchor},k}}. \tag{23}$$

2 How the estimator developed

2.1 Sampling became the endpoint

Extending the deterministic cumulant-aware moment pass described in Section 4.1 through all 32 nonlinear layers accumulated too much error. Across successive development versions, replacing the analytic endpoint with a 6,144-input Monte Carlo endpoint improved adjusted score from about 1.086×10^{-6} to 6.74×10^{-7}. Sampling was therefore the more reliable endpoint, even though it left substantial batch-to-batch variance.

The next changes removed sampling noise whose answer was already known. The input construction matched low-order Gaussian moments, paired inputs with their negatives, and replaced the sampled first-layer mean by its exact value. The first two changes are custom uses of ZCA-style whitening and antithetic sampling [2, 3]. Covariance matching produced the largest early gain. Sign pairing and exact first-layer recentering were smaller refinements. The exact first-layer formula appears below; the scored receipts are consolidated in Section 5.

$$\mathbb{E}[\text{ReLU}(z_{1q})] = \frac{\|W_{1,:,q}\|_2}{\sqrt{2\pi}}, \qquad q = 1, \ldots, 256. \tag{24}$$

2.2 Moving the control later exposed the bottleneck

Closure was not accurate enough to be the final answer, but it was useful as an estimate of an earlier population mean. A layer-8 control variate improved an archived adjusted score from roughly 3.42×10^{-7} to 2.77×10^{-7}. This established that an intermediate-layer residual could predict part of the output error.

The next experiment separated two questions that ordinary development runs mixed together: how much useful error signal exists at a layer, and how accurately can closure center that layer? The no-control

estimate and downstream transport were held fixed. One arm used the uncorrected moment-closure center. A diagnostic arm used an effectively exact center unavailable to the submission.

The result was decisive. With the uncorrected moment-closure center, raw MSE worsened as the control moved from layer 13 to 24. With the effectively exact center, raw MSE improved sharply and the layer-24 control removed 45.2% relative to no control. Later layers contained more useful information; inaccurate centering, not a lack of correlation with the output, was the bottleneck.

Table 1: Question tested: when centering error is removed, does a later control layer predict more output error? Raw final-layer MSE on 30 development MLPs; lower is better.

Control layer	No control	Closure center	Effectively exact center
13	1.928×10^{-6}	1.785×10^{-6}	1.565×10^{-6}
18	1.928×10^{-6}	1.845×10^{-6}	1.359×10^{-6}
24	1.928×10^{-6}	1.896×10^{-6}	1.056×10^{-6}

This was an unscored mechanism diagnostic: utilization and adjusted score were not measured, and the effectively exact center was unavailable to the submission.

2.3 The depth result defined the learning problem

The effectively exact-center experiment told us what to learn. We did not need a model to replace Monte Carlo or predict the final answer. We needed a model with the narrower job of correcting the analytic population mean at the control layer.

The direct difference between the sampled mean and analytic mean was deliberately excluded from the inputs. That quantity contains the sampling error the outer control must preserve; a model that copied it would center the residual on the current batch and erase the signal the control is meant to subtract. This is the same general overfitting hazard that motivates out-of-sample fitting in learned control variates [4].

Layer 24 retained more raw headroom, but it was too expensive in the tested implementation. A layer-24 board experiment reached raw MSE 1.4514×10^{-6}, utilization 13.61%, and adjusted score 1.9765×10^{-7}. It improved raw accuracy relative to the active predecessor but lost after score adjustment. Layer 22 with control strength 0.6 was the better accuracy–cost compromise.

3 Submission #327723

The submitted estimator first builds an analytic estimate of the layer-22 population mean using the cumulant-aware moment pass described in Section 4.1, then adds the frozen recurrent model's correction. It chooses between 6,000 and 8,600 inputs, constructs a covariance-matched antithetic cloud, pins the exact layer-1 mean, and propagates the batch with early mean recentering and probability-guided pruning.

At layer 22, the estimator subtracts its corrected analytic center from the sampled mean. It carries this residual through layers 23–32 using the actual weights and the observed fraction of open ReLU gates, then subtracts 60% of the transported residual from the output sample mean. Smaller multi-depth, tail, and activation-energy controls refine this estimate. Three levels of Strassen multiplication reduce the cost of the sampled matrix products [5].

Let a_{22} be the corrected analytic center, $\bar{h}_{22}$ and $\bar{h}_{32}$ the sampled means at layers 22 and 32, and $\widehat{T}_{22\to32}$ the empirical downstream transport. The central submitted correction is

$$r_{22} = \bar{h}_{22} - a_{22}, \tag{25}$$

$$\widehat{\mu}^{\text{main}} = \bar{h}_{32} - 0.6\,\widehat{T}_{22\to32}(r_{22}). \tag{26}$$

4 Learning the layer-22 center

4.1 The analytic base center

The base center is computed by deterministic, approximate moment propagation adapted from ARC's cumulant-and-Hermite framework for wide random MLPs [6]. The pass starts with the exact mean and covariance of the Gaussian input. Each affine layer transforms the maintained moments through the actual weight matrix. At each ReLU, the method starts from the exact marginal mean for a Gaussian preactivation and adds corrections based on maintained marginal third and fourth cumulants.

