SLIDING NETWORKS: A FAMILY OF RESIDUAL STRUCTURES PRESERVING FEATURE LEARNING AND FEATURE DIVERSITY

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep learning models tend to perform best at large widths and depths. The growing interest in scaling laws, especially in the language modeling domain, has motivated procedures for scaling up network widths and depths in order to preserve stability and computational efficiency of training dynamics. Existing depth-scaling approaches force a choice between retaining locally maximal *feature learning* and preserving a form of *feature diversity* in the residual stream arising from random differences between channels. We examine a *joint scaling recipe* that preserves both. With residual width $N_\flat$, hidden width $N_\sharp$, and depth L, the key quantity is, by default, the limiting ratio $N_\flat/(N_\sharp L) \to \alpha$, while correlated initialization (Half Copy) can partially relax the resulting shape constraints. Theory and residual-MLP experiments support convergence at fixed positive α to an SDE limit with locally maximal feature learning. Hidden computations can then be redistributed between width and depth while preserving this limit, like turning melody into harmony and back. Language-model experiments nevertheless show performance differences under regrouping at fixed parameter count. These results distinguish approximating a limit (ODE or SDE) from scaling for better performance.

1 INTRODUCTION

1.1 MOTIVATION AND PRIOR WORK

Deep learning has driven major advances in artificial intelligence. This approach takes its name from the composition of multiple layers of trainable transformations (see LeCun et al., 2015, for a review). Architectural landmarks such as AlexNet (Krizhevsky et al., 2012), ResNet (He et al., 2016), and the Transformer (Vaswani et al., 2017) helped establish the foundations of modern models, including large language models (see Zhao et al., 2026, for a survey).

As models grow larger, choices of depth, width, and optimization procedure become increasingly important because they largely govern the performance obtained from a given computational budget (Tay et al., 2022; Petty et al., 2024; Brown et al., 2022). Quantitative studies of this relationship have yielded empirical scaling laws linking performance to model size, data volume, and training compute (Kaplan et al., 2020). These laws have been used to guide the allocation of training budgets for larger models (Hoffmann et al., 2022). However, parameter count, whether measured at training or inference time, is only one of many factors shaping training dynamics and final performance (Sutskever et al., 2013). Exhaustively optimizing the other architectural and training choices at every model size is impractical (Bergstra & Bengio, 2012), so comparisons by parameter count remain conditional on the configurations explored.

Theory provides tools for understanding and classifying how parameterization and training hyperparameters affect learning as networks grow. For example, as network size grows under NTK scaling, changes in the activations of individual hidden neurons decay, while the network's output prediction dynamics approach those described by a fixed kernel (Jacot et al., 2018; Lee et al., 2019). Mean-field / μP scalings instead retain constant scale updates to these hidden features in two-layer (Mei et al., 2018; Chizat & Bach, 2018; Sirignano & Spiliopoulos, 2020) and deeper networks (Yang & Hu, 2021; Bordelon & Pehlevan, 2022). Parameterizations that preserve feature learning

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

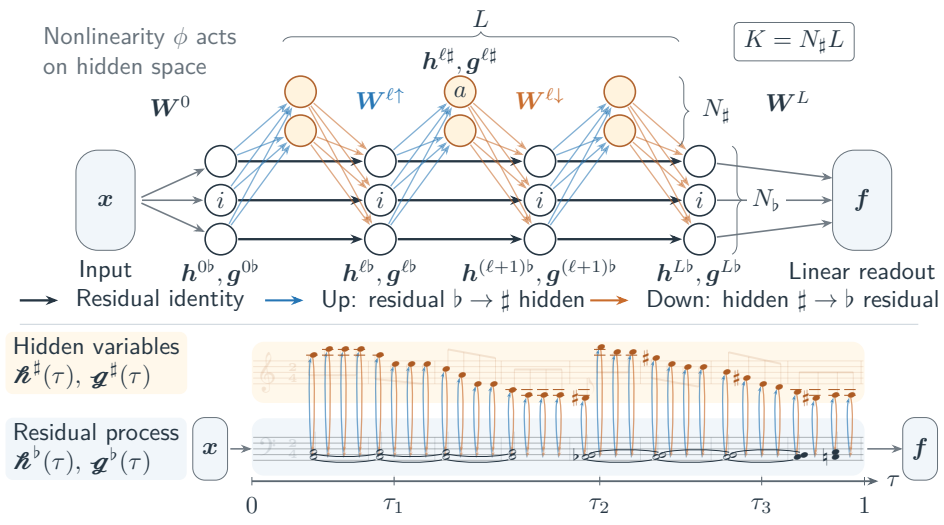


Figure 1: Upper: a standard residual MLP structure. Lower: the continuous-depth picture. Residual coordinates evolve through depth, while hidden units contribute at specific depths. The symbols $\flat$ and $\sharp$ distinguish the two parts. h denotes preactivations and g rescaled gradients (Jacobian).

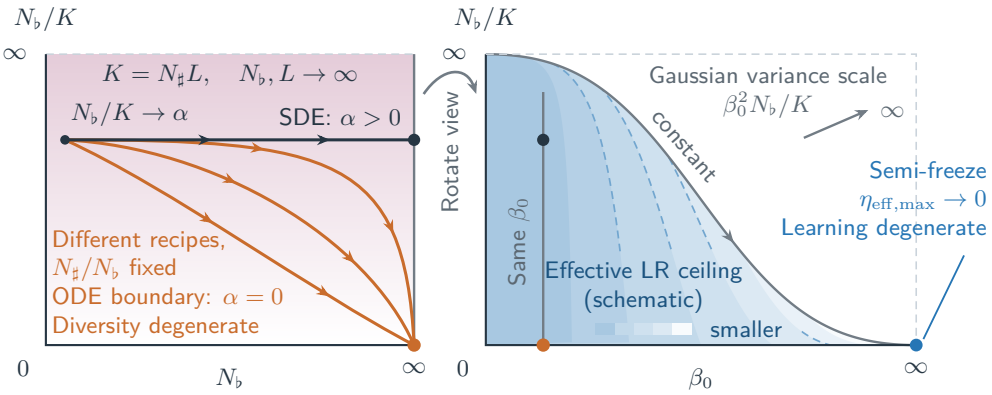


Figure 2: Left: growth paths and their limits. Right: the coupling–learning tradeoff within the limiting family. Colors indicate the variance of the random depth contribution on the left and the effective learning-rate ceiling on the right. The color trends and contours are schematic.

can help hyperparameter settings tuned on smaller models remain effective as model size increases, providing a systematic route to training and understanding larger networks (Yang et al., 2021; Dey et al., 2025).

However, for residual architectures such as Transformers, a large depth scaling limit that preserves feature learning does not by itself guarantee better performance and may suppress a form of feature diversity present at finite size. We use the MLP residual structure in a Transformer as a simple example (illustrated in the upper panel of Fig. 1), with attention temporarily set aside [1]. Here, the residual stream has width $N_\flat$, each block has hidden width $N_\sharp$, and the network contains L such blocks. In practice, one common choice is $N_\sharp = 4N_\flat$ (Vaswani et al., 2017). The contributions of hidden neurons accumulate in the residual stream as hidden width and depth increase. Controlling this accumulation is already necessary for stable, nondegenerate signal propagation at initialization (Schoenholz et al., 2017; Hayou et al., 2021; Noci et al., 2023). Scaling down each block's coupling

[1] Attention has the same projection structure when each head is treated as one finite-dimensional nonlinear channel, corresponding to an orange circle in Fig. 1; see Appendix B.5.

to the residual stream, together with corresponding learning-rate adjustments, can stabilize training while retaining feature learning, and yield a continuous-depth description for sufficiently large L (Bordelon et al., 2024b; Chizat, 2025). Yet the accompanying averaging may diminish the diversity of learned residual features.

The ratio of residual width to the total hidden population helps explain this tension between preserving feature learning and feature diversity: $N_\flat/K$, where $K = N_\sharp L$. The common choice $N_\sharp \asymp N_\flat \asymp N$ forces $N_\flat/K \to 0$ as $L \to \infty$, regardless of how N grows relative to L (Fig. 2, left). In this setting, Yang et al. (2024) already pointed out the dilemma that preserving feature learning requires stronger averaging over K, which washes out random depth variation in residual trajectories and yields an ODE-like limit that they regard as redundant because it loses feature diversity. Allowing the three sizes to vary independently, Chizat (2025) and Chaintron et al. (2026) rigorously establish this ODE regime when the ratio vanishes, while Jiang et al. (2026) develop related scaling recipes for mixture-of-experts networks, where total hidden capacity differs from the capacity activated per token. These works also point to non-ODE dynamics at nonzero corresponding ratios (Chaintron et al., 2026; Jiang et al., 2026).

1.2 Our perspective and main findings

Our theory examines network scaling through an SDE description (Theoretical Result 1) that retains random depth variation in residual trajectories (Fig. 3, left), with the ODE description recovered when this variation vanishes. In this description, residual coordinates follow random paths through depth, while individual hidden features learn at their respective depths (Fig. 1, lower). For SGD and Muon (Jordan et al., 2024a), with the latter understood as a spectral transform, the updates do not privilege a residual coordinate frame, allowing the residual statistics to remain jointly Gaussian, even though nonlinearities can make hidden features strongly non-Gaussian (Fig. 3, right).

One natural question is whether the training recipe can compensate for the effect of $N_\flat/K$ in the limiting dynamics, allowing more freedom on network shapes, such as $N_\flat$ and $N_\sharp$ proportional considered in Yang et al. (2024). Increasing the residual coupling strength β_0 can keep the random variance scale $\beta_0^2 N_\flat/K$ finite and nonzero at initialization as $N_\flat/K \to 0$ (Theoretical Result 2, first part). However, even one GD step in a linear network shows that this also amplifies learning feedback (Theoretical Result 2, second part). Controlling the size of residual updates can then force the effective learning rate for individual hidden features toward zero, approaching the semi-freeze boundary in the right panel of Fig. 2 (Appendix C.2.7). More structured recipes, such as correlated initialization, can instead preserve both feature learning and random depth increments.

Another question is whether it matters how $K = N_\sharp L$ is factored into $N_\sharp$ and L, as our limiting description suggests that only their product is essential. In the ODE regime, this width–depth interchange follows naturally from the framework of Chizat (2025). Theoretical Result 3 supports this picture by identifying the contributions to the finite-size error. Within this approximation, the same ordered hidden computations can be combined into wider blocks or spread across more layers, like turning melody into harmony and back. We call this operation *sliding*.

Measurements on practical networks such as GPT-2 show that contributions from randomly initialized residual blocks can dominate the input-embedding at initialization, making their scale important for describing finite networks. Our fixed α scaling limit balances both of these contributions across scales. Our *Half Copy* recipe uses correlated initialization within conventional shapes to achieve a similar effect. In the CIFAR-10 and Tiny Shakespeare comparisons, we conservatively estimate savings of 20% of total parameters for SDE and 40% for Half Copy when matching or improving the ODE recipe's mean training loss after fixed training schedules (Appendix H.4). These gains do not settle the choice of depth: at matched parameter budgets, using *sliding* to combine computations into one wide block gives the lowest training loss among the tested Tiny Shakespeare shapes.

2 Joint scaling and limiting dynamics

We study how residual structure shapes learning, focusing on the residual representation width $N_\flat$ and the associated population of $K = N_\sharp L$ nonlinear channels. These channels are arranged into L layers, with $N_\sharp$ channels per layer. The MLP structure in a Transformer provides a concrete example (Fig. 1, upper panel). Initially omitting biases and normalization, we write its computation

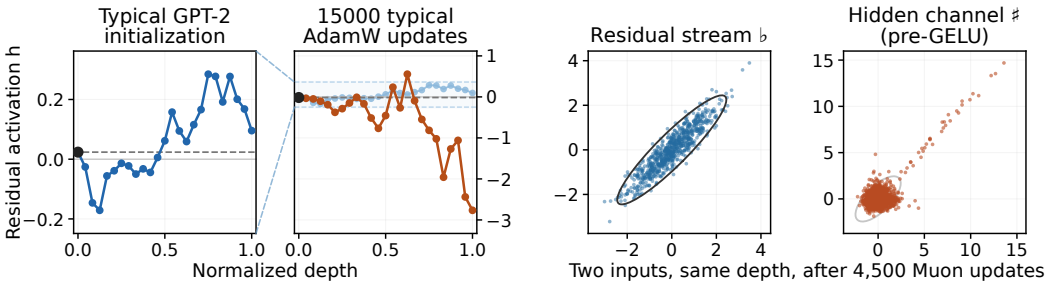


Figure 3: Selected GPT-2 activations (Appendix I). Left: one residual coordinate through depth. Right: paired standardized activations for two inputs; ellipses show covariance-matched Gaussian 95% contours.

as pseudocode on the left and in rescaled mathematical form on the right (`W` absorbs the displayed scaling factors before $\boldsymbol{W}$, while $\boldsymbol{W}$ is initialized with independent standard Gaussian entries):

```
h_flat[0] = W_in @ x

h_sharp[l] = W_up[l] @ h_flat[l]

h_flat[l+1] = (h_flat[l]
  + W_down[l] @ phi(h_sharp[l]))

f = W_out @ h_flat[L]
```

$$
\begin{aligned}
\boldsymbol{h}^{0\flat} &= \frac{1}{\sqrt{D_{\mathrm{i}}}}\boldsymbol{W}^{0}\boldsymbol{x}, \\
\boldsymbol{h}^{\ell\sharp} &= \frac{1}{\sqrt{N_\flat}}\boldsymbol{W}^{\ell\uparrow}\boldsymbol{h}^{\ell\flat}, \\
\boldsymbol{h}^{(\ell+1)\flat} &= \boldsymbol{h}^{\ell\flat} + \frac{\beta_0\sqrt{N_\flat}}{LN_\sharp}\boldsymbol{W}^{\ell\downarrow}\phi\big(\boldsymbol{h}^{\ell\sharp}\big), \\
\boldsymbol{f} &= \frac{1}{\gamma_0 N_\flat}\boldsymbol{W}^{L}\boldsymbol{h}^{L\flat}.
\end{aligned}
\tag{1}
$$

We hold the input and output dimensions D_{i} and D_{o}, the data, loss, and batch sequence fixed and compare networks at the same training step t. Token indices and the internal coordinates of each attention head likewise provide additional axes orthogonal to the network size $N_\flat, N_\sharp, L$ varying. With a tensor interpretation of the variables and maps, the same residual structure also accommodates attention and other nonlinear channel computations (Appendix A.1).

The output $\boldsymbol{f}$ is a macroscopic observable with fixed, task-defined coordinates. Internal coordinates have no such prescribed roles. Identity connections carry the same residual coordinates through depth, whereas each layer introduces a new set of hidden coordinates. A common rotation of the residual basis, accompanied by the corresponding weight transformations, leaves the computation unchanged; coordinatewise hidden nonlinearities instead select a preferred basis. We track backward propagation through the rescaled Jacobians of each output component:

$$
\boldsymbol{g}_q^{\ell\flat} = \gamma_0 N_\flat \frac{\partial f_q}{\partial \boldsymbol{h}^{\ell\flat}}, \qquad \boldsymbol{g}_q^{\ell\sharp} = \frac{\gamma_0 K}{\beta_0}\frac{\partial f_q}{\partial \boldsymbol{h}^{\ell\sharp}}, \quad q = 1, \ldots, D_{\mathrm{o}}. \tag{2}
$$

These factors compensate for the decreasing size of individual output derivatives as the network grows. Table 1 gives the initialization and optimizer scalings for the code weights, likewise compensating for network growth to retain order-one hidden-feature ($\boldsymbol{h}$ and $\boldsymbol{g}$) updates and their collective effect on the output. Optimizer conventions and the conversion to mathematical weights are given in Appendix A.

As networks grow, statistics across the many coordinates of $\boldsymbol{h}^\flat, \boldsymbol{h}^\sharp, \boldsymbol{g}^\flat, \boldsymbol{g}^\sharp$ are captured by the joint probability law of representative residual trajectories $(\boldsymbol{\mathcal{h}}^\flat, \boldsymbol{\mathcal{g}}^\flat)$ and hidden variables $(\boldsymbol{\mathcal{h}}^\sharp, \boldsymbol{\mathcal{g}}^\sharp)$ at each depth. The residual trajectories remain discrete at fixed L and become continuous trajectories indexed by $\tau = \ell/L \in [0, 1]$ when width and depth grow together (Appendix B.3.3).

Theoretical Result 1 (Optimizer-dependent limiting dynamics) *Consider networks trained with SGD, Muon, or Adam, using the scalings in Table 1 and Appendix A, keeping the remaining initialization and learning-rate constants fixed. As $N_\flat, L \to \infty$ and $N_\flat/K \to \alpha$, the training dynamics converge to a continuous-depth limit that generally depends on the optimizer. For Muon, this limit*

Quantity	Input	Up	Down	Output
Initialization std.	1	$1/\sqrt{N_\flat}$	$\sqrt{N_\flat}/K$	$1/N_\flat$
SGD learning rate	$N_\flat$	$K/N_\flat$	$N_\flat/K$	$1/N_\flat$
Adam learning rate	1	$1/N_\flat$	$1/K$	$1/N_\flat$
Muon learning rate	$\sqrt{N_\flat}$	$\sqrt{N_\sharp/N_\flat}$	$\sqrt{N_\flat N_\sharp}/K$	$1/\sqrt{N_\flat}$
Muon clipping threshold	$1/\sqrt{N_\flat}$	$\sqrt{N_\flat N_\sharp}/K$	$\sqrt{N_\sharp/N_\flat}$	$\sqrt{N_\flat}$

Table 1: Size dependence of initialization and updates for code weights $\mathtt{W}$ in Eq. 1, with $K = N_\sharp L$. Fixed task dimensions and multipliers, including β_0, γ_0, are omitted. Here, Muon is understood as a normalized spectral-clipping map defined in Appendix A.

additionally requires $N_\sharp \to \infty$. In each case, the variables $(\boldsymbol{\hbar}^\flat, \boldsymbol{g}_q^\flat, \boldsymbol{\hbar}^\sharp, \boldsymbol{g}_q^\sharp)$ satisfy optimizer-dependent self-consistent equations (system (60) or its optimizer- and model-specific variants in Appendix B), and the limiting macroscopic statistics (such as the output $\boldsymbol{f}$) are deterministic.