Through layer 18, the state contains a mean vector, a dense covariance matrix, a rank-capped factorization of selected third-order dependence, and one fourth cumulant per neuron. The third-order factorization is capped at 384 carrier terms; it is not a full third-order tensor. The fourth-order state is diagonal, not a full fourth-order tensor. The dense covariance update uses a cubic Hermite truncation plus a selected third-cumulant correction.

From layer 19 onward, a cheaper Gaussian-reference tail discards the third-order carrier and fourth-cumulant state while continuing to propagate the mean and dense covariance. The tail's layer-22 output is the base center; the learned model then corrects it. The input moments, affine transformations of the maintained state, and Gaussian ReLU identities are exact in isolation. The nonlinear distribution propagation is approximate because the Hermite series is truncated and only compressed third-order and marginal fourth-order information are retained.

The equations below show the affine update and marginal mean update. The covariance and carrier recurrences are summarized above rather than reproduced in full. Here ϕ and Φ are the standard-normal density and CDF, all scalar functions are applied coordinatewise, and $\kappa_{3,z,m}$ and $\kappa_{4,z,m}$ are the maintained marginal preactivation cumulants. The labels "cumulant-aware" and "Gaussian-reference tail" name two approximate branches; neither denotes an exact population mean.

$$\mu_0 = 0, \qquad \Sigma_0 = I_{256}, \tag{27}$$

$$\mu_{z,m} = \mu_{m-1} W_m, \qquad \Sigma_{z,m} = W_m^\top \Sigma_{m-1} W_m, \tag{28}$$

$$v_{z,m} = \operatorname{diag}(\Sigma_{z,m}), \qquad \alpha_m = \frac{\mu_{z,m}}{\sqrt{v_{z,m}}}, \tag{29}$$

$$g(\mu, v) = \sqrt{v}\,\phi\!\left(\frac{\mu}{\sqrt{v}}\right) + \mu\,\Phi\!\left(\frac{\mu}{\sqrt{v}}\right), \tag{30}$$

$$c_3(\mu, v) = -\frac{\mu\,\phi(\mu/\sqrt{v})}{v^{3/2}}, \tag{31}$$

$$c_4(\mu, v) = \frac{\left((\mu^2/v) - 1\right)\phi(\mu/\sqrt{v})}{v^{3/2}}, \tag{32}$$

$$\widetilde{\mu}_m^{\mathrm{CA}} \approx g(\mu_{z,m}, v_{z,m}) + \frac{\kappa_{3,z,m}}{6} c_3(\mu_{z,m}, v_{z,m}) + \frac{\kappa_{4,z,m}}{24} c_4(\mu_{z,m}, v_{z,m}), \tag{33}$$

$$\widetilde{\mu}_m^{\text{G-tail}} = g(\mu_{z,m}, v_{z,m}), \qquad m = 19, \ldots, 31, \tag{34}$$

$$a_m^{\mathrm{base}} = \begin{cases} \widetilde{\mu}_m^{\mathrm{CA}}, & 2 \le m \le 18, \\ \widetilde{\mu}_m^{\text{G-tail}}, & 19 \le m \le 31. \end{cases} \tag{35}$$

4.2 What the model predicts

At each supervised layer m from 2 through 31, the piecewise analytic pass above produces a base center a_m^{base}. A separate high-sample run produces an approximate reference mean μ_m^{ref}. Their difference, $\Delta\mu_m$, is the training target. One coordinate of this vector is the estimated error in the analytic population mean of one neuron. It is not an individual sampled activation and it is not the layer-32 answer.

The recurrent model outputs a normalized prediction $\widehat{u}_m$. A fixed positive scale σ_μ converts that output back to activation units. At deployment, the layer-22 prediction corrects the main analytic center. Predictions at layers 13, 16, 19, and 21 also correct the centers used by smaller multi-depth controls. The main corrected center is then handed to the intermediate-layer control described in Section 3.

$$\Delta\mu_m = \mu_m^{\mathrm{ref}} - a_m^{\mathrm{base}}, \qquad m = 2, \ldots, 31, \tag{36}$$

$$\widehat{u}_m \approx \frac{\Delta\mu_m}{\sigma_\mu}, \tag{37}$$

$$a_{22} = a_{22}^{\mathrm{base}} + \sigma_\mu \widehat{u}_{22}, \qquad \sigma_\mu = 0.0033820162061601877. \tag{38}$$

4.3 What information the model uses

At each layer the recurrent cell receives 16 values per neuron. Nine analytic channels report the closure's estimated mean, variance, Gaussian-reference gate probability, marginal third and fourth cumulants, post-ReLU variance, and three Gaussian-response coefficients. Five same-batch channels compare sampled and analytic variance, third-cumulant, and fourth-cumulant information. Two recurrent messages carry earlier predicted corrections through the next weight matrix.

The direct sampled-mean gap is not included. The 14 local channels are standardized with stored constants, except the gate-probability channel, which remains on its natural scale. The hidden state and both recurrent messages begin at zero. These inputs feed one GRU cell [7] with 96 hidden units, followed by a $96 \to 192 \to 2$ readout with a GELU nonlinearity [8], for 51,842 trained parameters. The

first output predicts the normalized mean correction. The second is an unsupervised auxiliary state used only by the recurrence. Exact channels, data, loss, and warm-start history are in Appendix B.

4.4 Does the learned center correction help?

A later probe changed one thing: the center used by the layer-22 control. One version used the analytic center plus the frozen model's predicted correction. The other replaced the model output by zero and therefore used the analytic center alone. It did *not* switch the layer-22 control itself off.

A separate *control weight* scaled the complete layer-22 adjustment after the residual was transported to the output. A zero control weight disabled that layer-22 control altogether; a nonzero weight retained it. This is why the analytic-only version can have a nonzero control weight: that number belongs to the underlying control variate, not to the learned correction. Earlier derivations denote this weight by γ (gamma). It is unrelated to β (beta), which denotes RMS bias in the sample-count model of Section 1.2.

The two versions searched different finite lists of control weights, shown in full below. Bold marks the lowest raw MSE observed within each row. All other components of the reduced estimator, including its separate activation-energy control, were held fixed.

Table 2: Finite-grid learned-center probe on the same 20 MLPs. Each entry is control weight → mean raw MSE, with MSE reported in units of 10^{-7}. Bold marks the best tested cell in each row.

Samples	Layer-22 center used	Tested control-weight cells
30,000	analytic center plus frozen learned correction	0.40 → 4.5955; **0.55 → 4.5385**; 0.70 → 4.7290
30,000	analytic center alone	0 → 5.9576; **0.10 → 5.4812**; 0.20 → 6.0747
60,000	analytic center plus frozen learned correction	**0.40 → 2.5710**†; 0.55 → 2.7849; 0.70 → 3.2400
60,000	analytic center alone	**0 → 3.1797**; 0.10 → 3.1945; 0.20 → 4.1908

†Smallest control weight tested in that row; not a known continuous optimum.

Comparing the best tested cells, the learned-corrected center lowered mean raw MSE by 17.2% at 30,000 samples and 19.1% at 60,000 samples. This was a 20-MLP test of the identical frozen model in a later reduced stack, not an ablation of the complete submission. Utilization and adjusted score were not recorded, so these percentages establish local raw-MSE value, not shares of the official score.

4.5 Same-batch features also weaken the control

Five model inputs come from the same Monte Carlo batch whose error the control is trying to remove. This makes them informative about closure error, but it also lets the model react to the particular batch fluctuation that should remain visible in the control residual.

A later diagnostic reran 48 fixed MLPs with eight independent batches each. Under that probe's downstream geometry, the propagated fluctuation in the learned correction was strongly aligned with layer-22 sample-mean noise. Before subtracting the same-batch learned correction, the control residual explained 66.9% of output-noise variance; afterward it explained only 42.5%. Replacing the learned correction by an unavailable across-batch average improved MSE by a further 22.6%. This diagnostic did not record utilization or adjusted score.

These results do not contradict the on/off probe. The learned correction can reduce enough analytic-center error to help overall while returning some batch variance to the estimator. Preserving the systematic correction without responding to same-batch noise is the clearest remaining opportunity.

Table 3: Question tested: does the learned correction reuse noise from the same Monte Carlo batch? Later diagnostic on 48 fixed MLPs with eight batches each; the across-batch average is unavailable to the submission.

Measurement	Result
Correlation between propagated learned-correction fluctuation and layer-22 sample-mean noise	0.885
Output-noise variance explained before same-batch learned correction	66.9%
Output-noise variance explained after same-batch learned correction	42.5%
MSE change using unavailable across-batch mean correction	22.6% lower

5 Results and evidence

The official score belongs to the complete estimator. We did not rerun that exact package with each component removed while also retuning the control strength and sample count. Component effects are therefore local to the named experiment and are not additive shares of the final score. Section 5.1 lists every archived milestone of the adjusted score in development order, so the complete trajectory from about 1.086×10^{-6} to the official 1.334×10^{-7} is visible in one place. The tables after it isolate the receipts behind the major steps. Every development step shown in this section lowered adjusted score (the one exception is a row explicitly labeled as a measurement-frame bridge, which re-grades the same package on the public board); development steps that *raised* adjusted score were reverted, and their measurements appear in Appendix C rather than here. "Not recorded" means that quantity was not saved for the same experiment; values from other panels are not substituted.

Table 4: Official Phase 1 public result for the complete submission. Private-split evaluations have not been released.

Raw MSE	Measured utilization	Adjusted score	Justified conclusion
1.218×10^{-6}	10.95%	1.334×10^{-7}	The complete package finished all 50 public networks. This row validates the composition, not an individual component share.