For SGD and Muon, the residual variables $(\boldsymbol{\hbar}^\flat, \boldsymbol{g}_q^\flat)$ remain jointly Gaussian across depth, inputs, and training steps. Adam does not generally preserve residual Gaussianity.

The statistical distinctions described above are visible in standard models. Figure 3 provides an illustration (detailed measurements follow in the experimental section): its right panel shows approximately Gaussian residual alongside non-Gaussian hidden pre-activations. Full equations can be developed along the lines of the dynamical mean-field formulation of Bordelon & Pehlevan (2022), or equivalently the tensor-program formulation of Yang & Hu (2021). We leave those equations, including their tensor formulation for Transformer attention, to Appendix B and present here only the minimal calculations needed to explain our picture.

Theoretical Result 2 (Zero- and one-step checkpoints) *At initialization, the limiting residual trajectory is a continuous Gaussian random walk through depth with independent increments. For the residual MLP structure used in Transformers, let $C_h(\tau; t)$ denote the limiting residual variance for a fixed input. With $c = \beta_0^2 \alpha$, the diffusion intensity is*

$$\partial_\tau C_h(\tau; 0) = \begin{cases} c\,\mathbb{E}[\phi(\sqrt{C_h(\tau;0)}Z)^2], & \textit{without RMSNorm}, \\ c\,\mathbb{E}[\phi(Z)^2], & \textit{with unit pre-RMSNorm}, \ \epsilon_{\mathrm{norm}} = 0. \end{cases} \tag{3}$$

Here $Z \sim \mathcal{N}(0,1)$, and $C_h(0;0) = \|\boldsymbol{x}\|^2/D_{\mathrm{i}}$ is the variance contributed by the input embedding; random residual increments supply the subsequent increase. For $\phi = \mathrm{ReLU}$, these give $C_h(\tau;0) = C_h(0;0)e^{c\tau/2}$ and $C_h(\tau;0) = C_h(0;0) + c\tau/2$, respectively (Appendix C.1).

For its linear specialization ($\phi = \mathrm{Id}$, without normalization), take $\|\boldsymbol{x}\|^2 = D_{\mathrm{i}}$, loss $\frac{1}{2}(f-1)^2$, and frozen input and readout maps. Starting from $C_h(\tau;0) = e^{c\tau}$, one simultaneous SGD step on both residual projections with effective local learning rate η_{eff} (Appendix C.2) gives

$$C_h(\tau;1) = e^{c\tau} + \eta_{\mathrm{eff}}^2 e^{c(1+\tau)} \left[c\tau + 2(2\beta_0^2 + c^2)\tau^2 \left(1 + \frac{2c\tau}{3}\right)\right]. \tag{4}$$

Figure 4 compares full finite-network updates with (4), showing close numerical agreement.

At initialization, comparing the contribution of Gaussian variance provides a practical distinction between ODE-like behavior, where the input contribution dominates, and SDE-like behavior, where residual random increments contribute substantially. Standard GPT-2 models clearly favor the latter (Fig. 3, left; Karpathy, 2024). Increasing β_0 as α decreases can preserve $c = \beta_0^2\alpha > 0$ and hence the initialization variance, but not the training dynamics: even the first-step energy change grows as $(\beta_0 \eta_{\mathrm{eff}})^2$. Controlling it requires $\eta_{\mathrm{eff}} = O(\beta_0^{-1})$ (the decreasing learning-rate ceiling illustrated in Fig. 2 (right)), weakening direct local hidden-feature learning. The calculation and finite-size checks are in Appendix C; the general nonlinear-channel description of this degeneration is given in Appendix B.5.1.

At fixed $N_\flat/K$, the same ordered hidden computations can be regrouped across width and depth (*sliding*), much like changing the step size of an Euler discretization. For example, label the projection matrices of two consecutive layers by 1 and 2. Gluing them stacks the up projections and

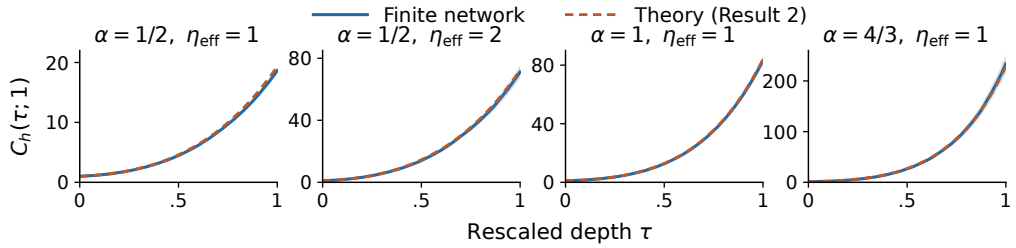


Figure 4: One-step variance: finite-network means (solid) versus Result 2 (dashed), with ± 2 Monte Carlo SE shaded; $\beta_0 = 1$, $L = 128$ (Appendix C.2.9).

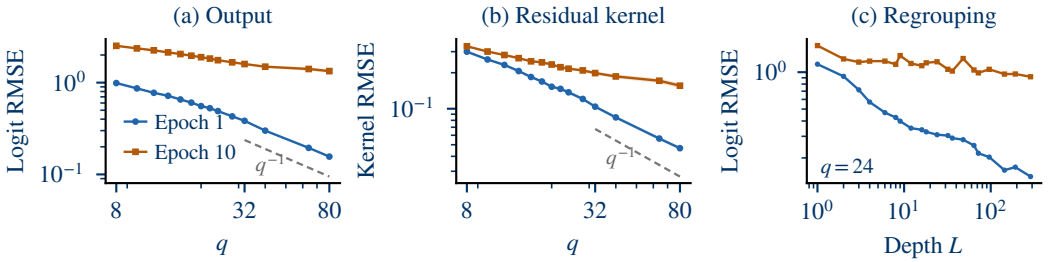


Figure 5: CIFAR-10 convergence: output and final-residual cosine-kernel errors against $q = 96$ ensemble means (left/middle; bands: ± 2 conditional SE; dashed: q^{-1}), and coupled regrouping against the finest finite partition at $N_\flat = K = 24^2$ (right).

concatenates the down projections, so both groups read the same residual state:

$$\widetilde{\boldsymbol{h}}^\flat = \boldsymbol{h}^\flat + \frac{\beta_0 \sqrt{N_\flat}}{K} \begin{bmatrix} \boldsymbol{W}_1^\downarrow & \boldsymbol{W}_2^\downarrow \end{bmatrix} \phi\left(\frac{1}{\sqrt{N_\flat}} \begin{bmatrix} \boldsymbol{W}_1^\uparrow \\ \boldsymbol{W}_2^\uparrow \end{bmatrix} \boldsymbol{h}^\flat \right). \tag{5}$$

Pairwise gluing doubles $N_\sharp$ and halves L, leaving K and the coupling factor unchanged. This regrouping of the same ordered weights is reversible. Finite-size errors can be understood as

$$\begin{matrix} \text{Residual-kernel} \\ \text{fluctuations} \end{matrix} + \begin{matrix} \text{Euler} \\ \text{discretization} \end{matrix} + \begin{matrix} \text{Hidden-population} \\ \text{fluctuations} \end{matrix} + \begin{matrix} \text{Spectral bias} \\ \text{(Muon).} \end{matrix}$$

The following result gives their combined scales.

Theoretical Result 3 (Finite-size error scales) *At a fixed training step, the root-mean-square error of a macroscopic observable, such as the output $\boldsymbol{f}$, relative to its optimizer's limit at $\alpha = N_\flat / K$ is*

$$\textit{SGD and Adam:} \quad O\left(\frac{1}{\sqrt{N_\flat}} + \frac{1}{L} + \frac{1}{\sqrt{N_\sharp L}} \right), \qquad \textit{Muon:} \quad O\left(\frac{1}{\sqrt{N_\flat}} + \frac{1}{L} + \frac{1}{N_\sharp} \right). \tag{6}$$

Constants depend on the task, training step, and optimizer settings. Details are in Appendix D.

3 Experiments

3.1 Do the networks converge to the predicted limit?

Output and kernel comparisons provide effective empirical checks of a proposed scaling limit (Bordelon & Pehlevan, 2022; 2023; Bordelon et al., 2024c;b). We use CIFAR-10 residual MLPs because their scalar hidden units permit flexible regrouping, allowing large depths at modest parameter counts. Under joint growth with $N_\flat = q^2$ and $N_\sharp = L = q$, output and final-residual cosine-kernel errors relative to the largest-network ensemble decrease with q, both early and later in training (Fig. 5, left and middle). Along this family, Theoretical Result 3 predicts $O(q^{-1})$ errors. At epoch

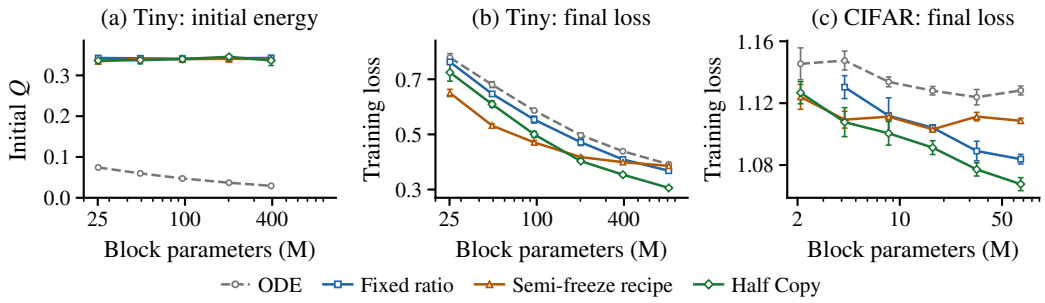


Figure 6: Larger-scale comparisons, counting block weights only. Tiny: initial Q and loss after 5,000 Adam steps; CIFAR-10: loss after five SGD epochs. Means $\pm$ sample SD over three seeds; full ranges: Appendices G.2, H.

one, slopes fitted over $q = 22$–80 are -0.95 and -0.90, respectively, quantitatively consistent with this rate at short training times. Here we test the reverse of the gluing in Eq. 5: wider blocks are split into more, narrower layers, keeping $N_\flat = K$ and the ordered hidden computations fixed. Successive groups then read updated residual states. Output discrepancies from the finest finite partition generally decrease with depth (Fig. 5, right). Here $N_\flat = K$ is fixed, so the sampling terms in our error estimate do not decrease with depth, and a clean L^{-1} discretization rate need not be visible. Further checks and protocols are given in Appendix G.1.

3.2 What should a scaling recipe preserve?

A scaling recipe should account for both the magnitude of the Gaussian residual contribution (feature variance) and the effective local learning rate (feature learning). We compare scaling routes for Tiny Shakespeare Transformers and CIFAR-10 residual MLPs, starting from a common small-network anchor within each task, tuned one hyperparameter at a time. Full protocols are given in Appendices G.2 and H. While our computational budget limits large-scale sweeps, measurements from individual GPT-2 runs using initialization and learning-rate settings in Karpathy (2024) without re-tuning also support the SDE picture (Fig. 7). At initialization, the cumulative squared residual increments Q are about one hundred times the input-embedding energy E_0: random depth variation is a dominant contribution. After training, the squared total displacement exceeds Q by a substantial positive cross-depth term C, indicating learned correlations alongside the still-large increment energy. A useful statistical description must account for both.

The ODE recipe follows Table 1 while increasing both widths and depth by a factor R, so $N_\flat/K$ decreases as R^{-1}. The semi-freeze recipe uses the same shapes but increases β_0 by $\sqrt{R}$ to compensate for this loss of initial random variance, at the cost of reducing the effective learning rate of individual hidden features, as illustrated by Theoretical Result 2. The SDE recipe instead keeps the coupling and learning-rate constants fixed and reallocates dimensions to preserve $N_\flat/K$. For Tiny Shakespeare, growth factors $(R^{3/2}, R^{1/2}, R)$ for $(N_\flat, N_\sharp, L)$ match both the ODE depth and its R^3 residual-block parameter budget. Attention heads follow the same allocation principle at fixed head dimension; task-specific geometries are given in Appendices G.2 and H.

Despite its unusual width–depth allocation, the SDE recipe achieves lower mean training loss than ODE on both tasks at larger scales (Fig. 6). Even in the displayed examples, where semi-freeze initially leads, continued growth reverses its advantage over SDE. The CIFAR curve additionally shows an increase in loss with model size.

No single budget can match every resource across differently shaped networks. We use residual-block parameter count as our common comparison axis, matching budgets proportional to R^3 and counting both MLP and attention projections in Transformers. For the Transformer allocation above, ignoring integer rounding gives the following ratios for $R > 1$ at fixed batch size and precision:

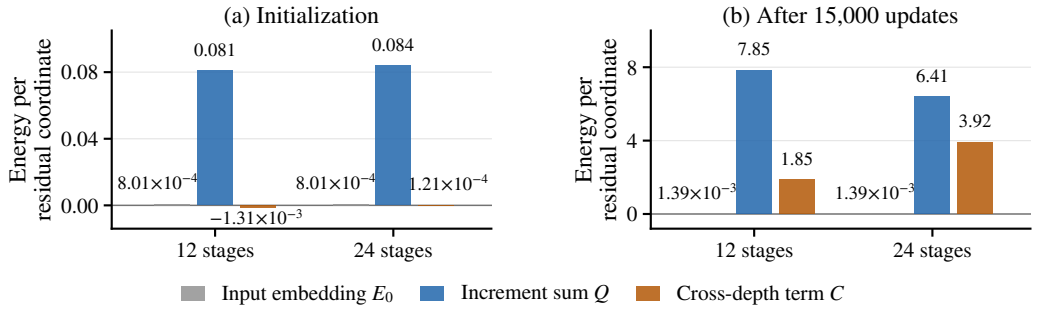


Figure 7: Residual energy before and after training. For complete attention–MLP stage increments $\Delta \boldsymbol{h}_\ell$, $E_0 = \|\boldsymbol{h}_0\|^2/N_\flat$, $Q = \sum_\ell \|\Delta \boldsymbol{h}_\ell\|^2/N_\flat$, and $C = \|\boldsymbol{h}_L - \boldsymbol{h}_0\|^2/N_\flat - Q$. Each quantity is computed per token and averaged over the same four 1024-token sequences. The signed cross-depth term C is retained; the panels use different vertical scales (Appendix I.2).

Resource	SDE / ODE (or semi-freeze)	Higher cost
Residual-block parameters	1	Equal
Input/output parameters and weight storage	$\sqrt{R}$	SDE
KV cache at fixed sequence length	$1/\sqrt{R}$	ODE

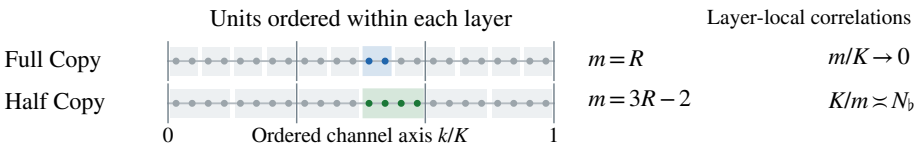
Endpoint parameters become a smaller fraction of the total as scale grows. In the largest Tiny comparison, rounding even leaves SDE with slightly fewer total parameters (814.81M versus 815.22M), yet its loss is lower (0.3680 versus 0.3916). The KV-cache difference need not be small. The corresponding result in residual MLPs, which have no attention or KV cache, shows that the tendency is not confined to token-based models.

We propose the *Half Copy* recipe to retain conventional network shapes while maintaining both the initial Gaussian residual contribution and the effective local learning-rate scale. In both CIFAR-10 and Tiny Shakespeare, it outperforms the fixed-ratio SDE and ODE recipes (Fig. 6). Against ODE, geometry, coupling, and learning-rate constants are unchanged: this isolates initialization from shape and opens another direction for scaling-recipe design.

Consider doubling $N_\sharp$, and hence K, at fixed $N_\flat, L, \beta_0$. Copying each hidden unit's up- and down-projections within its layer exactly cancels the halved $1/K$ coefficient. This preserves the random residual variance, but only trivially: twice as many hidden parameters compute exactly the same function. Half Copy instead uses larger, partly correlated groups. For a doubling, four units with identical up-projections and down-projection correlation $\rho = 1/3$ supply the same variance factor as two identical copies:

$$U_a = U_{p(a)}, \qquad V_a = \sqrt{\rho}\, Z_{p(a)} + \sqrt{1-\rho}\, E_a, \qquad \rho = \frac{1}{3}. \tag{7}$$

Here $p(a)$ labels a layer-local group; parent weights and innovations are independent standard Gaussians. Each weight keeps its Gaussian marginal, while independent innovations allow the separately trainable copies to develop distinct features (Appendix F.4). With both widths and depth growing by a factor R and the scalings of Table 1, groups of approximately $3R$ units compensate for the falling $N_\flat/K$; attention heads are grouped analogously. Correlations remain local to each layer, while the total number of independent parent groups grows with $N_\flat$ (Appendix E).