5.1 The full adjusted-score trajectory

Two measurement frames appear in the development record. Through the early era, every arm ran at or below the 10% utilization floor, where adjusted score is exactly raw MSE times 0.10, so the development grader and the public board apply the same multiplier and differ only through their evaluation populations. Once the intermediate-layer control pushed utilization above the floor, the frames diverged: the development grader recorded 2.77×10^{-7} for the same package the public board graded at 3.05×10^{-7}, because the two meters price above-floor compute differently. Table 5 therefore reports the floor-era development milestones, and Table 6 switches to public-board grades at that same package and stays on the board through the official result. Successive rows share their endpoints, so each row's starting score is the previous row's result; two early rows aggregate smaller promotions whose individual paired receipts were not retained.

Table 5: Adjusted-score trajectory, development-record era. All arms at or near the utilization floor, where the multiplier is clamped at 0.10 in every grading frame. Scores in units of 10^{-7}; each step starts from the previous row's score.

Step	Adjusted	Note
Starting point: fully analytic endpoint	≈ 10.86	Deterministic moment propagation through all 32 layers.
Monte Carlo endpoint replaced the analytic endpoint	6.74	Method pivot across successive versions.
Covariance-matched input cloud	3.77	The promotion also changed the sample count.
Exact layer-1 mean pinned	3.73	Receipt in Table 7.
Smaller sampler and knob promotions	3.59	Aggregated; individual paired receipts not retained.
Two-point sign pairing	3.51	Receipt in Table 7.
Smaller tuning promotions	3.42	Aggregated; individual paired receipts not retained.
Layer-8 intermediate-layer control	2.77	Composite promotion; utilization rose above the floor, ending frame agreement.

Table 6: Adjusted-score trajectory, public-board era. Every archived board milestone of the package lineage that became submission #327723, in units of 10^{-7}. Each step starts from the previous row's score; the first row re-grades the last development-era package on the board.

Step	Adjusted	Note
Layer-8-control package graded on the public board	3.05	Measurement-frame bridge, not a method change; the development grader had priced above-floor utilization about 10% optimistically.
Closure-anchor and control refinements	2.901	Successive promotions; individual receipts not retained.
Probability-guided pruning, in the explicit-gather form the board credits	2.297	Compute trade; the isolated off/on receipt is in Table 9.
Earlier pruning start layer	2.259	Pure compute cut.
Pruning and control-strength retune	2.214	Final configuration of this era.
Sample count cut to the utilization floor	1.674	Raw error rose while billed compute fell more; detail in Table 9.
Dispatch-count reduction	1.658	Predictions bit-identical to the predecessor.
Stricter dispatch floor	1.649	Predictions bit-identical.
Gated dead-path cuts, composite promotion	1.596	Small raw change plus a further compute cut.
Layer-22 control with the learned-corrected center replaced the earlier control, with billing fusions	1.575	Raw MSE fell 22.5% while utilization rose off the floor; detail in Table 8.
Tail-mean innovation ladder and dead-tail dispatch cut	1.544	Promoted together; detail in Table 8.
Billing-alignment cuts	1.519	Symmetry-aware pricing and dtype-idempotent casts; predictions bit-identical.
Three-level Strassen multiplication	1.366	Raw unchanged at meaningful precision; receipt in Table 9.
Final Phase 1 package, submission #327723	1.334	Same statistical estimator with one remaining billing-symmetry shortcut removed; graded at a lower utilization draw. The official public result.

5.2 Sampling backbone

These successive grader records explain why sampling became the endpoint and how its input cloud was improved. The covariance-matching row is a sequential promotion, not a one-switch ablation of the final sampler; a separate matched 30-MLP test corroborated the mechanism with a 35.4% raw-MSE reduction. All arms shown here were below the 10% floor. The two seams between rows are the aggregated smaller promotions listed in Table 5.

Table 7: Sampling results. Each arrow compares the named predecessor with the named successor.

Method / test	Raw MSE	Utilization	Adjusted score	Justified conclusion
Analytic endpoint → plain MC	$\approx 1.086 \times 10^{-5} \rightarrow 6.74 \times 10^{-6}$	$\approx 9.77\% \rightarrow 9.53\%$	$1.086 \times 10^{-6} \rightarrow 6.74 \times 10^{-7}$	The sampled endpoint beat that analytic predecessor. This was a method pivot, not an isolated switch.
Plain MC → covariance-matched MC	$6.74 \times 10^{-6} \rightarrow 3.77 \times 10^{-6\dagger}$	$9.53\% \rightarrow 9.75\%$	$6.74 \times 10^{-7} \rightarrow 3.77 \times 10^{-7}$	The covariance-matched successor scored 44.1% lower; the comparison also changed the sample count.
Add exact layer-1 mean	$3.77 \times 10^{-6\dagger} \rightarrow 3.73 \times 10^{-6\dagger}$	$9.75\% \rightarrow 9.75\%$	$3.77 \times 10^{-7} \rightarrow 3.73 \times 10^{-7}$	A 1.1% local score improvement at unchanged measured utilization.
Add two-point sign pairing	$3.59 \times 10^{-6\dagger} \rightarrow 3.51 \times 10^{-6\dagger}$	$9.80\% \rightarrow 9.80\%$	$3.59 \times 10^{-7} \rightarrow 3.51 \times 10^{-7}$	A 2.2% local gain for this predecessor; it is not the isolated marginal of the later four-block sampler.

†Raw MSE is inferred as adjusted score divided by 0.10 because the archived grader receipt reported that arm as floor-clamped.

5.3 Intermediate control and analytic center

The first row measures the composite promotion that established the intermediate-layer control. The second measures the board interval in which the layer-22 control with the learned-corrected center replaced the earlier control: raw MSE fell 22.5%, but the added analytic work raised utilization off the floor, so adjusted score improved only 1.3%. The third is the tail-control promotion that followed. The unscored mechanism diagnostics that motivated the layer-22 design are reported in Table 1 (effectively exact-center depth probe) and Table 2 (frozen learned-center probe). A more faithful analytic center that lowered raw error but *worsened* adjusted score through its extra compute is recorded in Appendix C.

Table 8: Control and center results. Each arrow compares the named predecessor with the named successor.

Method / test	Raw MSE	Utilization	Adjusted score	Justified conclusion
No intermediate control → layer-8 controlled stack	$3.42 \times 10^{-6\dagger} \rightarrow 8.72 \times 10^{-7}$	$9.80\% \rightarrow 31.8\%$	$3.42 \times 10^{-7} \rightarrow 2.77 \times 10^{-7}$	The composite promotion improved adjusted score 19.0%; it is not a control-off ablation of #327723.
Earlier control → layer-22 control with learned-corrected center, with billing fusions	$1.595934 \times 10^{-6} \rightarrow 1.236214 \times 10^{-6}$	$10.00\% \rightarrow 12.74\%^{\ddagger}$	$1.596194 \times 10^{-7} \rightarrow 1.575247 \times 10^{-7}$	The board interval containing the main mechanism's promotion cut raw MSE 22.5%; its closure cost raised utilization, so adjusted improved 1.3%. A composite interval, not a single-switch ablation.
Add tail-mean innovation ladder and dead-tail dispatch cut	$1.236214 \times 10^{-6} \rightarrow 1.2184 \times 10^{-6}$	$12.74\%^{\ddagger} \rightarrow 12.68\%$	$1.575247 \times 10^{-7} \rightarrow 1.5444 \times 10^{-7}$	A small tail-control accuracy gain and a free dispatch cut promoted as one change; Appendix C bounds the tail control's isolated effect.

†Raw MSE is inferred as adjusted score divided by 0.10 because the archived grader receipt reported that arm as floor-clamped. ‡Utilization inferred as adjusted score divided by raw MSE; not separately archived.

5.4 Compute and utilization

The compute experiments answer a different question: they trade raw accuracy or algorithm structure against the utilization multiplier. The floor-move row shows the sharpest version — raw error *rose* while adjusted score improved 24.4% — because below-floor compute is free and the metric rewards the cheaper operating point. Pruning is the same trade inside one component. Strassen changes the multiplication algorithm while leaving the statistical estimator fixed; that row uses an archived board predecessor pair predating the fully clean final package, and a separate matched 100-MLP gate corroborated it with a negligible raw change and an 18.08% reduction in billed multiplication work.

Table 9: Compute and utilization results. The pruning row is a paired off/on measurement on 250 development networks; the others are board predecessor–successor pairs.

Method / test	Raw MSE	Utilization	Adjusted score	Justified conclusion
Sample count cut to the utilization floor	not retained $\rightarrow 1.600656 \times 10^{-6}$	$\approx 25\%^{*} \rightarrow 10.46\%$	$2.2143 \times 10^{-7} \rightarrow 1.674414 \times 10^{-7}$	Cutting samples raised raw error but improved adjusted score 24.4%; the utilization floor makes the cheaper operating point optimal.
Probability-guided pruning, off $\rightarrow$ on	$2.043738 \times 10^{-6} \rightarrow 2.125605 \times 10^{-6}$	$12.9904\% \rightarrow 9.2907\%$	$2.654889 \times 10^{-7} \rightarrow 2.125605 \times 10^{-7}$	Raw MSE worsened 4.0%, but adjusted score improved 19.9%; pruning is a compute trade, not an accuracy gain.
Dispatch and dead-path cuts, three successive promotions	$1.600656 \times 10^{-6} \rightarrow 1.595934 \times 10^{-6}$	$10.46\% \rightarrow 10.00\%$	$1.674414 \times 10^{-7} \rightarrow 1.596194 \times 10^{-7}$	Removing dispatch operations and gating dead paths left raw error nearly unchanged while adjusted improved 4.7%.
Billing-alignment cuts	1.2184×10^{-6}, bit-identical	$12.68\% \rightarrow 12.47\%$	$1.5444 \times 10^{-7} \rightarrow 1.5192 \times 10^{-7}$	Symmetry-aware pricing and dtype-idempotent casts changed no prediction bit; adjusted improved 1.6% through billing alone.
Ordinary multiplication $\rightarrow$ three-level Strassen	$1.218441 \times 10^{-6} \rightarrow 1.218437 \times 10^{-6}$	$12.470\% \rightarrow 11.2142\%$	$1.5192 \times 10^{-7} \rightarrow 1.366374 \times 10^{-7}$	Raw MSE was unchanged at meaningful precision; adjusted score improved 10.06% through lower compute.