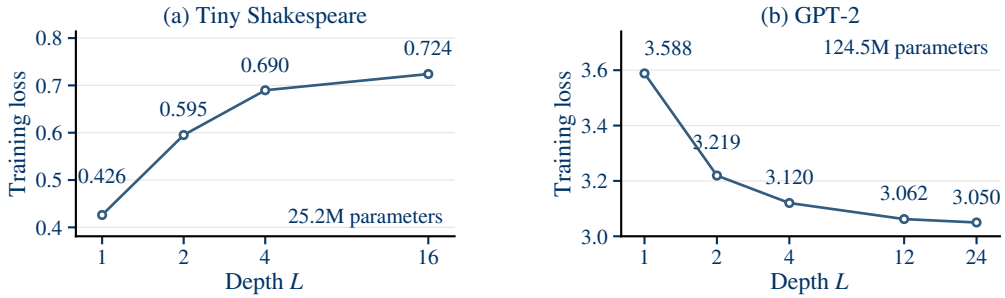


Figure 8: Width–depth allocation and finite performance. Left: Tiny Shakespeare training loss on 512 fixed sequences at a fixed parameter budget, after 5,000 Adam updates (Appendix H.5). Right: GPT-2 training loss averaged over the final 100 updates at matched parameter and token budgets (Appendix I.1).

3.3 What shape is actually best? Depends on the task

Chizat (2025) derives sharp error estimates for approximating the ODE limit, offering a way to guide width–depth allocation. Our estimate (Theoretical Result 3) invites the same exercise: balancing its three contributions under a compute budget $C \propto N_\flat N_\sharp L$ suggests a scaling recipe,

$$\text{error proxy:} \quad \frac{1}{\sqrt{N_\flat}} + \frac{1}{L} + \frac{1}{\sqrt{N_\sharp L}}, \qquad \text{balanced recipe:} \quad N_\sharp \asymp L \asymp \sqrt{N_\flat}. \tag{8}$$

But a smaller approximation error need not buy a lower loss, and our SDE picture gets no exemption. Tiny Shakespeare supplies the punchline: in the single-seed Half Copy scan, *sliding* to one block gives the lowest training loss at fixed parameter budget, with layer-local correlation groups rebuilt at each depth (Fig. 8, left). There is no contradiction: shallow, wide networks have their own limits, obtained at fixed depth along the way to our continuous-depth description (Appendix B). Continuous depth has no automatic claim to better performance. Nor does one block: in our GPT-2 experiment, twelve or twenty-four stages substantially outperform one at matched parameter and token budgets (Fig. 8, right). Although we have some theoretical guidance and intuition for when random residual variation makes an SDE description useful, we still lack a theory, or even a reliable intuition, for when depth should beat width.

4 Conclusion

We identify joint width–depth scalings that retain local hidden-feature learning together with random residual variation. Our measurements show that this variation can be a major contribution to residual features at initialization. Half Copy illustrates how initialization correlations can offer an alternative to unusual width–depth allocations.

Our analysis keeps the data and training horizon fixed as model size grows and centers on dense MLP residual blocks. Extending it to sparse mixture-of-experts models may build on the scaling analysis of Jiang et al. (2026). Training times growing with model size may require more than limiting dynamics and leading finite-size corrections (Bordelon & Pehlevan, 2023; Bordelon et al., 2024a). Our compute budget also limits the experimental scope; substantially larger-scale comparisons are needed to assess whether these ideas lead to more compute-efficient Transformer training.

Limits provide a language for comparing network statistics across sizes, but neither closed equations nor convergence guarantee better finite-model performance. Our SDE family faces the same test as ODE limits. Finite-network behavior should guide which statistics a useful limit retains, rather than be altered merely to fit a convenient limit. This perspective calls for initialization and scaling recipes that help finite models approach useful limits efficiently, with their value assessed through finite-model performance.

REPRODUCIBILITY STATEMENT

The appendices give the model, optimizer, initialization, data, and measurement conventions for each experiment. Numerical data accompany the displayed curves and tables. Controlled calculations include independent finite-step checks, and empirical comparisons specify their seeds, finite references, and uncertainty summaries. The language-model distribution measurements use fixed probes across training, with raw residual and output-Jacobian vectors retained separately from their standardized plots.

STATEMENT ON AI ASSISTANCE

AI assistants were used for literature checks, drafting and revising proofs, mathematical and numerical verification, experiment design, implementation and analysis, figure generation, translation, and manuscript preparation. They also supported long-running experiments through periodic automated check-ins ("heartbeats") and unattended workflows, including progress monitoring, fault diagnosis and recovery, and checkpoint preservation, within author-defined protocols and resource budgets. The authors are responsible for the claims, interpretation, and submitted manuscript.

REFERENCES

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016. URL https://arxiv.org/abs/1607.06450.

James Bergstra and Yoshua Bengio. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, 13(10):281–305, 2012. URL https://jmlr.org/papers/v13/bergstra12a.html.

Blake Bordelon and Cengiz Pehlevan. Self-consistent dynamical field theory of kernel evolution in wide neural networks. In *Advances in Neural Information Processing Systems*, volume 35, pp. 32240–32256, 2022. URL https://arxiv.org/abs/2205.09653.

Blake Bordelon and Cengiz Pehlevan. Dynamics of finite width kernel and prediction fluctuations in mean field neural networks. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/1ec69275e9f002ee068f5d68380f3290-Abstract-Conference.html.

Blake Bordelon, Alexander Atanasov, and Cengiz Pehlevan. A dynamical model of neural scaling laws. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 4345–4382, 2024a. URL https://proceedings.mlr.press/v235/bordelon24a.html.

Blake Bordelon, Hamza Tahir Chaudhry, and Cengiz Pehlevan. Infinite limits of multi-head transformer dynamics. In *Advances in Neural Information Processing Systems*, volume 37, 2024b. doi: 10.52202/079017-1130. URL https://papers.neurips.cc/paper_files/paper/2024/hash/3eff068e195daace49955348de9f8398-Abstract-Conference.html.

Blake Bordelon, Lorenzo Noci, Mufan Li, Boris Hanin, and Cengiz Pehlevan. Depthwise hyperparameter transfer in residual networks: Dynamics and scaling limit. In *International Conference on Learning Representations*, 2024c. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/5f5f7b6080dcadced61cf5d96f7c6dde-Abstract-Conference.html.

Jason Ross Brown, Yiren Zhao, Ilia Shumailov, and Robert D. Mullins. Wide attention is the way forward for transformers?, 2022. URL https://arxiv.org/abs/2210.00640. Presented at the NeurIPS 2022 Workshop on All Things Attention: Bridging Different Perspectives on Attention.

Louis-Pierre Chaintron, Lénaïc Chizat, and Javier Maass. ResNets of all shapes and sizes: Convergence of training dynamics in the large-scale limit, 2026. URL https://arxiv.org/abs/2603.18168.

Lénaïc Chizat. The hidden width of deep ResNets: Tight error bounds and phase diagram, 2025. URL https://arxiv.org/abs/2509.10167.

Lénaïc Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In *Advances in Neural Information Processing Systems*, volume 31, 2018. URL https://arxiv.org/abs/1805.09545.

Nolan Dey, Bin Claire Zhang, Lorenzo Noci, Mufan Li, Blake Bordelon, Shane Bergsma, Cengiz Pehlevan, Boris Hanin, and Joel Hestness. Don't be lazy: CompleteP enables compute-efficient deep transformers. In *Advances in Neural Information Processing Systems*, 2025. URL https://papers.nips.cc/paper_files/paper/2025/hash/c917d8b9e01427f3184d80ade22f4d1f-Abstract-Conference.html.

Soufiane Hayou, Eugenio Clerico, Bobby He, George Deligiannidis, Arnaud Doucet, and Judith Rousseau. Stable ResNet. In *Proceedings of The 24th International Conference on Artificial Intelligence and Statistics*, volume 130 of *Proceedings of Machine Learning Research*, pp. 1324–1332, 2021. URL https://proceedings.mlr.press/v130/hayou21a.html.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016. URL https://openaccess.thecvf.com/content_cvpr_2016/html/He_Deep_Residual_Learning_CVPR_2016_paper.html.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs), 2016. URL https://arxiv.org/abs/1606.08415.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, Thomas Hennigan, Eric Noland, Katherine Millican, George van den Driessche, Bogdan Damoc, Aurelia Guy, Simon Osindero, Karén Simonyan, Erich Elsen, Oriol Vinyals, Jack Rae, and Laurent Sifre. An empirical analysis of compute-optimal large language model training. In *Advances in Neural Information Processing Systems*, volume 35, pp. 30016–30030, 2022. doi: 10.52202/068431-2176. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/c1e2faff6f588870935f114ebe04a3e5-Abstract.html.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, volume 31, pp. 8571–8580, 2018. URL https://arxiv.org/abs/1806.07572.

Tianze Jiang, Blake Bordelon, Cengiz Pehlevan, and Boris Hanin. Hyperparameter transfer with mixture-of-expert layers, 2026. URL https://arxiv.org/abs/2601.20205v3. ICML 2026.

Keller Jordan, Yuchen Jin, Vlado Boza, Jiacheng You, Franz Cesista, Laker Newhouse, and Jeremy Bernstein. Muon: An optimizer for hidden layers in neural networks, 2024a. URL https://kellerjordan.github.io/posts/muon/.

Keller Jordan et al. modded-nanogpt. GitHub repository, 2024b. URL https://github.com/KellerJordan/modded-nanogpt. 2024-10-10 Muon training record.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL https://arxiv.org/abs/2001.08361.

Andrej Karpathy. char-rnn: Multi-layer recurrent neural networks for character-level language models. GitHub repository, 2015. URL https://github.com/karpathy/char-rnn. Tiny Shakespeare dataset.

Andrej Karpathy. build-nanogpt: Building GPT-2 from scratch. https://github.com/karpathy/build-nanogpt, 2024. Commit 6104ab1b53920f6e2159749676073ff7d815c1fa.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015. URL https://arxiv.org/abs/1412.6980.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. URL https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, volume 25, 2012. URL https://papers.nips.cc/paper_files/paper/2012/hash/c399862d3b9d6b76c8436e924a68c45b-Abstract.html.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. doi: 10.1038/nature14539. URL https://www.nature.com/articles/nature14539.

Jaehoon Lee, Lechao Xiao, Samuel S. Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://arxiv.org/abs/1902.06720.

Jingyuan Liu, Jianlin Su, Xingcheng Yao, Zhejun Jiang, Guokun Lai, Yulun Du, Yidao Qin, Weixin Xu, Enzhe Lu, Junjie Yan, Yanru Chen, Huabin Zheng, Yibo Liu, Shaowei Liu, Bohong Yin, Weiran He, Han Zhu, Yuzhi Wang, Jianzhou Wang, Mengnan Dong, Zheng Zhang, Yongsheng Kang, Hao Zhang, Xinran Xu, Yutao Zhang, Yuxin Wu, Xinyu Zhou, and Zhilin Yang. Muon is scalable for LLM training, 2025. URL https://arxiv.org/abs/2502.16982.

Yizhou Liu, Sara Kangaslahti, Ziming Liu, and Jeff Gore. Inverse depth scaling from most layers being similar, 2026. URL https://arxiv.org/abs/2602.05970. ICML 2026.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Anton Lozhkov, Loubna Ben Allal, Leandro von Werra, and Thomas Wolf. FineWeb-Edu: The finest collection of educational content, 2024. URL https://huggingface.co/datasets/HuggingFaceFW/fineweb-edu.

Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018. doi: 10.1073/pnas.1806579115. URL https://arxiv.org/abs/1804.06561.

Lorenzo Noci, Chuning Li, Mufan Bill Li, Bobby He, Thomas Hofmann, Chris J. Maddison, and Daniel M. Roy. The shaped transformer: Attention models in the infinite depth-and-width limit. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/aa31dc84098add7dd2ffdd20646f2043-Abstract-Conference.html.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. PyTorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://arxiv.org/abs/1912.01703.

Jackson Petty, Sjoerd van Steenkiste, Ishita Dasgupta, Fei Sha, Dan Garrette, and Tal Linzen. The impact of depth on compositional generalization in transformer language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 7239–7252. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.naacl-long.402. URL https://aclanthology.org/2024.naacl-long.402/.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. Technical report, OpenAI, 2019. URL https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf.

Samuel S. Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation. In *International Conference on Learning Representations*, 2017. URL https://arxiv.org/abs/1611.01232.

Justin Sirignano and Konstantinos Spiliopoulos. Mean field analysis of neural networks: A law of large numbers. *SIAM Journal on Applied Mathematics*, 80(2):725–752, 2020. doi: 10.1137/18M1192184. URL https://arxiv.org/abs/1805.01053.

Jianlin Su. Higher-order MuP: Simpler yet more powerful spectral-condition scaling. Scientific Spaces, March 2025. URL https://spaces.ac.cn/archives/10795. Blog post in Chinese; title translated into English.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024. doi: 10.1016/j.neucom.2023.127063. URL https://doi.org/10.1016/j.neucom.2023.127063.

Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton. On the importance of initialization and momentum in deep learning. In *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pp. 1139–1147, 2013. URL https://proceedings.mlr.press/v28/sutskever13.html.

Yi Tay, Mostafa Dehghani, Jinfeng Rao, William Fedus, Samira Abnar, Hyung Won Chung, Sharan Narang, Dani Yogatama, Ashish Vaswani, and Donald Metzler. Scale efficiently: Insights from pre-training and fine-tuning transformers. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2109.10686.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30, 2017. URL https://papers.nips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html.

Greg Yang and Edward J. Hu. Tensor programs IV: Feature learning in infinite-width neural networks. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pp. 11727–11737. PMLR, 2021. URL https://proceedings.mlr.press/v139/yang21c.html.

Greg Yang, Edward J. Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tuning large neural networks via zero-shot hyperparameter transfer. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://papers.nips.cc/paper_files/paper/2021/hash/8df7c2e3c3c3be098ef7b382bd2c37ba-Abstract.html.

Greg Yang, Dingli Yu, Chen Zhu, and Soufiane Hayou. Tensor programs VI: Feature learning in infinite depth neural networks. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/f2ce452a4a7352feb7a9fb885267ff24-Abstract-Conference.html.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 4791–4800. Association for Computational Linguistics, 2019. doi: 10.18653/v1/P19-1472. URL https://aclanthology.org/P19-1472/.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. In *Advances in Neural Information Processing Systems*, volume 32, 2019. URL https://arxiv.org/abs/1910.07467.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Zican Dong, Yupeng Hou, Beichen Zhang, Yingqian Min, Junjie Zhang, Peiyu Liu, Xiaolei Wang, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Yiwen Hu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models. *Frontiers of Computer Science*, 20:2012627, 2026. doi: 10.1007/s11704-026-60308-3. URL https://link.springer.com/article/10.1007/s11704-026-60308-3.

A FINITE-NETWORK AND OPTIMIZER CONVENTIONS

A.1 A TRANSFORMER AS TWO POPULATIONS OF NONLINEAR CHANNELS

A Transformer alternates two kinds of nonlinear channel within the same residual stream: attention heads and MLP units. Consider a causal decoder with L attention–MLP stages, residual width $N_\flat$, $N_{\sharp\mathrm{A}}$ heads per stage, and $N_{\sharp\mathrm{M}}$ MLP units per stage. Each head has fixed even internal dimension d, whereas the number of heads can vary independently of the residual width. There are L updates of each kind, hence $K_\mathrm{A} = LN_{\sharp\mathrm{A}}$ and $K_\mathrm{M} = LN_{\sharp\mathrm{M}}$; the intermediate state $\boldsymbol{h}^{(\ell+1/2)\flat}$ records the attention update before the MLP update.

The concrete model uses separate pre-RMSNorm before each branch, rotary positional encoding in attention, GELU, and a final RMSNorm. Its projections have no biases, the input and readout weights are untied, and dropout is omitted. For S tokens, residual states are the columns of $\boldsymbol{h}^{\ell\flat} \in \mathbb{R}^{N_\flat \times S}$. Normalization acts on each column:

$$\mathcal{N}(\boldsymbol{h}_\mathfrak{s}) = \frac{\boldsymbol{h}_\mathfrak{s}}{\sqrt{\|\boldsymbol{h}_\mathfrak{s}\|^2/N_\flat + \epsilon_\mathrm{norm}}}, \qquad \epsilon_\mathrm{norm} > 0. \tag{9}$$

The input columns $\boldsymbol{x}_\mathfrak{s} \in \mathbb{R}^{D_\mathrm{i}}$ encode tokens; $\boldsymbol{x}_\mathfrak{s} = \sqrt{D_\mathrm{i}}\,\boldsymbol{e}_{\mathrm{token}(\mathfrak{s})}$ makes the input map below an embedding lookup. Position information is supplied by the rotary maps.