*The immediate predecessor's utilization was not retained; the adjacent archived grade in that era measured 24.6%.

No exact final-package removal of the main layer-22 control, learned center, smaller energy or tail controls, adaptive sample count, or fusions was archived. Appendix C records weaker development evidence, and the development steps that raised adjusted score, without promoting those measurements to final-score contributions.

6 Limitations, reproducibility, and disclosure

Approximate centering, early recentering, pruning, empirical transport, fitted coefficients, and same-batch learned inputs make the estimator biased. The high-sample training references estimate population means; they are not exact truth. The sample-derived features used a deployment-style predecessor path and an earlier deterministic sample-seed convention, then underwent the deterministic rebase described in Appendix B; they were not generated by the exact submitted path.

The 800-network diagnostic set was excluded from gradients in the final continuations, but inherited weights had already trained on those networks in an earlier stage. Public-board and whole-estimator evidence informed model selection. Development panels differ, so their effects are never added. Apple MPS training was not bit-deterministic; the selected weights and embedded constants are fixed and bitwise-verified.

This write-up describes exactly one graded package: **submission #327723**. The weights are embedded numerical constants, and there is no lookup keyed by an evaluation network or evaluation seed.

Research-process disclosure. Development used an autonomous edit–evaluate–retain workflow inspired by Karpathy's *autoresearch* pattern [11], together with targeted literature reviews. The reviews supplied background, diagnostic hypotheses, and rejected alternatives; citations below are not claims that every surveyed method entered the submitted estimator.

A The exact control identity

The clean derivation in Section 1 holds the analytic center and transport fixed while the batch changes. The submitted estimator is more general: five learned inputs and the empirical ReLU gates come from the current batch. For the exact identity, let e be the output error before the main layer-22 correction and let p be the complete transported control produced on that same run. The corrected error is $e - \gamma p$.

No independence or unbiasedness assumption is needed. The term linear in γ measures alignment between the actual control and the error we want to remove. The quadratic term charges everything carried by the control, including useful signal, analytic-center error, and same-batch noise. If the construction of p is held fixed while only its scalar coefficient changes, the unconstrained optimum is alignment divided by total control energy. This identity applies to any anchor, not only the cumulant-aware moment closure used here.

$$\mathrm{MSE}(\gamma) := \frac{\mathbb{E}\|e - \gamma p\|_2^2}{256} \tag{39}$$

$$= \mathrm{MSE}(0) - 2\gamma \frac{\mathbb{E}[e^\top p]}{256} + \gamma^2 \frac{\mathbb{E}\|p\|_2^2}{256}, \tag{40}$$

$$\gamma^\star = \frac{\mathbb{E}[e^\top p]}{\mathbb{E}\|p\|_2^2}. \tag{41}$$

B Learned-correction specification

B.1 Reference targets

Training used width-256, depth-32 random MLPs from the challenge distribution. For every MLP, a separate high-sample run evaluated 160,000 Gaussian inputs and their negatives, giving 320,000 forward

trajectories. It inserted the exact layer-1 mean and averaged each neuron's activation at layers 2–31. The resulting 256-component vector μ_m^{ref} is a variance-reduced estimate of the layer-m population mean, not exact truth.

The stored target was the difference between the high-sample reference and the piecewise analytic center defined in Section 4.1. Layer 32 had zero loss weight. Coordinates classified as analytically dead were assigned the negative analytic center, forcing the corrected center to zero there. The target itself is defined in Equation (36).

$$\mu_m^{\mathrm{ref}} \approx \mathbb{E}_x[h_m(x)], \qquad m = 2, \ldots, 31, \tag{42}$$

B.2 Fixed training data

The later anchor-specific corpus contained 4,000 fixed MLPs. The first 3,200 were used for gradient updates and 800 were used for stage diagnostics. The exact seed sets appear below. One deployment-style predecessor run per MLP produced the model-input tensor for layers 1–31 using covariance matching, antithetics, early recentering, pruning, and 6,000–8,600 samples. This was not the exact submitted execution path: feature collection continued through layer 31 and used earlier pruning and sample-count constants. Closure-dependent channels were then deterministically rebased to the piecewise center.

The 800 diagnostic MLPs were excluded from gradients during the later stages, but all had already appeared in the 15,000-MLP gradient corpus used by the pretrained ancestor. They were also inspected repeatedly. They are a reused stage diagnostic, not an out-of-sample validation set.

Table 10: Fixed corpus used in the later anchor-specific training stages.

Use	MLPs	Exact seeds
Gradient updates	3,200	3000000–3002999 and 3013500–3013699
Stage diagnostics	800	3013700–3014499

B.3 The 16 per-layer inputs

For one neuron at one layer, let μ_z and v_z be the closure preactivation mean and variance, let σ_z be the corresponding standard deviation, and let α be the standardized mean. The symbols Φ and ϕ denote the standard-normal CDF and density. Here κ_3 is the scalar marginal centered third moment and κ_4 is the scalar marginal centered fourth moment minus three times variance squared; neither is a full multivariate cumulant tensor. The first nine channels describe the analytic closure. The next five compare moments measured from the current sample cloud with closure. The direct sampled-mean gap is deliberately absent. The last two inputs are messages formed from the model's preceding outputs and the current weight matrix. In their formula, $\odot$ denotes elementwise multiplication and $W_m^{\odot 2}$ denotes the entrywise square of W_m.

Channels 0–13 use stored affine standardization except the open-probability channel, which remains on its natural scale. Those constants and the output scale came from the 15,000-MLP predecessor rather than being refitted on the later 3,200 examples. Deployment folds this affine transform into the GRU weights. Before layer 2, the hidden state and both recurrent messages are zero.

Table 11: The nine analytic local channels.

Index	Exact value	Role
0	preactivation mean μ_z	analytic location
1	preactivation variance v_z	analytic scale
2	Gaussian-reference gate probability $\Phi(\alpha)$	reference ReLU gate
3	preactivation $\kappa_{3,z}^{\mathrm{cl}}$	third-cumulant signal
4	preactivation $\kappa_{4,z}^{\mathrm{cl}}$	fourth-cumulant signal
5	post-ReLU variance v_h^{cl}	analytic scale after ReLU
6	$\phi(\alpha)/\sigma_z$	order-2 Gaussian response
7	$-\mu_z\phi(\alpha)/(\sigma_z v_z)$	order-3 Gaussian response
8	$(\alpha^2-1)\phi(\alpha)/(\sigma_z v_z)$	order-4 Gaussian response

Table 12: The five same-batch local channels.

Index	Exact value
9	sampled post-ReLU variance minus v_h^{cl}
10	sampled preactivation $\kappa_3 - \kappa_{3,z}^{\mathrm{cl}}$
11	sampled preactivation $\kappa_4 - \kappa_{4,z}^{\mathrm{cl}}$
12	sampled post-ReLU third cumulant $\kappa_{3,h}$
13	sampled post-ReLU fourth cumulant $\kappa_{4,h}$

$$\sigma_z = \sqrt{v_z}, \qquad \alpha = \frac{\mu_z}{\sigma_z}, \tag{43}$$

$$p_{\mu,m} = \Phi(\alpha_m) \odot (\widehat{u}_{m-1} W_m), \qquad p_{q,m} = q_{m-1} W_m^{\odot 2}. \tag{44}$$

B.4 Architecture

The shipped model is a single GRU cell with input width 16 and hidden width 96. Its readout is a linear map from 96 to 192 units, a GELU nonlinearity, and a final linear map to two outputs, for 51,842 trained parameters. The first output is the normalized mean correction $\widehat{u}_m$. The second output is the auxiliary value q_m used by the recurrence. It was not directly supervised in the final stage and is not claimed to estimate calibrated variance.

B.5 Training objective

The final stage supervised the first output at layers 2–31. It weighted layers 16–30 twice as heavily as layers 2–15 and 31; layer 32 had zero weight. The stage used Adam [9] for 25 epochs with learning rate 4.6×10^{-4}, weight decay 10^{-6}, batch size 128, and random seed 2. Loading a warm start restored model weights but restarted the optimizer.

One earlier width-96 stage used a mixed objective at layers 18, 20, 24, 28, and 30: 30% direct correction error and 70% error after a stored downstream transport. All other supervised layers kept the direct objective.

$$\mathcal{L}_{\text{direct}} = \sum_{m=2}^{31} w_m \, \text{mean}_{\text{MLP, neuron}} \left(\widehat{u}_m - \frac{\Delta\mu_m}{\sigma_\mu} \right)^2, \tag{45}$$

$$w_m = \begin{cases} 2, & 16 \le m \le 30, \\ 1, & 2 \le m \le 15 \text{ or } m = 31. \end{cases} \tag{46}$$

B.6 Warm-start lineage and model selection

The shipped model was not trained from scratch in its final 25 epochs. It began as a width-48 model trained on a larger predecessor corpus. It was adapted to the piecewise center and later 3,200/800 corpus, continued once, widened from 48 to 96 hidden units using a function-preserving construction in the spirit of Net2Net [10], with observed output drift of about 1.6×10^{-8}, trained with the mixed downstream-aware objective, and finally returned to direct loss.

The later stages shared one diagnostic: downstream-transported correction fit on the reused 800-network panel, pooled over neurons and averaged over layers 20, 24, and 28. It was neither independent validation nor the challenge score. Whole-estimator tests overruled this proxy twice: A3 epoch 50 and A4 epoch 25 were passed onward even though earlier snapshots had better proxy values. A single smooth training curve would therefore misrepresent the selection process.

P1 used seeds 3000000–3011999 and 3013500–3016499 for gradients, and 3012000–3013199 for its 1,200-network diagnostic. Every MLP in the later 3,200/800 corpus had therefore appeared in the ancestral gradient set.

Table 13: Warm-start lineage. Every trained continuation restarted the optimizer.