The full computation retains the code/mathematical-weight distinction of Eq. 1: each code matrix absorbs its displayed coefficient, and the mathematical matrices have independent standard Gaussian entries at initialization. The first and last lines run once; the intervening lines repeat for $\ell = 0, \dots, L-1$, using that stage's code weights. The attention operations run over all heads $a = 1, \dots, N_{\sharp\mathrm{A}}$:

`h = Win @ x`	$\boldsymbol{h}^{0\flat} = \frac{1}{\sqrt{D_\mathrm{i}}}\boldsymbol{W}^0\boldsymbol{x}$
`q[a],k[a],v[a] = split(Wqkv[a] @ rms(h))`	$\begin{pmatrix}\boldsymbol{q}_a\\ \boldsymbol{k}_a\\ \boldsymbol{v}_a\end{pmatrix} = \frac{1}{\sqrt{N_\flat}}\begin{pmatrix}\boldsymbol{W}_a^{\ell Q}\\ \boldsymbol{W}_a^{\ell K}\\ \boldsymbol{W}_a^{\ell V}\end{pmatrix}\mathcal{N}(\boldsymbol{h}^{\ell\flat})$
`o[a] = attend(q[a],k[a],v[a])`	$\boldsymbol{o}_a = \boldsymbol{v}_a P_a^\mathsf{T}$
`hhalf = h + sum(WO[a] @ o[a])`	$\boldsymbol{h}^{(\ell+1/2)\flat} = \boldsymbol{h}^{\ell\flat} + \frac{\beta_{0,\mathrm{A}}\sqrt{N_\flat}}{K_\mathrm{A}\sqrt{d}}\sum_{a=1}^{N_{\sharp\mathrm{A}}}\boldsymbol{W}_a^{\ell O}\boldsymbol{o}_a$
`z = Wup @ rms(hhalf)`	$\boldsymbol{h}^{\ell\sharp\mathrm{M}} = \frac{1}{\sqrt{N_\flat}}\boldsymbol{W}_\mathrm{M}^{\ell\uparrow}\mathcal{N}(\boldsymbol{h}^{(\ell+1/2)\flat})$
`h = hhalf + Wdown @ gelu(z)`	$\boldsymbol{h}^{(\ell+1)\flat} = \boldsymbol{h}^{(\ell+1/2)\flat} + \frac{\beta_{0,\mathrm{M}}\sqrt{N_\flat}}{K_\mathrm{M}}\boldsymbol{W}_\mathrm{M}^{\ell\downarrow}\,\mathrm{GELU}(\boldsymbol{h}^{\ell\sharp\mathrm{M}})$
`f = Wout @ rms(h)`	$\boldsymbol{f} = \frac{1}{\gamma_0 N_\flat}\boldsymbol{W}^L\mathcal{N}(\boldsymbol{h}^{L\flat}).$

(10)

Here $\boldsymbol{q}_a, \boldsymbol{k}_a, \boldsymbol{v}_a \in \mathbb{R}^{d\times S}$, $\boldsymbol{W}_a^{\ell Q}, \boldsymbol{W}_a^{\ell K}, \boldsymbol{W}_a^{\ell V} \in \mathbb{R}^{d\times N_\flat}$, and $\boldsymbol{W}_a^{\ell O} \in \mathbb{R}^{N_\flat \times d}$. The MLP up/down matrices have dimensions $N_{\sharp\mathrm{M}} \times N_\flat$ and $N_\flat \times N_{\sharp\mathrm{M}}$, respectively; $\boldsymbol{f} \in \mathbb{R}^{D_\mathrm{o}\times S}$ contains the output logits. The branch constants $\beta_{0,\mathrm{A}}, \beta_{0,\mathrm{M}}$ play the role of β_0 separately for attention and the MLP.

The attention operation mixes tokens within each head. Let $R_\mathfrak{s}$ be the block-diagonal rotary matrix at token position $\mathfrak{s}$, with two-dimensional rotations by $(\mathfrak{s}-1)\omega_j$ for fixed frequencies ω_j, $j = 0, \dots, d/2-1$. In the line `attend`, the causal attention weights are

$$[P_a]_{\mathfrak{s}\mathfrak{r}} = \begin{cases} \dfrac{\exp((R_\mathfrak{s}\boldsymbol{q}_{a,\mathfrak{s}})^\mathsf{T}(R_\mathfrak{r}\boldsymbol{k}_{a,\mathfrak{r}})/\sqrt{d})}{\sum_{\mathfrak{r}'=1}^{\mathfrak{s}}\exp((R_\mathfrak{s}\boldsymbol{q}_{a,\mathfrak{s}})^\mathsf{T}(R_{\mathfrak{r}'}\boldsymbol{k}_{a,\mathfrak{r}'})/\sqrt{d})}, & \mathfrak{r} \le \mathfrak{s}, \\ 0, & \mathfrak{r} > \mathfrak{s}. \end{cases} \tag{11}$$

The $1/\sqrt{d}$ in these logits and the $1/\sqrt{d}$ in the residual output coefficient are separate normalizations. The latter keeps the output scale per head explicit, as in our Transformer experiments.

Packing the Q/K/V projections makes attention another up–nonlinearity–down computation. For one head, define $\boldsymbol{W}_{\mathrm{A},a}^{\ell\uparrow} = [(\boldsymbol{W}_a^{\ell Q})^\mathsf{T}, (\boldsymbol{W}_a^{\ell K})^\mathsf{T}, (\boldsymbol{W}_a^{\ell V})^\mathsf{T}]^\mathsf{T}$ and $\boldsymbol{W}_{\mathrm{A},a}^{\ell\downarrow} = \boldsymbol{W}_a^{\ell O}$. For one MLP unit,

take the corresponding row of $\boldsymbol{W}_{\mathrm{M}}^{\ell\uparrow}$ and column of $\boldsymbol{W}_{\mathrm{M}}^{\ell\downarrow}$. The two channel maps are

$$\begin{aligned} \phi_{\mathrm{A}} : (\boldsymbol{q}, \boldsymbol{k}, \boldsymbol{v}) \longmapsto \frac{1}{\sqrt{d}} \boldsymbol{v} P(\boldsymbol{q}, \boldsymbol{k})^{\mathsf{T}}, \quad \phi_{\mathrm{A}} : \mathbb{R}^{3d \times S} \to \mathbb{R}^{d \times S}, \\ \phi_{\mathrm{M}} : \boldsymbol{z} \longmapsto \mathrm{GELU}(\boldsymbol{z}), \qquad \phi_{\mathrm{M}} : \mathbb{R}^{1 \times S} \to \mathbb{R}^{1 \times S}, \end{aligned} \tag{12}$$

where P includes the rotary maps and causal mask above. Applying ϕ_{M} is tokenwise; applying ϕ_{A} uses the whole head's token array.

Either residual update now has the same form:

$$\begin{aligned} \boldsymbol{h}_a^{\sharp} &= \frac{1}{\sqrt{N_\flat}} \boldsymbol{W}_a^{\uparrow} \mathcal{N}(\boldsymbol{h}^{\flat}), \\ \boldsymbol{h}^{+\flat} &= \boldsymbol{h}^{\flat} + \frac{\beta_0 \sqrt{N_\flat}}{L N_\sharp} \sum_{a=1}^{N_\sharp} \boldsymbol{W}_a^{\downarrow} \phi(\boldsymbol{h}_a^{\sharp}). \end{aligned} \tag{13}$$

For the first update use $(\phi, N_\sharp, \beta_0) = (\phi_{\mathrm{A}}, N_{\sharp\mathrm{A}}, \beta_{0,\mathrm{A}})$; for the second use $(\phi_{\mathrm{M}}, N_{\sharp\mathrm{M}}, \beta_{0,\mathrm{M}})$ and the residual state already updated by attention. The attention output factor $1/\sqrt{d}$ has moved into ϕ_{A}; the mathematical weights and the computed function are unchanged. Each population occurs L times, although a stage contains two residual updates. With $\mathcal{N} = \mathrm{Id}$, this is Eq. 1, written as a sum over channels.

Tokens and coordinates within a head specify what a channel does; they are not the channel population $N_\sharp$. Pre-normalization is shared residual preprocessing and remains explicit in Eq. 13. Neither the MLP width $N_{\sharp\mathrm{M}}$ nor the concatenated attention width $dN_{\sharp\mathrm{A}}$ is required to equal $N_\flat$. The structural variables are therefore still the residual width, the two populations of $\sharp$ channels, and their allocation over depth. Biases and affine normalization parameters can be added at their usual locations. Their parameter and symmetry conventions must then also be specified; the displayed example uses non-affine RMSNorm throughout.

A.2 Adam and Muon

Adam. Adam (Kingma & Ba, 2015) rescales each gradient coordinate using its own recent history. It maintains exponential moving averages m and v of the gradient and its entrywise square, and updates

$$W^{+} = W - \eta \frac{\widehat{m}}{\sqrt{\widehat{v}} + \epsilon}, \tag{14}$$

where the hats denote bias correction for zero-initialized moments, and the square root and division are entrywise.

Muon. Muon (Jordan et al., 2024a) transforms the singular values of a momentum update matrix while retaining its singular vectors. For $D = U \operatorname{diag}(s_j) V^{\mathsf{T}}$, the ideal operation maps every nonzero singular value to one and leaves zero at zero. In practice, the input matrix is first normalized, then a few Newton–Schulz iterations apply an odd quintic polynomial $p(s) = as + bs^3 + cs^5$ to its singular values (Jordan et al., 2024a; Liu et al., 2025).

Following the spectral-clipping viewpoint of Su (2025), we model the finite-iteration spectral action by normalized matrix clipping:

$$\begin{aligned} W^{+} = W - \eta_M F_\lambda(D_W), \qquad D_W = \nabla_W \mathcal{L}, \\ F_\lambda(D) = \mathrm{mclip}(D/\lambda) = U \operatorname{diag}\big(\min(s_j/\lambda, 1)\big) V^{\mathsf{T}}, \qquad \lambda > 0. \end{aligned} \tag{15}$$

With momentum, the same map acts on the accumulated update matrix. This model retains a linear response near zero and saturation at larger singular values, with $F_\lambda(0) = 0$.

A.3 Scalings for feature learning

Feature-learning scaling keeps individual hidden activations random and $\Theta(1)$ at initialization, while the macroscopic output $\boldsymbol{f}$ may tend to zero as the network grows. Learning rates are chosen to retain

$\Theta(1)$ changes in intermediate features during training. The SGD and Adam choices in Table 2 follow these standard principles (Bordelon et al., 2024b; Chizat, 2025). The table gives the scales in both code and standard-Gaussian coordinates.

For Adam, write a parameter group's gradient as $D_t^j = d_j R_t^j$, where d_j is its entry scale, and let M_t^j, V_t^j be the moments of the normalized gradients R_t^j. Then

$$\widehat{m}_t^j = d_j \widehat{M}_t^j, \qquad \widehat{v}_t^j = d_j^2 \widehat{V}_t^j, \qquad \frac{\widehat{m}_t^j}{\sqrt{\widehat{v}_t^j} + \epsilon_j} = \frac{\widehat{M}_t^j}{\sqrt{\widehat{V}_t^j} + \epsilon_j / d_j}. \tag{16}$$

Thus $\epsilon_j = d_j \epsilon_0$ removes d_j from the normalized update. For example, the code up-projection has $d_\uparrow \asymp 1/K$, so $\epsilon_\uparrow \asymp \epsilon_0/K$.

For Muon, consider the code up-projection for one example and a scalar output. Suppressing fixed multipliers, its rank-one gradient and sole nonzero singular value have scales

$$D = \nabla_{\texttt{W_up}} \mathcal{L} \propto \frac{1}{K} \boldsymbol{g}_1^{\ell\sharp} (\boldsymbol{h}^{\ell\flat})^\mathsf{T}, \qquad \sigma(D) \asymp \frac{\sqrt{N_\flat N_\sharp}}{K}. \tag{17}$$

Choosing $\lambda \asymp \sqrt{N_\flat N_\sharp}/K$ compensates for this gradient dilution, including both matrix dimensions. The spectral map preserves the gradient's right singular vector, which is aligned with $\boldsymbol{h}^{\ell\flat}$, so the direct feature change is

$$\frac{\|(\Delta \texttt{W_up}) \boldsymbol{h}^{\ell\flat}\|}{\sqrt{N_\sharp}} \asymp \eta_M \sqrt{\frac{N_\flat}{N_\sharp}}. \tag{18}$$

Hence $\eta_M \asymp \sqrt{N_\sharp / N_\flat}$ gives a $\Theta(1)$ feature update, as in the table.

Optimizer	Quantity	W_in	W_up	W_down	W_out
Initialization std.		1	$1/\sqrt{N_\flat}$	$\sqrt{N_\flat}/K$	$1/N_\flat$
SGD	LR (code)	$N_\flat$	$K/N_\flat$	$N_\flat/K$	$1/N_\flat$
	LR (standard Gaussian)	$N_\flat$	K	K	$N_\flat$
Muon	LR (code)	$\sqrt{N_\flat}$	$\sqrt{N_\sharp/N_\flat}$	$\sqrt{N_\flat N_\sharp}/K$	$1/\sqrt{N_\flat}$
	Clip λ (code)	$1/\sqrt{N_\flat}$	$\sqrt{N_\flat N_\sharp}/K$	$\sqrt{N_\sharp/N_\flat}$	$\sqrt{N_\flat}$
	LR (standard Gaussian)	$\sqrt{N_\flat}$	$\sqrt{N_\sharp}$	$\sqrt{N_\sharp}$	$\sqrt{N_\flat}$
	Clip λ (standard Gaussian)	$1/\sqrt{N_\flat}$	$\sqrt{N_\sharp}/K$	$\sqrt{N_\sharp}/K$	$1/\sqrt{N_\flat}$
Adam	LR (code)	1	$1/N_\flat$	$1/K$	$1/N_\flat$
	ϵ (code)	$1/N_\flat$	$1/K$	$1/N_\flat$	1
	LR (standard Gaussian)	1	$1/\sqrt{N_\flat}$	$1/\sqrt{N_\flat}$	1
	ϵ (standard Gaussian)	$1/N_\flat$	$1/(K\sqrt{N_\flat})$	$1/(K\sqrt{N_\flat})$	$1/N_\flat$

Table 2: Complete size scalings in code and standard-Gaussian coordinates. Fixed task dimensions and hyperparameter multipliers, including ϵ_0, are suppressed.

B Optimizer-dependent dynamics and population descriptions

The SGD equations follow by accumulating the parameter updates before taking a population limit. Throughout this appendix, $K = N_\sharp L$. All four matrices in (1) are trained, and t counts training steps while ℓ counts residual blocks. We first give the finite identities and their population response equations, then construct the corresponding spectral and moment histories for Muon and Adam. Their implications for residual Gaussianity follow in the same section.

At step s, let $\mathcal{B}_s$ be a batch of B_s examples and set

$$\Delta_p(\boldsymbol{x}', y', s) = -\frac{\partial \mathcal{L}_{\mathrm{ex}}(\boldsymbol{f}(\boldsymbol{x}', s), y')}{\partial f_p}, \qquad p = 1, \dots, D_{\mathrm{o}}. \tag{19}$$

The loss is averaged over the batch, and all parameter groups are updated simultaneously. Any output-component averaging belongs to $\mathcal{L}_{\mathrm{ex}}$. For fixed positive constants $\eta_{\mathrm{in}}, \eta_{\uparrow}, \eta_{\downarrow}, \eta_{\mathrm{out}}$, the standard-coordinate SGD learning rates are

$$\begin{aligned} \eta_{\mathrm{in}}^{\mathrm{W}} &= \eta_{\mathrm{in}}\gamma_0^2 N_\flat, \quad & \eta_{\uparrow}^{\mathrm{W}} &= \eta_{\uparrow}\gamma_0^2 K, \\ \eta_{\downarrow}^{\mathrm{W}} &= \eta_{\downarrow}\gamma_0^2 K, \quad & \eta_{\mathrm{out}}^{\mathrm{W}} &= \eta_{\mathrm{out}}\gamma_0^2 N_\flat. \end{aligned} \tag{20}$$

The four constants may differ; Table 1 suppresses them and the fixed factor γ_0^2.

B.1 Backward recursion and parameter gradients

The normalized output Jacobians defined in (2) obey the exact recursion

$$\begin{aligned} \boldsymbol{g}_q^{L\flat} &= (\boldsymbol{W}^L)^{\mathsf{T}} \boldsymbol{e}_q, \\ \boldsymbol{g}_q^{\ell\sharp} &= \phi'(\boldsymbol{h}^{\ell\sharp}) \odot \frac{(\boldsymbol{W}^{\ell\downarrow})^{\mathsf{T}} \boldsymbol{g}_q^{(\ell+1)\flat}}{\sqrt{N_\flat}}, \\ \boldsymbol{g}_q^{\ell\flat} &= \boldsymbol{g}_q^{(\ell+1)\flat} + \frac{\beta_0\sqrt{N_\flat}}{K} (\boldsymbol{W}^{\ell\uparrow})^{\mathsf{T}} \boldsymbol{g}_q^{\ell\sharp}. \end{aligned} \tag{21}$$

The input and training step are the same throughout each line and have been suppressed. In particular, there is no factor $\phi'(\boldsymbol{h}^{\ell\flat})$ in the residual recursion. The four parameter derivatives are

$$\begin{aligned} \nabla_{\boldsymbol{W}^0} f_q &= \frac{\boldsymbol{g}_q^{0\flat} \boldsymbol{x}^{\mathsf{T}}}{\gamma_0 N_\flat \sqrt{D_{\mathrm{i}}}}, & \nabla_{\boldsymbol{W}^L} f_q &= \frac{\boldsymbol{e}_q (\boldsymbol{h}^{L\flat})^{\mathsf{T}}}{\gamma_0 N_\flat}, \\ \nabla_{\boldsymbol{W}^{\ell\uparrow}} f_q &= \frac{\beta_0}{\gamma_0 K \sqrt{N_\flat}} \boldsymbol{g}_q^{\ell\sharp} (\boldsymbol{h}^{\ell\flat})^{\mathsf{T}}, & \nabla_{\boldsymbol{W}^{\ell\downarrow}} f_q &= \frac{\beta_0}{\gamma_0 K \sqrt{N_\flat}} \boldsymbol{g}_q^{(\ell+1)\flat} \phi(\boldsymbol{h}^{\ell\sharp})^{\mathsf{T}}. \end{aligned} \tag{22}$$

Multiplication by Δ_q, summation over q, and batch averaging produce the descent direction. Thus the loss feedback and the output Jacobian remain separate.

B.2 Exact discrete learning histories

For a current query $z = (\boldsymbol{x}, t)$ and a past example $z' = (\boldsymbol{x}', s)$, define the empirical kernels

$$\begin{aligned} \Phi^{\ell\flat}(z, z') &= \frac{(\boldsymbol{h}^{\ell\flat}(z))^{\mathsf{T}} \boldsymbol{h}^{\ell\flat}(z')}{N_\flat}, \quad & \Phi^{\ell\sharp}(z, z') &= \frac{\phi(\boldsymbol{h}^{\ell\sharp}(z))^{\mathsf{T}} \phi(\boldsymbol{h}^{\ell\sharp}(z'))}{N_\sharp}, \\ G_{qp}^{\ell\flat}(z, z') &= \frac{(\boldsymbol{g}_q^{\ell\flat}(z))^{\mathsf{T}} \boldsymbol{g}_p^{\ell\flat}(z')}{N_\flat}, \quad & G_{qp}^{\ell\sharp}(z, z') &= \frac{(\boldsymbol{g}_q^{\ell\sharp}(z))^{\mathsf{T}} \boldsymbol{g}_p^{\ell\sharp}(z')}{N_\sharp}. \end{aligned} \tag{23}$$

The loss-weighted training history will be written as

$$\mathcal{S}_t[F_p(z, z')] = \sum_{s=0}^{t-1} \frac{1}{B_s} \sum_{(\boldsymbol{x}', y') \in \mathcal{B}_s} \sum_{p=1}^{D_{\mathrm{o}}} \Delta_p(\boldsymbol{x}', y', s)\, F_p(z, z'). \tag{24}$$

Accumulating the up-projection updates exactly gives

$$\boldsymbol{W}^{\ell\uparrow}(t) = \boldsymbol{W}^{\ell\uparrow}(0) + \frac{\eta_{\uparrow}\gamma_0\beta_0}{\sqrt{N_\flat}} \mathcal{S}_t[\boldsymbol{g}_p^{\ell\sharp}(z')(\boldsymbol{h}^{\ell\flat}(z'))^{\mathsf{T}}]. \tag{25}$$

The down-projection has the corresponding outer product of $\boldsymbol{g}_p^{(\ell+1)\flat}$ and $\phi(\boldsymbol{h}^{\ell\sharp})$. Substitution into the network equations gives the forward histories

$$\begin{aligned} \boldsymbol{h}^{\ell\sharp}(z) &= \boldsymbol{\chi}^{\ell\sharp}(z) + \eta_{\uparrow}\gamma_0\beta_0\, \mathcal{S}_t[\boldsymbol{g}_p^{\ell\sharp}(z')\Phi^{\ell\flat}(z, z')], \\ \boldsymbol{h}^{(\ell+1)\flat}(z) &= \boldsymbol{h}^{\ell\flat}(z) + \boldsymbol{\chi}^{(\ell+1)\flat}(z) + \frac{\eta_{\downarrow}\gamma_0\beta_0^2}{L}\, \mathcal{S}_t[\boldsymbol{g}_p^{(\ell+1)\flat}(z')\Phi^{\ell\sharp}(z, z')]. \end{aligned} \tag{26}$$

Here the parts containing the initial matrices are

$$\begin{aligned}
\boldsymbol{\chi}^{\ell\sharp}(z) &= \frac{\boldsymbol{W}^{\ell\uparrow}(0)\boldsymbol{h}^{\ell\flat}(z)}{\sqrt{N_\flat}},\\
\boldsymbol{\chi}^{(\ell+1)\flat}(z) &= \frac{\beta_0\sqrt{N_\flat}}{K}\boldsymbol{W}^{\ell\downarrow}(0)\phi(\boldsymbol{h}^{\ell\sharp}(z)),\\
\boldsymbol{\xi}_q^{\ell\sharp}(z) &= \frac{(\boldsymbol{W}^{\ell\downarrow}(0))^{\mathsf{T}}\boldsymbol{g}_q^{(\ell+1)\flat}(z)}{\sqrt{N_\flat}},\\
\boldsymbol{\xi}_q^{\ell\flat}(z) &= \frac{\beta_0\sqrt{N_\flat}}{K}(\boldsymbol{W}^{\ell\uparrow}(0))^{\mathsf{T}}\boldsymbol{g}_q^{\ell\sharp}(z).
\end{aligned} \tag{27}$$

The backward histories are

$$\begin{aligned}
\boldsymbol{g}_q^{\ell\sharp}(z) &= \phi'(\boldsymbol{h}^{\ell\sharp}(z))\odot\left\{\boldsymbol{\xi}_q^{\ell\sharp}(z)+\eta_\downarrow\gamma_0\beta_0\,\mathcal{S}_t[\phi(\boldsymbol{h}^{\ell\sharp}(z'))G_{qp}^{(\ell+1)\flat}(z,z')]\right\},\\
\boldsymbol{g}_q^{\ell\flat}(z) &= \boldsymbol{g}_q^{(\ell+1)\flat}(z)+\boldsymbol{\xi}_q^{\ell\flat}(z)+\frac{\eta_\uparrow\gamma_0\beta_0^2}{L}\,\mathcal{S}_t[\boldsymbol{h}^{\ell\flat}(z')G_{qp}^{\ell\sharp}(z,z')].
\end{aligned} \tag{28}$$

Hidden inner products supply $N_\sharp/K = 1/L$ in the residual equations. The hidden equations themselves retain order-one learning-history coefficients. This separates the update of an individual hidden feature from its contribution to a residual increment.

For the endpoints, write $\kappa_{\mathrm{in}}(\boldsymbol{x},\boldsymbol{x}') = \boldsymbol{x}^{\mathsf{T}}\boldsymbol{x}'/D_{\mathrm{i}}$. Their histories are

$$\begin{aligned}
\boldsymbol{h}^{0\flat}(z) &= \frac{\boldsymbol{W}^0(0)\boldsymbol{x}}{\sqrt{D_{\mathrm{i}}}}+\eta_{\mathrm{in}}\gamma_0\,\mathcal{S}_t[\boldsymbol{g}_p^{0\flat}(z')\kappa_{\mathrm{in}}(\boldsymbol{x},\boldsymbol{x}')],\\
\boldsymbol{g}_q^{L\flat}(z) &= (\boldsymbol{W}^L(0))^{\mathsf{T}}\boldsymbol{e}_q+\eta_{\mathrm{out}}\gamma_0\,\mathcal{S}_t[\delta_{pq}\boldsymbol{h}^{L\flat}(z')],\\
f_q(z) &= \frac{\boldsymbol{e}_q^{\mathsf{T}}\boldsymbol{W}^L(0)\boldsymbol{h}^{L\flat}(z)}{\gamma_0 N_\flat}+\eta_{\mathrm{out}}\,\mathcal{S}_t[\delta_{pq}\Phi^{L\flat}(z,z')].
\end{aligned} \tag{29}$$

All fields in (26)–(29) are evaluated in the trained network. The identities therefore include the full finite update, including interactions between parameter groups updated at the same step.

B.3 Dynamical mean-field equations from Gaussian weight integration

Weight reuse makes the initial projections in (27) correlated with the fields they multiply. Integrating those Gaussian weights and taking the population saddle separates the projections into Gaussian sources and responses to weight reuse. The derivation below retains the exact discrete training histories. Take a finite input set containing the training examples and probes, and finitely many training steps; denote their pairs $z = (\boldsymbol{x}, t)$ by $\mathcal{Z}$.

The representative fields retain the main-text notation $(\boldsymbol{\hbar}^{\ell\flat}, \boldsymbol{\hbar}^{\ell\sharp}, \boldsymbol{\mathscr{g}}_q^{\ell\flat}, \boldsymbol{\mathscr{g}}_q^{\ell\sharp})$. The second-moment kernels are defined self-consistently by

$$\begin{aligned}
\Phi^{\ell\flat}(z,z') &= \mathbb{E}[\boldsymbol{\hbar}^{\ell\flat}(z)\boldsymbol{\hbar}^{\ell\flat}(z')], \quad \Phi^{\ell\sharp}(z,z') = \mathbb{E}[\phi(\boldsymbol{\hbar}^{\ell\sharp}(z))\phi(\boldsymbol{\hbar}^{\ell\sharp}(z'))],\\
G_{qp}^{\ell\flat}(z,z') &= \mathbb{E}[\boldsymbol{\mathscr{g}}_q^{\ell\flat}(z)\boldsymbol{\mathscr{g}}_p^{\ell\flat}(z')], \quad G_{qp}^{\ell\sharp}(z,z') = \mathbb{E}[\boldsymbol{\mathscr{g}}_q^{\ell\sharp}(z)\boldsymbol{\mathscr{g}}_p^{\ell\sharp}(z')].
\end{aligned} \tag{30}$$

The hidden kernels are second moments and may include nonzero means.

B.3.1 Gaussian weight integration and the population saddle

The response terms can be derived by integrating the initial Gaussian matrices, while retaining the exact discrete training histories. Fix L and a finite set $\mathcal{Z}$ containing every training and probe input at steps $0,\dots,T$. All sums over $z, z' \in \mathcal{Z}$ below are ordinary finite sums; batch weights and loss derivatives occur only in $\mathcal{S}_t$ from (24). For one block, use the same ℓ throughout.

Thus $\boldsymbol{W}^{\ell\uparrow}(0)$ is $N_\sharp \times N_\flat$, $\boldsymbol{W}^{\ell\downarrow}(0)$ is $N_\flat \times N_\sharp$, and their entries are independent standard Gaussians. The four projections in (27), with the depth index suppressed, are

$$\begin{aligned}
\boldsymbol{\chi}^{\sharp} &= \frac{\boldsymbol{W}^{\ell\uparrow}(0)\boldsymbol{h}^{\ell\flat}}{\sqrt{N_\flat}}, & \boldsymbol{\xi}_q^{\flat} &= \frac{\beta_0\sqrt{N_\flat}}{K}(\boldsymbol{W}^{\ell\uparrow}(0))^{\mathsf{T}}\boldsymbol{g}_q^{\ell\sharp},\\
\boldsymbol{\chi}^{\flat} &= \frac{\beta_0\sqrt{N_\flat}}{K}\boldsymbol{W}^{\ell\downarrow}(0)\phi(\boldsymbol{h}^{\ell\sharp}), & \boldsymbol{\xi}_q^{\sharp} &= \frac{(\boldsymbol{W}^{\ell\downarrow}(0))^{\mathsf{T}}\boldsymbol{g}_q^{(\ell+1)\flat}}{\sqrt{N_\flat}}.
\end{aligned} \tag{31}$$

In particular, the residual projections already contain β_0/L. There is no activation applied to $\boldsymbol{h}^{\ell\flat}$.

The training constraints are retained. Let Γ collect the forward, backward, projection, and output fields. The generating functional integrates them against the dynamical measure

$$\mathcal{D}_{\rm dyn} = \mathcal{D}_{\rm bdry} \prod_{\ell,z} \left[\delta[\boldsymbol{E}^{\sharp}_{h,\ell}(z)] \delta[\boldsymbol{E}^{\flat}_{h,\ell}(z)] \prod_{q} \delta[\boldsymbol{E}^{\sharp}_{g,\ell q}(z)] \delta[\boldsymbol{E}^{\flat}_{g,\ell q}(z)] \right]. \tag{32}$$

Here the four $\boldsymbol{E}$ are the left sides minus the right sides of (26) and (28); $\mathcal{D}_{\rm bdry}$ enforces (29), including the output used in the loss feedback. These constraints encode simultaneous SGD, with every update depending on steps $s < t$. Ordering the computation by training step, then forward and backward passes, gives a triangular constraint Jacobian with unit diagonal. Consequently the generating functional $\mathscr{Z}[J]$ satisfies $\mathscr{Z}[0] = 1$, also after arbitrary additive perturbations of the four projection equations. The endpoint Gaussian averages can be carried out separately; the calculation below treats the bulk matrices.

Insert Fourier representations of the four projection constraints. Their field-only exponent is

$$iS_{\rm F} = i \sum_{\ell,z} \left[\widehat{\boldsymbol{\chi}}^{\sharp} \cdot \boldsymbol{\chi}^{\sharp} + \widehat{\boldsymbol{\chi}}^{\flat} \cdot \boldsymbol{\chi}^{\flat} + \sum_{q} \left(\widehat{\boldsymbol{\xi}}^{\sharp}_{q} \cdot \boldsymbol{\xi}^{\sharp}_{q} + \widehat{\boldsymbol{\xi}}^{\flat}_{q} \cdot \boldsymbol{\xi}^{\flat}_{q} \right) \right]. \tag{33}$$

The remaining initial-weight dependence is $\exp[-i \sum_{ai} \Lambda^{\uparrow}_{ai} W^{\ell\uparrow}_{ai}(0) - i \sum_{ia} \Lambda^{\downarrow}_{ia} W^{\ell\downarrow}_{ia}(0)]$, where, for each layer,

$$\begin{aligned} \Lambda^{\uparrow}_{ai} &= \sum_{z} \left[\frac{\widehat{\chi}^{\sharp}_{a}(z) h^{\ell\flat}_{i}(z)}{\sqrt{N_{\flat}}} + \frac{\beta_0 \sqrt{N_{\flat}}}{K} \sum_{q} \widehat{\xi}^{\flat}_{qi}(z) g^{\ell\sharp}_{q,a}(z) \right], \\ \Lambda^{\downarrow}_{ia} &= \sum_{z} \left[\frac{\beta_0 \sqrt{N_{\flat}}}{K} \widehat{\chi}^{\flat}_{i}(z) \phi(h^{\ell\sharp}_{a}(z)) + \frac{1}{\sqrt{N_{\flat}}} \sum_{q} \widehat{\xi}^{\sharp}_{qa}(z) g^{(\ell+1)\flat}_{q,i}(z) \right]. \end{aligned} \tag{34}$$

Thus the full generating functional before the bulk Gaussian average is

$$\mathscr{Z}[J] = \mathbb{E}_{\rm bdry} \int D\Gamma \, D\widehat{\chi} \, D\widehat{\xi} \, \mathcal{D}_{\rm dyn} \, e^{J\cdot\Gamma + iS_{\rm F}} \prod_{\ell} \mathbb{E}_{\boldsymbol{W}^{\ell\uparrow}(0), \boldsymbol{W}^{\ell\downarrow}(0)} e^{-i\Lambda^{\uparrow}_{\ell} : \boldsymbol{W}^{\ell\uparrow}(0) - i\Lambda^{\downarrow}_{\ell} : \boldsymbol{W}^{\ell\downarrow}(0)}. \tag{35}$$

The outer expectation averages the initial input and readout matrices.

At this point all fields are integration variables, so the Gaussian average is exact despite their dependence on the trained weights:

$$\mathbb{E}_{\boldsymbol{W}^{\ell\uparrow}(0), \boldsymbol{W}^{\ell\downarrow}(0)} e^{-i\Lambda^{\uparrow} : \boldsymbol{W}^{\ell\uparrow}(0) - i\Lambda^{\downarrow} : \boldsymbol{W}^{\ell\downarrow}(0)} = \exp\left[-\frac{1}{2} \sum_{ai} (\Lambda^{\uparrow}_{ai})^2 - \frac{1}{2} \sum_{ia} (\Lambda^{\downarrow}_{ia})^2 \right]. \tag{36}$$

Macroscopic kernels separate the coordinates. Insert the four empirical second moments in (23) and two mixed kernels

$$\begin{aligned} A^{\flat}_{q}(z, z') &= -\frac{i}{N_{\flat}} \boldsymbol{h}^{\ell\flat}(z) \cdot \widehat{\boldsymbol{\xi}}^{\flat}_{q}(z'), \\ A^{\sharp}_{q}(z, z') &= -\frac{i}{N_{\sharp}} \phi(\boldsymbol{h}^{\ell\sharp}(z)) \cdot \widehat{\boldsymbol{\xi}}^{\sharp}_{q}(z'). \end{aligned} \tag{37}$$

Their response interpretation will follow from the Fourier integral. For a generic second moment $C(z, z') = \boldsymbol{v}(z) \cdot \boldsymbol{v}(z')/d$, its constraint contributes

$$1 \;\propto\; \int DC \, D\widehat{C} \, \exp\left\{ \sum_{z,z'} \widehat{C}(z, z') [dC(z, z') - \boldsymbol{v}(z) \cdot \boldsymbol{v}(z')] \right\}. \tag{38}$$

Apply this identity to $(\Phi^{\flat}, \Phi^{\sharp}, G^{(\ell+1)\flat}, G^{\sharp})$ with respective widths $(N_{\flat}, N_{\sharp}, N_{\flat}, N_{\sharp})$; for G, the pair (q, z) is a single covariance index. Conjugate integration contours and field-independent normalizing constants are suppressed. The two mixed-kernel constraints contribute

$$\begin{aligned} S_A = -\sum_{q,z,z'} \big[& N_{\flat} \widehat{A}^{\flat}_{q} A^{\flat}_{q} + i \widehat{A}^{\flat}_{q} \, \boldsymbol{h}^{\ell\flat}(z) \cdot \widehat{\boldsymbol{\xi}}^{\flat}_{q}(z') \\ & + N_{\sharp} \widehat{A}^{\sharp}_{q} A^{\sharp}_{q} + i \widehat{A}^{\sharp}_{q} \, \phi(\boldsymbol{h}^{\ell\sharp}(z)) \cdot \widehat{\boldsymbol{\xi}}^{\sharp}_{q}(z') \big]. \end{aligned} \tag{39}$$

Undisplayed arguments on the conjugate and mixed kernels are (z, z'). In the dynamical measure, empirical kernels are simultaneously replaced by these constrained macroscopic variables.