Stage / starting point	Change made in this stage	Model passed onward
P1 / scratch	Width 48; 15,000 gradient MLPs; earlier features and targets; direct loss	P1 epoch 145
P2 ← P1 e145	Continued the same corpus and loss at a lower learning rate	P2 epoch 15
A1 ← P2 e15	Switched to the cumulant-aware-through-18/Gaussian-tail target and 3,200/800 corpus	A1 epoch 10
A2 ← A1 e10	Continued the A1 direct objective	A2 epoch 10
W ← A2 e10	Widened hidden state 48 to 96; no training; output drift 1.6×10^{-8}	Widened A2
A3 ← widened A2	Mixed direct and downstream-transported loss	A3 epoch 50
A4 ← A3 e50	Returned to direct loss with a new learning rate and seed	A4 epoch 25 (shipped)

Table 14: Common reused-panel proxy. Higher is better; this is not validation performance or the challenge score.

Stage	Best in stage	Passed-onward fit
P2 (re-evaluated)	not selected here	79.59% (P2 e15)
A1	80.16% (e10)	80.16% (e10)
A2	80.21% (e10)	80.21% (e10)
Exact widening	no training	80.21%
A3	80.26% (e45)	79.81% (e50)
A4	80.32% (e20)	**79.64% (e25, shipped)**

C Supporting component evidence

The following measurements explain smaller choices or steps in the development history, and record the tested changes that *raised* adjusted score and were therefore reverted. They come from different predecessor stacks and evidence classes. They are not final-package marginals and must not be added to the main results. "Not recorded" again means that the same experiment did not preserve that benchmark quantity.

Table 15: Secondary evidence retained for auditability, including reverted score-raising steps. Percent-only raw results are reported as such rather than converted into unsupported absolute scores.

Component / test	Raw MSE	Utilization	Adjusted score	Justified conclusion
Bare → more faithful layer-8 analytic center	$8.72 \times 10^{-7} \rightarrow 6.15 \times 10^{-7}$	31.8% → 52.5%	$2.77 \times 10^{-7} \rightarrow 3.23 \times 10^{-7}$	Better centering lowered raw MSE 29.5%, but its compute cost worsened adjusted score 16.6%; the change was reverted. The sharpest early receipt of the accuracy–cost tension in Section 1.
Early mean recentering, layer 5 → 8	$1.811535 \times 10^{-6} \rightarrow 1.782260 \times 10^{-6}$	not recorded together	not recorded together	The sequential depth sweep supported stopping direct recentering at layer 8; layers 9 and 11 reversed.
Two-energy control, off → on	$5.693497 \times 10^{-7} \rightarrow 5.473606 \times 10^{-7}$	not recorded	not recorded	Raw MSE fell 3.86% on a reduced 20-MLP stack at $N = 32{,}768$; the effect is coupled to the main control strength.
Earlier residual taps, off → on	0.639% lower on a fresh 2,000-MLP panel; 0.317% lower on the board	not validly isolated	not validly isolated	A small raw benefit appeared twice, but utilization drift prevents a score attribution.
Tail mean innovations, off → on	0.236% lower on an 800-MLP panel; 0.191% lower on paired board arms	not recorded	not recorded	A small predecessor gain; a later configuration was neutral, so the final marginal is unresolved.
Mean carry, top 32 → all dropped units	approximately unchanged at 1.6516×10^{-6}	10.29% → 10.22%	$1.700075 \times 10^{-7} \rightarrow 1.687716 \times 10^{-7}$	Removing top-K selection preserved raw error while reducing overhead in successive board versions.
Adaptive sample count and fusions	no isolated result	no isolated result	no isolated result	Present in the submission; no numerical contribution is claimed.

References

[1] A. B. Owen. *Monte Carlo Theory, Methods and Examples.* artowen.su.domains/mc, 2013.

[2] A. Kessy, A. Lewin, and K. Strimmer. Optimal whitening and decorrelation. *The American Statistician*, 72(4):309–314, 2018. doi:10.1080/00031305.2016.1277159.

[3] J. M. Hammersley and K. W. Morton. A new Monte Carlo technique: antithetic variates. *Proceedings of the Cambridge Philosophical Society*, 52:449–475, 1956. doi:10.1017/S0305004100031455.

[4] B. D. Tracey and D. H. Wolpert. Reducing the error of Monte Carlo algorithms by learning control variates. arXiv:1606.02261, 2016.

[5] V. Strassen. Gaussian elimination is not optimal. *Numerische Mathematik*, 13:354–356, 1969. doi:10.1007/BF02165411.

[6] W. Wu, V. Lecomte, M. Winer, G. Robinson, J. Hilton, and P. Christiano. Estimating the expected output of wide random MLPs more efficiently than sampling. arXiv:2605.05179, 2026.

[7] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Empirical Methods in Natural Language Processing*, 2014. arXiv:1406.1078.

[8] D. Hendrycks and K. Gimpel. Gaussian error linear units (GELUs). arXiv:1606.08415, 2016.

[9] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. arXiv:1412.6980.

[10] T. Chen, I. Goodfellow, and J. Shlens. Net2Net: Accelerating learning via knowledge transfer. In *International Conference on Learning Representations*, 2016. arXiv:1511.05641.

[11] A. Karpathy. Autoresearch: an autonomous edit–evaluate–retain research loop. github.com/karpathy/autoresearch, 2026.