The readout is another macroscopic variable to be closed. Since $\boldsymbol{g}_q^{L\flat}(z) = (\boldsymbol{W}^L(t))^\mathsf{T}\boldsymbol{e}_q$, it is exactly $f_q(z) = (\gamma_0 N_\flat)^{-1}\boldsymbol{h}^{L\flat}(z)\cdot\boldsymbol{g}_q^{L\flat}(z)$. Represent its constraint in $\mathcal{D}_{\mathrm{bdry}}$ using a conjugate $\widehat{f}_q(z)$ with exponent

$$S_f = \sum_{q,z}\widehat{f}_q(z)\left[\gamma_0 N_\flat f_q(z) - \sum_{i=1}^{N_\flat} h_i^{L\flat}(z) g_{q,i}^{L\flat}(z)\right]. \tag{40}$$

This replaces the readout delta; the input and terminal-Jacobian constraints remain in $\mathcal{D}_{\mathrm{bdry}}$. At fixed f, the loss feedback Δ_p is shared by all coordinates, and the second term of (40) is a sum of single-coordinate terms. Averaging the independent initial endpoint matrices also preserves this factorization.

Expanding the two squares in (36) now gives the following averaged action for one block:

$$\begin{aligned} S_{\mathrm{av}} = \sum_{z,z'}\Big[&-\frac{1}{2}\Phi^\flat(z,z')\,\widehat{\boldsymbol{\chi}}^\sharp(z)\cdot\widehat{\boldsymbol{\chi}}^\sharp(z') \\ &-\frac{\beta_0^2 N_\flat}{2KL}\sum_{q,p} G_{qp}^\sharp(z,z')\,\widehat{\boldsymbol{\xi}}_q^\flat(z)\cdot\widehat{\boldsymbol{\xi}}_p^\flat(z') \\ &-i\frac{\beta_0 N_\flat}{K}\sum_q A_q^\flat(z,z')\,\widehat{\boldsymbol{\chi}}^\sharp(z)\cdot\boldsymbol{g}_q^{\ell\sharp}(z') \\ &-\frac{\beta_0^2 N_\flat}{2KL}\Phi^\sharp(z,z')\,\widehat{\boldsymbol{\chi}}^\flat(z)\cdot\widehat{\boldsymbol{\chi}}^\flat(z') \\ &-\frac{1}{2}\sum_{q,p} G_{qp}^{(\ell+1)\flat}(z,z')\,\widehat{\boldsymbol{\xi}}_q^\sharp(z)\cdot\boldsymbol{\xi}_p^\sharp(z') \\ &-i\frac{\beta_0}{L}\sum_q A_q^\sharp(z,z')\,\widehat{\boldsymbol{\chi}}^\flat(z)\cdot\boldsymbol{g}_q^{(\ell+1)\flat}(z')\Big]. \end{aligned} \tag{41}$$

For example, the up-matrix cross term is $-(\beta_0/K)\sum_{q,z,z'}(\widehat{\boldsymbol{\chi}}^\sharp(z)\cdot\boldsymbol{g}_q^{\ell\sharp}(z'))(\boldsymbol{h}^{\ell\flat}(z)\cdot\widehat{\boldsymbol{\xi}}_q^\flat(z'))$. Substitution of (37) gives the factor $-i\beta_0 N_\flat/K$ in the third line. The down-matrix cross term similarly gives $-i\frac{\beta_0}{L}$ in the last line. The action, together with the constrained histories, factors over $N_\flat$ residual coordinates and $N_\sharp$ hidden coordinates at each block. Each residual coordinate retains its entire depth history; it is not factored separately at different depths.

The physical saddle is normalized and causal. Varying the conjugate kernels gives the population second moments (30) and the single-coordinate versions of (37). Varying the mixed kernels gives

$$\begin{aligned} \widehat{A}_q^\flat(z,z') &= -i\frac{\beta_0}{L}\,\mathbb{E}[\widehat{\chi}^\sharp(z)\boldsymbol{\mathcal{g}}_q^{\ell\sharp}(z')], \\ \widehat{A}_q^\sharp(z,z') &= -i\frac{\beta_0 N_\flat}{K}\,\mathbb{E}[\widehat{\chi}^\flat(z)\boldsymbol{\mathcal{g}}_q^{(\ell+1)\flat}(z')]. \end{aligned} \tag{42}$$

Expectations here refer to one representative coordinate of the corresponding population. The remaining conjugate kernels vanish at zero physical source:

$$\widehat{\Phi}^\flat = \widehat{\Phi}^\sharp = \widehat{G}^\flat = \widehat{G}^\sharp = 0. \tag{43}$$

Two ingredients establish this statement. First, differentiating $\mathscr{Z}[0] = 1$ twice with respect to additive projection perturbations makes every pure Fourier-field second moment vanish. Without the

history terms, the four kernel saddle equations would read

$$\begin{aligned}
\widehat{\Phi}^{\flat}(z,z') &= \frac{1}{2N_{\flat}}\,\mathbb{E}[\widehat{\boldsymbol{\chi}}^{\sharp}(z)\cdot\widehat{\boldsymbol{\chi}}^{\sharp}(z')],\\
\widehat{\Phi}^{\sharp}(z,z') &= \frac{\beta_0^2 N_{\flat}}{2KLN_{\sharp}}\,\mathbb{E}[\widehat{\boldsymbol{\chi}}^{\flat}(z)\cdot\widehat{\boldsymbol{\chi}}^{\flat}(z')],\\
\widehat{G}^{(\ell+1)\flat}_{qp}(z,z') &= \frac{1}{2N_{\flat}}\,\mathbb{E}[\widehat{\boldsymbol{\xi}}^{\sharp}_{q}(z)\cdot\widehat{\boldsymbol{\xi}}^{\sharp}_{p}(z')],\\
\widehat{G}^{\sharp}_{qp}(z,z') &= \frac{\beta_0^2 N_{\flat}}{2KLN_{\sharp}}\,\mathbb{E}[\widehat{\boldsymbol{\xi}}^{\flat}_{q}(z)\cdot\widehat{\boldsymbol{\xi}}^{\flat}_{p}(z')].
\end{aligned} \tag{44}$$

The vector expectations in this display include the coordinate sums.

Second, the explicit histories in $\mathcal{D}_{\mathrm{dyn}}$ also depend on Φ and G. Their variations vanish at the causal saddle, rather than providing extra terms in (44). To see this, temporarily exponentiate the dynamical constraints with Fourier fields $\widehat{E}^{\sharp}_{h}, \widehat{E}^{\flat}_{h}, \widehat{E}^{\sharp}_{g,q}, \widehat{E}^{\flat}_{g,q}$. Ignoring their displayed learning-rate coefficients and the common past loss feedback, the four variations give

$$\begin{aligned}
\delta\Phi^{\flat}: &\quad \mathbb{E}[-i\widehat{E}^{\sharp}_{h}(z)\boldsymbol{\varkappa}^{\ell\sharp}_{p}(z')] = 0,\\
\delta\Phi^{\sharp}: &\quad \mathbb{E}[-i\widehat{E}^{\flat}_{h}(z)\boldsymbol{\varkappa}^{(\ell+1)\flat}_{p}(z')] = 0,\\
\delta G^{(\ell+1)\flat}_{qp}: &\quad \mathbb{E}[-i\widehat{E}^{\sharp}_{g,q}(z)\phi'(\boldsymbol{\hbar}^{\ell\sharp}(z))\phi(\boldsymbol{\hbar}^{\ell\sharp}(z'))] = 0,\\
\delta G^{\sharp}_{qp}: &\quad \mathbb{E}[-i\widehat{E}^{\flat}_{g,q}(z)\boldsymbol{\hbar}^{\ell\flat}(z')] = 0, \qquad z=(\boldsymbol{x},t),\quad z'=(\boldsymbol{x}',s),\quad s<t.
\end{aligned} \tag{45}$$

Each expectation is the response of its accompanying fields to a current additive perturbation of the indicated equation. All primed fields precede that perturbation. In the third line, $\boldsymbol{\hbar}^{\ell\sharp}(z)$ also precedes the current backward pass, so its activation derivative has zero response to that perturbation. This argument applies to every output pair (q,p) and preserves the strict discrete-step ordering. It justifies (43) for the full training functional, not only for its Gaussian-matrix factor.

The readout closes by the same causal saddle. Variation with respect to $\widehat{f}$ gives

$$f_q(z) = \gamma_0^{-1}\mathbb{E}[\boldsymbol{\hbar}^{L\flat}(z)\boldsymbol{\varkappa}^{L\flat}_{q}(z)]. \tag{46}$$

If S_{dyn} denotes the Fourier exponent of the dynamical constraints other than the readout, variation with respect to f gives

$$0 = \gamma_0 N_{\flat}\widehat{f}_q(z') + \mathbb{E}\left[\frac{\partial S_{\mathrm{dyn}}}{\partial f_q(z')}\right], \qquad \widehat{f}_q(z') = 0. \tag{47}$$

Indeed, $f_q(\boldsymbol{x}',s)$ enters these histories only through loss feedback used at later steps $t>s$. Its derivative multiplies one of the four zero responses in (45) or an endpoint counterpart:

$$\begin{aligned}
&\mathbb{E}[-i\widehat{E}_{h,0}(z)\boldsymbol{\varkappa}^{0\flat}_{p}(z')] = 0,\\
&\mathbb{E}[-i\widehat{E}_{g,L,q}(z)\boldsymbol{\hbar}^{L\flat}(z')] = 0, \qquad z=(\boldsymbol{x},t),\quad z'=(\boldsymbol{x}',s),\quad s<t.
\end{aligned} \tag{48}$$

The additional loss derivatives are also past quantities. Thus no readout-conjugate term survives in the representative equations, while (46) determines their feedback.

Gaussian sources and Fourier responses close the equations. At this saddle the quadratic Fourier terms are decoupled by

$$\exp\left(-\frac{1}{2}\widehat{v}^{\mathsf{T}}C\widehat{v}\right) = \mathbb{E}_{v\sim\mathcal{N}(0,C)}\exp(-i\widehat{v}^{\mathsf{T}}v). \tag{49}$$

This characteristic-function identity also applies to singular positive-semidefinite kernels. The four Gaussian covariances are

$$C_{u^{\sharp}} = \Phi^{\flat}, \qquad C_{r^{\sharp}} = G^{(\ell+1)\flat}, \qquad C_{u^{\flat}} = \frac{\beta_0^2 N_{\flat}}{KL}\Phi^{\sharp}, \qquad C_{r^{\flat}} = \frac{\beta_0^2 N_{\flat}}{KL}G^{\sharp}. \tag{50}$$

For r, covariance indices include q, p. There are no mixed quadratic terms between these four Fourier families. Their Gaussian sources can therefore be taken independent across families and blocks, with their displayed within-family input, training-step, and output correlations.

Integrating the remaining linear Fourier fields gives

$$\begin{aligned}
\chi^{\sharp}(z) &= u^{\sharp}(z) + \frac{\beta_0 N_\flat}{K} \sum_{q,z'} A_q^{\flat}(z,z') \boldsymbol{g}_q^{\ell\sharp}(z'), \\
\chi^{\flat}(z) &= u^{\flat}(z) + \beta_0/L \sum_{q,z'} A_q^{\sharp}(z,z') \boldsymbol{g}_q^{(\ell+1)\flat}(z'), \\
\xi_q^{\flat}(z) &= r_q^{\flat}(z) + \sum_{z'} \widehat{A}_q^{\flat}(z',z) \boldsymbol{h}^{\ell\flat}(z'), \\
\xi_q^{\sharp}(z) &= r_q^{\sharp}(z) + \sum_{z'} \widehat{A}_q^{\sharp}(z',z) \phi(\boldsymbol{h}^{\ell\sharp}(z')).
\end{aligned} \tag{51}$$

The transposed arguments in the last two lines follow from (39). Because a source perturbation enters the Fourier exponent with coefficient $-i\widehat{v}$, its averaged response is

$$\mathbb{E} \frac{\partial \mathcal{O}}{\partial v(z')} = -i\, \mathbb{E}[\mathcal{O}\, \widehat{v}(z')]. \tag{52}$$

The normalization derivative is zero. This differentiates the full single-coordinate computation while holding the macroscopic kernels and feedback fixed; it does not treat $\mathcal{O}$ as independent of the source. Consequently

$$\begin{aligned}
A_q^{\flat}(z,z') &= \mathbb{E} \frac{\partial \boldsymbol{h}^{\ell\flat}(z)}{\partial r_q^{\flat}(z')}, \qquad A_q^{\sharp}(z,z') = \mathbb{E} \frac{\partial \phi(\boldsymbol{h}^{\ell\sharp}(z))}{\partial r_q^{\sharp}(z')}, \\
\widehat{A}_q^{\flat}(z,z') &= \beta_0/L\, B_q^{\sharp}(z',z), \quad \widehat{A}_q^{\sharp}(z,z') = \frac{\beta_0 N_\flat}{K} B_q^{\flat}(z',z).
\end{aligned} \tag{53}$$

Here $B_q^{\sharp}(z,z') = \mathbb{E}[\partial \boldsymbol{g}_q^{\ell\sharp}(z)/\partial u^{\sharp}(z')]$ and $B_q^{\flat}(z,z') = \mathbb{E}[\partial \boldsymbol{g}_q^{(\ell+1)\flat}(z)/\partial u^{\flat}(z')]$, as in (56). In particular, the signs of all four reaction terms are positive.

The population-counting coefficients can equivalently be written as

$$\frac{\beta_0}{L} \frac{N_\flat}{N_\sharp} = \frac{\beta_0 N_\flat}{K}, \qquad \left(\frac{\beta_0}{L}\right)^2 \frac{N_\flat}{N_\sharp} = \frac{\beta_0^2 N_\flat}{KL}. \tag{54}$$

Substituting (51) and (53) into the exact histories gives precisely (57) and (58), with source covariances (55). The Gaussian integration itself is finite-dimensional and exact; the population approximation is the saddle-point replacement of the empirical kernels and responses by their self-consistent values. Training time has remained discrete throughout.

B.3.2 Self-consistent fields and boundary conditions

For each layer introduce centered Gaussian source families $u^{\ell\sharp}, r_q^{\ell\sharp}, u^{(\ell+1)\flat}, r_q^{\ell\flat}$. The residual sources already include their branch coefficients. At the ratio $N_\flat/K$ in this fixed-depth width limit, their nonzero covariances are

$$\begin{aligned}
\mathbb{E}[u^{\ell\sharp}(z) u^{\ell\sharp}(z')] &= \Phi^{\ell\flat}(z,z'), \\
\mathbb{E}[r_q^{\ell\sharp}(z) r_p^{\ell\sharp}(z')] &= G_{qp}^{(\ell+1)\flat}(z,z'), \\
\mathbb{E}[u^{(\ell+1)\flat}(z) u^{(\ell+1)\flat}(z')] &= \frac{\beta_0^2 N_\flat}{KL} \Phi^{\ell\sharp}(z,z'), \\
\mathbb{E}[r_q^{\ell\flat}(z) r_p^{\ell\flat}(z')] &= \frac{\beta_0^2 N_\flat}{KL} G_{qp}^{\ell\sharp}(z,z').
\end{aligned} \tag{55}$$

Source families and layers are independent in this Gaussian-source representation. Their input, training-step, and output-component correlations are the kernels displayed above; the effects of using each matrix and its transpose enter through the responses below.

Define the averaged responses

$$
\begin{aligned}
A_p^{\ell\flat}(z,z') &= \mathbb{E}\frac{\partial \boldsymbol{\hbar}^{\ell\flat}(z)}{\partial r_p^{\ell\flat}(z')}, \qquad A_p^{\ell\sharp}(z,z') = \mathbb{E}\frac{\partial \phi(\boldsymbol{\hbar}^{\ell\sharp}(z))}{\partial r_p^{\ell\sharp}(z')}, \\
B_q^{\ell\flat}(z,z') &= \mathbb{E}\frac{\partial \boldsymbol{g}_q^{(\ell+1)\flat}(z)}{\partial u^{(\ell+1)\flat}(z')}, \quad B_q^{\ell\sharp}(z,z') = \mathbb{E}\frac{\partial \boldsymbol{g}_q^{\ell\sharp}(z)}{\partial u^{\ell\sharp}(z')}.
\end{aligned}
\tag{56}
$$

These derivatives perturb a source argument while holding the population kernels and macroscopic feedback fixed. They follow the full perturbed computation, including the activation derivative within $\boldsymbol{g}_q^{\ell\sharp}$. Nonsmooth activations require the corresponding derivatives in expectation.

The Gaussian integral and response identities above give hidden reaction coefficients $\beta_0 \frac{N_\flat}{K}$ and residual reaction coefficients β_0/L. Substituting them into the exact histories yields

$$
\begin{aligned}
\boldsymbol{\hbar}^{\ell\sharp}(z) &= u^{\ell\sharp}(z) + \beta_0 \frac{N_\flat}{K} \sum_{p,z'} A_p^{\ell\flat}(z,z') \boldsymbol{g}_p^{\ell\sharp}(z') \\
&\quad + \eta_\uparrow \gamma_0 \beta_0 \mathcal{S}_t[\boldsymbol{g}_p^{\ell\sharp}(z') \Phi^{\ell\flat}(z,z')], \\
\boldsymbol{g}_q^{\ell\sharp}(z) &= \phi'(\boldsymbol{\hbar}^{\ell\sharp}(z)) \left\{ \begin{array}{l} r_q^{\ell\sharp}(z) + \beta_0 \dfrac{N_\flat}{K} \displaystyle\sum_{z'} B_q^{\ell\flat}(z,z') \phi(\boldsymbol{\hbar}^{\ell\sharp}(z')) \\ \\ + \eta_\downarrow \gamma_0 \beta_0 \mathcal{S}_t[\phi(\boldsymbol{\hbar}^{\ell\sharp}(z')) G_{qp}^{(\ell+1)\flat}(z,z')] \end{array} \right\},
\end{aligned}
\tag{57}
$$

$$
\begin{aligned}
\boldsymbol{\hbar}^{(\ell+1)\flat}(z) - \boldsymbol{\hbar}^{\ell\flat}(z) &= u^{(\ell+1)\flat}(z) + \frac{\beta_0}{L} \sum_{p,z'} A_p^{\ell\sharp}(z,z') \boldsymbol{g}_p^{(\ell+1)\flat}(z') \\
&\quad + \frac{\eta_\downarrow \gamma_0 \beta_0^2}{L} \mathcal{S}_t[\boldsymbol{g}_p^{(\ell+1)\flat}(z') \Phi^{\ell\sharp}(z,z')], \\
\boldsymbol{g}_q^{\ell\flat}(z) - \boldsymbol{g}_q^{(\ell+1)\flat}(z) &= r_q^{\ell\flat}(z) + \frac{\beta_0}{L} \sum_{z'} B_q^{\ell\sharp}(z,z') \boldsymbol{\hbar}^{\ell\flat}(z') \\
&\quad + \frac{\eta_\uparrow \gamma_0 \beta_0^2}{L} \mathcal{S}_t[\boldsymbol{\hbar}^{\ell\flat}(z') G_{qp}^{\ell\sharp}(z,z')].
\end{aligned}
\tag{58}
$$

Here $\mathcal{S}_t$ retains the batches and earlier training steps of (24), with loss derivatives evaluated at the limiting output. Response sums run over $\mathcal{Z}$; the dependencies of the training computation determine which responses vanish.

For the boundaries, let $u_{\mathrm{in}}(\boldsymbol{x})$ have covariance $\kappa_{\mathrm{in}}(\boldsymbol{x},\boldsymbol{x}')$ and let $r_{\mathrm{out},q}$ be independent standard Gaussians. They are independent of the bulk sources and are reused at all training steps. The boundary equations and readout close the feedback loop:

$$
\begin{aligned}
\boldsymbol{\hbar}^{0\flat}(z) &= u_{\mathrm{in}}(\boldsymbol{x}) + \eta_{\mathrm{in}} \gamma_0 \mathcal{S}_t[\boldsymbol{g}_p^{0\flat}(z') \kappa_{\mathrm{in}}(\boldsymbol{x},\boldsymbol{x}')], \\
\boldsymbol{g}_q^{L\flat}(z) &= r_{\mathrm{out},q} + \eta_{\mathrm{out}} \gamma_0 \mathcal{S}_t[\delta_{pq} \boldsymbol{\hbar}^{L\flat}(z')], \\
f_q(z) &= \gamma_0^{-1} \mathbb{E}[\boldsymbol{\hbar}^{L\flat}(z) \boldsymbol{g}_q^{L\flat}(z)].
\end{aligned}
\tag{59}
$$

Together these equations determine kernels, responses, covariances, and loss feedback from one population law. The final identity uses the linear readout at depth L.

B.3.3 FINITE AND LIMITING OBJECTS

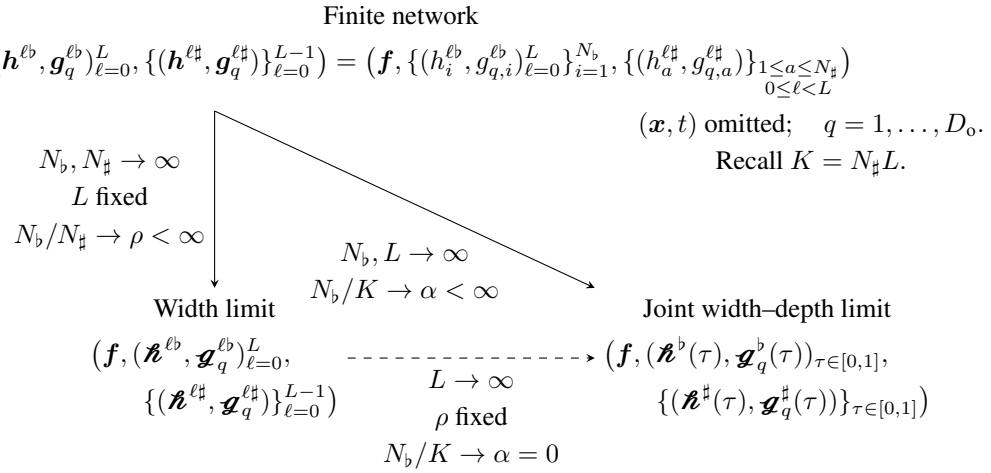


Figure 9: Finite random histories and two scaling paths. The dashed path keeps the residual-to-hidden ratio fixed before increasing depth, and therefore has $\alpha = 0$. The diagonal permits a positive limiting $N_\flat/K$. These are different scaling families, not an assertion that two identical limiting operations can be interchanged.

B.4 COMPLETE CONTINUOUS-DEPTH SGD SYSTEM

The continuous-depth system couples residual trajectories to local hidden variables and trainable input and readout maps. Take $L \to \infty$ with $N_\flat/K \to \alpha$ and write $\tau \in [0,1]$, $z = (\boldsymbol{x}, t)$, and $z' = (\boldsymbol{x}', s)$. Training steps remain discrete, with the four learning-rate constants in (20). Below, $\mathbb{E}_{\mathcal{B}_s}$ denotes the average over the examples $(\boldsymbol{x}', y')$ in batch $\mathcal{B}_s$, and $\Delta_p(\boldsymbol{x}', y', s) = -\partial_{f_p}\mathcal{L}_{\mathrm{ex}}(\boldsymbol{f}(\boldsymbol{x}', s), y')$. Response sums run over the input–step pairs $z' \in \mathcal{Z}$; learning sums run only over $s < t$.

The kernels are determined by the same population law:

$$\begin{aligned}
\Phi^{\flat}(\tau; z, z') &= \mathbb{E}[\boldsymbol{\mathcal{h}}^{\flat}(\tau, z)\boldsymbol{\mathcal{h}}^{\flat}(\tau, z')], \\
\Phi^{\sharp}(\tau; z, z') &= \mathbb{E}[\phi(\boldsymbol{\mathcal{h}}^{\sharp}(\tau, z))\phi(\boldsymbol{\mathcal{h}}^{\sharp}(\tau, z'))], \\
G^{\flat}_{qp}(\tau; z, z') &= \mathbb{E}[\boldsymbol{\mathcal{g}}^{\flat}_q(\tau, z)\boldsymbol{\mathcal{g}}^{\flat}_p(\tau, z')], \\
G^{\sharp}_{qp}(\tau; z, z') &= \mathbb{E}[\boldsymbol{\mathcal{g}}^{\sharp}_q(\tau, z)\boldsymbol{\mathcal{g}}^{\sharp}_p(\tau, z')].
\end{aligned} \tag{60a}$$

The centered Gaussian sources have covariances

$$\begin{aligned}
\mathbb{E}[du(\tau, z)du(\tau, z')] &= \beta_0^2\alpha\,\Phi^{\sharp}(\tau; z, z')\,d\tau, \\
\mathbb{E}[dr_q(\tau, z)dr_p(\tau, z')] &= \beta_0^2\alpha\,G^{\sharp}_{qp}(\tau; z, z')\,d\tau, \\
\mathbb{E}[u^{\sharp}(\tau, z)u^{\sharp}(\tau, z')] &= \Phi^{\flat}(\tau; z, z'), \\
\mathbb{E}[r^{\sharp}_q(\tau, z)r^{\sharp}_p(\tau, z')] &= G^{\flat}_{qp}(\tau; z, z').
\end{aligned} \tag{60b}$$

Different source families and depth slices are independent; correlations across inputs, training steps, and output components are retained above. In the four equations below, all bulk fields and kernels on the right-hand side are evaluated at the displayed depth τ.

The residual variables satisfy the depth-increment equations

$$
\begin{aligned}
d\boldsymbol{\hbar}^{\flat}(\tau,z) &= du(\tau,z) + \beta_0 \sum_{p,z'} \mathbb{E}\left[\frac{\delta\phi(\boldsymbol{\hbar}^{\sharp}(z))}{\delta r_p^{\sharp}(z')}\right] \boldsymbol{g}_p^{\flat}(z')\, d\tau \\
&\quad + \eta_{\downarrow}\gamma_0\beta_0^2 \sum_{s=0}^{t-1} \mathbb{E}_{\mathcal{B}_s}\left[\sum_p \Delta_p(\boldsymbol{x}',y',s)\boldsymbol{g}_p^{\flat}(z')\Phi^{\sharp}(z,z')\right] d\tau, \\
-d\boldsymbol{g}_q^{\flat}(\tau,z) &= dr_q(\tau,z) + \beta_0 \sum_{z'} \mathbb{E}\left[\frac{\delta \boldsymbol{g}_q^{\sharp}(z)}{\delta u^{\sharp}(z')}\right] \boldsymbol{\hbar}^{\flat}(z')\, d\tau \\
&\quad + \eta_{\uparrow}\gamma_0\beta_0^2 \sum_{s=0}^{t-1} \mathbb{E}_{\mathcal{B}_s}\left[\sum_p \Delta_p(\boldsymbol{x}',y',s)\boldsymbol{\hbar}^{\flat}(z')G_{qp}^{\sharp}(z,z')\right] d\tau.
\end{aligned}
\tag{60c}
$$

The minus sign records the backward difference $\boldsymbol{g}_q^{\ell\flat} - \boldsymbol{g}_q^{(\ell+1)\flat}$. The hidden variables do not accumulate over depth; at each slice they obey

$$
\begin{aligned}
\boldsymbol{\hbar}^{\sharp}(\tau,z) &= u^{\sharp}(\tau,z) + \beta_0\alpha \sum_{p,z'} \mathbb{E}\left[\frac{\delta \boldsymbol{\hbar}^{\flat}(z)}{\delta\, dr_p(z')}\right] \boldsymbol{g}_p^{\sharp}(z') \\
&\quad + \eta_{\uparrow}\gamma_0\beta_0 \sum_{s=0}^{t-1} \mathbb{E}_{\mathcal{B}_s}\left[\sum_p \Delta_p(\boldsymbol{x}',y',s)\boldsymbol{g}_p^{\sharp}(z')\Phi^{\flat}(z,z')\right], \\
\boldsymbol{g}_q^{\sharp}(\tau,z) &= \phi'(\boldsymbol{\hbar}^{\sharp}(\tau,z))\Bigg\{ r_q^{\sharp}(\tau,z) + \beta_0\alpha \sum_{z'} \mathbb{E}\left[\frac{\delta \boldsymbol{g}_q^{\flat}(z)}{\delta\, du(z')}\right] \phi(\boldsymbol{\hbar}^{\sharp}(z')) \\
&\quad + \eta_{\downarrow}\gamma_0\beta_0 \sum_{s=0}^{t-1} \mathbb{E}_{\mathcal{B}_s}\left[\sum_p \Delta_p(\boldsymbol{x}',y',s)\phi(\boldsymbol{\hbar}^{\sharp}(z'))G_{qp}^{\flat}(z,z')\right]\Bigg\}.
\end{aligned}
\tag{60d}
$$

Responses differentiate the full perturbed computation at fixed population kernels and loss feedback. In particular, differentiating $\boldsymbol{g}_q^{\sharp}$ includes its factor $\phi'(\boldsymbol{\hbar}^{\sharp})$. Derivatives with respect to du and dr mean perturbing a single depth increment before taking the continuous-depth limit:

$$
\begin{aligned}
\mathbb{E}\frac{\delta \boldsymbol{\hbar}^{\flat}(\tau,z)}{\delta\, dr_p(\tau,z')} &= \lim_{\substack{L\to\infty \\ \ell/L\to\tau}} \mathbb{E}\frac{\partial \boldsymbol{\hbar}^{\ell\flat}(z)}{\partial r_p^{\ell\flat}(z')}, \\
\mathbb{E}\frac{\delta \boldsymbol{g}_q^{\flat}(\tau,z)}{\delta\, du(\tau,z')} &= \lim_{\substack{L\to\infty \\ \ell/L\to\tau}} \mathbb{E}\frac{\partial \boldsymbol{g}_q^{(\ell+1)\flat}(z)}{\partial u^{(\ell+1)\flat}(z')}.
\end{aligned}
\tag{60e}
$$

The finite-slice sources already include the residual coupling, so this change of notation introduces no additional scaling factor.

Training the input and readout maps supplies the boundary conditions. Let $u_{\mathrm{in}}(\boldsymbol{x})$ and $r_{\mathrm{out},q}$ be centered Gaussian sources, independent of each other and of the bulk sources, with

$$
\mathbb{E}[u_{\mathrm{in}}(\boldsymbol{x})u_{\mathrm{in}}(\boldsymbol{x}')] = \kappa_{\mathrm{in}}(\boldsymbol{x},\boldsymbol{x}') = \frac{\boldsymbol{x}^{\mathsf{T}}\boldsymbol{x}'}{D_{\mathrm{i}}}, \quad \mathbb{E}[r_{\mathrm{out},q}r_{\mathrm{out},p}] = \delta_{qp}. \tag{60f}
$$

The same sources are reused at every training step. The boundary updates and output are

$$
\begin{aligned}
\boldsymbol{\hbar}^{\flat}(0,z) &= u_{\mathrm{in}}(\boldsymbol{x}) + \eta_{\mathrm{in}}\gamma_0 \sum_{s=0}^{t-1} \mathbb{E}_{\mathcal{B}_s}\left[\sum_p \Delta_p(\boldsymbol{x}',y',s)\boldsymbol{g}_p^{\flat}(0,z')\kappa_{\mathrm{in}}(\boldsymbol{x},\boldsymbol{x}')\right], \\
\boldsymbol{g}_q^{\flat}(1,z) &= r_{\mathrm{out},q} + \eta_{\mathrm{out}}\gamma_0 \sum_{s=0}^{t-1} \mathbb{E}_{\mathcal{B}_s}\left[\Delta_q(\boldsymbol{x}',y',s)\boldsymbol{\hbar}^{\flat}(1,z')\right], \\
f_q(z) &= \frac{1}{\gamma_0}\mathbb{E}[\boldsymbol{\hbar}^{\flat}(1,z)\boldsymbol{g}_q^{\flat}(1,z)] \\
&= \frac{1}{\gamma_0}\mathbb{E}[\boldsymbol{\hbar}^{\flat}(1,z)r_{\mathrm{out},q}] + \eta_{\mathrm{out}} \sum_{s=0}^{t-1} \mathbb{E}_{\mathcal{B}_s}\left[\Delta_q(\boldsymbol{x}',y',s)\Phi^{\flat}(1;z,z')\right].
\end{aligned}
\tag{60g}
$$

The output determines the loss feedback Δ_p for the next simultaneous SGD update, closing the system.

With deterministic population coefficients, the residual system and its boundary conditions are linear in the jointly Gaussian sources. A unique solution selected by the finite-step computation therefore has jointly Gaussian residual histories, despite its nonlinear hidden populations and changing covariance kernels. The finite identities and Gaussian weight averages are exact; the population description uses their self-consistent saddle. One route takes width large at fixed L, with $N_\flat/N_\sharp = (N_\flat/K)L$, then increases L. Uniform control of the remainders connects this construction to simultaneous growth, including bounded hidden width. The same weight integral extends to tensor-valued channels below, followed by the corresponding Muon and Adam optimizer-state histories.

B.5 Attention and other finite-dimensional channels

System (60) extends to finite-dimensional nonlinear channels by retaining token and internal coordinates as explicit tensor axes, separate from the residual and channel populations being averaged. The scalar activation and its backward derivative become a channel map and its full adjoint Jacobian; kernels retain both token arguments and contract internal coordinates without averaging over them. An entire attention head is one such channel.

B.5.1 Tensor population equations

The up–nonlinearity–down structure also describes attention when one channel contains a full head. Fix the number of tokens S and the head dimensions $d_{\mathrm{k}}, d_{\mathrm{v}}$, and let $N_{\sharp\mathrm{A}}$ be the number of heads in each of L attention blocks. Write $K_{\mathrm{A}} = LN_{\sharp\mathrm{A}}$ and set $d_\uparrow = 2d_{\mathrm{k}} + d_{\mathrm{v}}$. For token states $\boldsymbol{h}^\flat_{\mathfrak{s}} \in \mathbb{R}^{N_\flat}$, take $\boldsymbol{W}^\uparrow_a \in \mathbb{R}^{d_\uparrow \times N_\flat}$ and $\boldsymbol{W}^\downarrow_a \in \mathbb{R}^{N_\flat \times d_{\mathrm{v}}}$. Suppressing the layer index, one block is

$$\begin{aligned} \boldsymbol{h}^{\sharp\mathrm{A}}_{a,\mathfrak{s}} &= \frac{1}{\sqrt{N_\flat}} \boldsymbol{W}^\uparrow_a \boldsymbol{h}^\flat_{\mathfrak{s}}, \qquad & (\boldsymbol{o}_{a,\mathfrak{s}})_{\mathfrak{s}=1}^{S} &= \psi_S((\boldsymbol{h}^{\sharp\mathrm{A}}_{a,\mathfrak{s}})_{\mathfrak{s}=1}^{S}), \\ \boldsymbol{h}^{+\flat}_{\mathfrak{s}} &= \boldsymbol{h}^\flat_{\mathfrak{s}} + \frac{\beta_0\sqrt{N_\flat}}{K_{\mathrm{A}}} \sum_{a=1}^{N_{\sharp\mathrm{A}}} \boldsymbol{W}^\downarrow_a \boldsymbol{o}_{a,\mathfrak{s}}. \end{aligned} \tag{61}$$

The input and output maps retain the main-text conventions, applied tokenwise or followed by a fixed readout over tokens. All entries of $\boldsymbol{W}^\uparrow_a, \boldsymbol{W}^\downarrow_a$ may be initialized as independent standard Gaussians; fixed factors associated with the head dimensions can be included in ψ_S or in the projection constants.

For a general channel map $\psi_S : \mathbb{R}^{S\times d_\uparrow} \to \mathbb{R}^{S\times d_\downarrow}$, use $K = N_\sharp L$. For attention, $N_\sharp, K$ denote the head count and budget $N_{\sharp\mathrm{A}}, K_{\mathrm{A}}$ in (61). The numbers $S, d_\uparrow, d_\downarrow$ are fixed. An attention head is one channel, including its complete token-coupled map ψ_S; its internal dimension is not an additional width limit. Let $\upsilon = (z, \mathfrak{s})$, where $z = (\boldsymbol{x}, t)$ and $\mathfrak{s}$ is a token index. Output indices q, p are distinct from token indices. For a representative channel, introduce a pre-Jacobian cotangent $R_{\ell,q}(z) \in \mathbb{R}^{S\times d_\downarrow}$, so that

$$\boldsymbol{\mathscr{g}}^{\ell\sharp}_{q,\mathfrak{s}b}(z) = \sum_{\mathfrak{p}=1}^{S}\sum_{c=1}^{d_\downarrow} \frac{\partial\psi_S(\boldsymbol{\mathscr{h}}^{\ell\sharp}(z))_{\mathfrak{p}c}}{\partial \boldsymbol{\mathscr{h}}^{\ell\sharp}_{\mathfrak{s}b}(z)} R_{\ell,q,\mathfrak{p}c}(z). \tag{62}$$

This retains the full channel Jacobian, including its cross-token entries.

The four population kernels are

$$\begin{aligned} \Psi^{\ell\flat}(\upsilon,\upsilon') &= \mathbb{E}[\boldsymbol{\mathscr{h}}^{\ell\flat}(\upsilon)\boldsymbol{\mathscr{h}}^{\ell\flat}(\upsilon')], \quad \Psi^{\ell\sharp}(\upsilon,\upsilon') = \mathbb{E}\sum_{c=1}^{d_\downarrow} \psi_S(\boldsymbol{\mathscr{h}}^{\ell\sharp}(z))_{\mathfrak{s}c}\psi_S(\boldsymbol{\mathscr{h}}^{\ell\sharp}(z'))_{\mathfrak{s}'c}, \\ G^{\ell\flat}_{qp}(\upsilon,\upsilon') &= \mathbb{E}[\boldsymbol{\mathscr{g}}^{\ell\flat}_q(\upsilon)\boldsymbol{\mathscr{g}}^{\ell\flat}_p(\upsilon')], \quad G^{\ell\sharp}_{qp}(\upsilon,\upsilon') = \mathbb{E}\sum_{b=1}^{d_\uparrow} \boldsymbol{\mathscr{g}}^{\ell\sharp}_{q,b}(\upsilon)\boldsymbol{\mathscr{g}}^{\ell\sharp}_{p,b}(\upsilon'). \end{aligned} \tag{63}$$

The corresponding empirical kernels divide by $N_\flat$ or $N_\sharp$, respectively, with no division by $S, d_\uparrow$, or $d_\downarrow$. Thus each kernel retains both token arguments, while the displayed internal-coordinate contractions are unnormalized traces.

The centered Gaussian sources have nonzero covariances

$$\begin{aligned}
\mathbb{E}[u_b^{\ell\sharp}(\upsilon)u_{b'}^{\ell\sharp}(\upsilon')] &= \delta_{bb'}\Psi^{\ell\flat}(\upsilon,\upsilon'),\\
\mathbb{E}[r_{q,c}^{\ell\sharp}(\upsilon)r_{p,c'}^{\ell\sharp}(\upsilon')] &= \delta_{cc'}G_{qp}^{(\ell+1)\flat}(\upsilon,\upsilon'),\\
\mathbb{E}[u^{(\ell+1)\flat}(\upsilon)u^{(\ell+1)\flat}(\upsilon')] &= \frac{\beta_0^2 N_\flat}{KL}\Psi^{\ell\sharp}(\upsilon,\upsilon'),\\
\mathbb{E}[r_q^{\ell\flat}(\upsilon)r_p^{\ell\flat}(\upsilon')] &= \frac{\beta_0^2 N_\flat}{KL}G_{qp}^{\ell\sharp}(\upsilon,\upsilon').
\end{aligned} \tag{64}$$

Different source families and layers are independent in this representation; the resulting trained fields are not. In particular, $r_q^{\ell\sharp}$ has $d_\downarrow$ coordinates before the Jacobian in (62).

The four averaged responses are

$$\begin{aligned}
A_p^{\ell\flat}(\upsilon,\upsilon') &= \mathbb{E}\frac{\partial \boldsymbol{\hbar}^{\ell\flat}(\upsilon)}{\partial r_p^{\ell\flat}(\upsilon')}, \qquad A_p^{\ell\sharp}(\upsilon,\upsilon') = \mathbb{E}\sum_{c=1}^{d_\downarrow}\frac{\partial \psi_S(\boldsymbol{\hbar}^{\ell\sharp}(z))_{\mathfrak{s}c}}{\partial r_{p,c}^{\ell\sharp}(\upsilon')},\\
B_q^{\ell\flat}(\upsilon,\upsilon') &= \mathbb{E}\frac{\partial \boldsymbol{\mathscr{g}}_q^{(\ell+1)\flat}(\upsilon)}{\partial u^{(\ell+1)\flat}(\upsilon')}, \quad B_q^{\ell\sharp}(\upsilon,\upsilon') = \mathbb{E}\sum_{b=1}^{d_\uparrow}\frac{\partial \boldsymbol{\mathscr{g}}_{q,b}^{\ell\sharp}(\upsilon)}{\partial u_b^{\ell\sharp}(\upsilon')}.
\end{aligned} \tag{65}$$

As in (56), these derivatives hold the population kernels and loss feedback fixed and differentiate the entire perturbed computation. The two hidden responses trace the matching internal index but retain the two token arguments, including cross-token responses. The self-consistent hidden and residual equations are

$$\begin{aligned}
\boldsymbol{\hbar}_{\mathfrak{s}b}^{\ell\sharp}(z) &= u_{\mathfrak{s}b}^{\ell\sharp}(z) + \beta_0\frac{N_\flat}{K}\sum_{p,z',\mathfrak{r}}A_p^{\ell\flat}(z,\mathfrak{s};z',\mathfrak{r})\boldsymbol{\mathscr{g}}_{p,\mathfrak{r}b}^{\ell\sharp}(z')\\
&\quad + \eta_\uparrow\gamma_0\beta_0\,\mathcal{S}_t\left[\sum_{\mathfrak{r}}\Psi^{\ell\flat}(z,\mathfrak{s};z',\mathfrak{r})\boldsymbol{\mathscr{g}}_{p,\mathfrak{r}b}^{\ell\sharp}(z')\right],\\
R_{\ell,q,\mathfrak{s}c}(z) &= r_{q,\mathfrak{s}c}^{\ell\sharp}(z) + \beta_0\frac{N_\flat}{K}\sum_{z',\mathfrak{r}}B_q^{\ell\flat}(z,\mathfrak{s};z',\mathfrak{r})\psi_S(\boldsymbol{\hbar}^{\ell\sharp}(z'))_{\mathfrak{r}c}\\
&\quad + \eta_\downarrow\gamma_0\beta_0\,\mathcal{S}_t\left[\sum_{\mathfrak{r}}G_{qp}^{(\ell+1)\flat}(z,\mathfrak{s};z',\mathfrak{r})\psi_S(\boldsymbol{\hbar}^{\ell\sharp}(z'))_{\mathfrak{r}c}\right].
\end{aligned} \tag{66}$$

$$\begin{aligned}
\boldsymbol{\hbar}_{\mathfrak{s}}^{(\ell+1)\flat}(z) - \boldsymbol{\hbar}_{\mathfrak{s}}^{\ell\flat}(z) &= u_{\mathfrak{s}}^{(\ell+1)\flat}(z) + \frac{\beta_0}{L}\sum_{p,z',\mathfrak{r}}A_p^{\ell\sharp}(z,\mathfrak{s};z',\mathfrak{r})\boldsymbol{\mathscr{g}}_{p,\mathfrak{r}}^{(\ell+1)\flat}(z')\\
&\quad + \frac{\eta_\downarrow\gamma_0\beta_0^2}{L}\,\mathcal{S}_t\left[\sum_{\mathfrak{r}}\Psi^{\ell\sharp}(z,\mathfrak{s};z',\mathfrak{r})\boldsymbol{\mathscr{g}}_{p,\mathfrak{r}}^{(\ell+1)\flat}(z')\right],\\
\boldsymbol{\mathscr{g}}_{q,\mathfrak{s}}^{\ell\flat}(z) - \boldsymbol{\mathscr{g}}_{q,\mathfrak{s}}^{(\ell+1)\flat}(z) &= r_{q,\mathfrak{s}}^{\ell\flat}(z) + \frac{\beta_0}{L}\sum_{z',\mathfrak{r}}B_q^{\ell\sharp}(z,\mathfrak{s};z',\mathfrak{r})\boldsymbol{\hbar}_{\mathfrak{r}}^{\ell\flat}(z')\\
&\quad + \frac{\eta_\uparrow\gamma_0\beta_0^2}{L}\,\mathcal{S}_t\left[\sum_{\mathfrak{r}}G_{qp}^{\ell\sharp}(z,\mathfrak{s};z',\mathfrak{r})\boldsymbol{\hbar}_{\mathfrak{r}}^{\ell\flat}(z')\right].
\end{aligned} \tag{67}$$

The response sums run over the same finite query set as in the MLP construction. Only the history terms carry the loss-weighted, strictly earlier-step sum $\mathcal{S}_t$.

For completeness, a shared input map and a fixed linear token readout make the boundary contractions explicit. Let $h_{i\mathfrak{s}}^{0\flat} = D_{\mathrm{i}}^{-1/2}\sum_j W_{ij}^0 x_{\mathfrak{s}j}$ and $f_q = (\gamma_0 N_\flat)^{-1}\sum_{i,\mathfrak{s}}W_{qi}^L\omega_{\mathfrak{s}}h_{i\mathfrak{s}}^{L\flat}$, where $\omega_{\mathfrak{s}}$ is fixed.

With $\kappa_{\mathrm{in}}(\boldsymbol{x},\mathfrak{s};\boldsymbol{x}',\mathfrak{r}) = \boldsymbol{x}_{\mathfrak{s}}^{\mathsf{T}}\boldsymbol{x}'_{\mathfrak{r}}/D_{\mathrm{i}}$, the counterparts of (59) are

$$\begin{aligned}
\boldsymbol{\hbar}_{\mathfrak{s}}^{0\flat}(z) &= u_{\mathrm{in},\mathfrak{s}}(\boldsymbol{x}) + \eta_{\mathrm{in}}\gamma_0\,\mathcal{S}_t\left[\sum_{\mathfrak{r}} \kappa_{\mathrm{in}}(\boldsymbol{x},\mathfrak{s};\boldsymbol{x}',\mathfrak{r})\boldsymbol{g}_{p,\mathfrak{r}}^{0\flat}(z')\right],\\
\boldsymbol{g}_{q,\mathfrak{s}}^{L\flat}(z) &= \omega_{\mathfrak{s}}\left\{r_{\mathrm{out},q} + \eta_{\mathrm{out}}\gamma_0\,\mathcal{S}_t\left[\delta_{pq}\sum_{\mathfrak{r}}\omega_{\mathfrak{r}}\boldsymbol{\hbar}_{\mathfrak{r}}^{L\flat}(z')\right]\right\},\\
f_q(z) &= \gamma_0^{-1}\mathbb{E}\sum_{\mathfrak{s}}\boldsymbol{\hbar}_{\mathfrak{s}}^{L\flat}(z)\boldsymbol{g}_{q,\mathfrak{s}}^{L\flat}(z).
\end{aligned} \tag{68}$$

Here u_{in} has covariance κ_{in} and $r_{\mathrm{out},q}$ are independent standard Gaussians, reused throughout training as in the MLP construction. Other fixed token readouts give their corresponding finite contractions. Equations (63)–(68), including the loss derivatives obtained from this readout, form the general-channel population closure.

Setting $S = d_\uparrow = d_\downarrow = 1$, $\omega_1 = 1$, and $\psi_1 = \phi$ identifies these Ψ kernels with the MLP Φ kernels and recovers (57)–(59). The continuous-depth passage uses the same equations with the displayed token kernels: residual source covariances become $\beta_0^2\alpha\Psi^\sharp\,d\tau$ and $\beta_0^2\alpha G^\sharp\,d\tau$, and each residual reaction or training term proportional to $1/L$ becomes its corresponding drift. The hidden equations remain local to the depth τ, with $N_\flat/K$ replaced by α. The fixed internal dimensions remain inside ψ_S, its Jacobian, and the unnormalized traces throughout.

The suppression of local learning can also be read directly from these general-channel equations. At fixed $c = \beta_0^2\alpha$, the hidden reaction coefficient is $\beta_0\alpha = c/\beta_0$, while the two direct SGD history coefficients are $\eta_{\mathrm{eff},j} = \eta_j\gamma_0\beta_0$, $j \in \{\uparrow,\downarrow\}$. Along a family with $\eta_{\mathrm{eff},j} = O(\beta_0^{-1})$, (66) gives, on a finite query and training window with bounded residual kernels, responses, loss feedback, and channel second moments,

$$\|\boldsymbol{\hbar}_\beta^\sharp - u_\beta^\sharp\|_{L^2} + \|R_\beta - r_\beta^\sharp\|_{L^2} = O(\beta_0^{-1}). \tag{69}$$

Thus both the direct learning histories and the reaction corrections disappear from the local hidden equations. This applies to a finite-dimensional channel map ψ_S, including attention, through its full Jacobian rather than a coordinatewise activation derivative. The channel output $\psi_S(\boldsymbol{\hbar}_\beta^\sharp)$ and its backward Jacobian action can remain nonlinear, and the Gaussian-source covariance may still evolve through the residual dynamics. The degeneration concerns direct local learning, not the disappearance of all activation changes.

B.5.2 Gaussian weight calculation

The Gaussian weight integral makes the required contractions explicit. For the finite fields at layer ℓ, use $h_{i\mathfrak{s}}^{\ell\flat}, g_{q,i\mathfrak{s}}^{\ell\flat}, h_{a\mathfrak{s}b}^{\ell\sharp}, \psi_S(\boldsymbol{h}_a^{\ell\sharp})_{\mathfrak{s}c}, g_{q,a\mathfrak{s}b}^{\ell\sharp}$ for the finite fields before selecting representative coordinates. The four initial-weight projections are

$$\begin{aligned}
\chi_{a\mathfrak{s}b}^\sharp &= N_\flat{}^{-1/2}\sum_i W_{abi}^{\ell\uparrow}(0)h_{i\mathfrak{s}}^{\ell\flat}, & \xi_{q,a\mathfrak{s}c}^\sharp &= N_\flat{}^{-1/2}\sum_i W_{iac}^{\ell\downarrow}(0)g_{q,i\mathfrak{s}}^{(\ell+1)\flat},\\
\chi_{i\mathfrak{s}}^\flat &= (\beta_0\sqrt{N_\flat}/K)\sum_{a,c} W_{iac}^{\ell\downarrow}(0)\psi_S(\boldsymbol{h}_a^{\ell\sharp})_{\mathfrak{s}c}, & \xi_{q,i\mathfrak{s}}^\flat &= (\beta_0\sqrt{N_\flat}/K)\sum_{a,b} W_{abi}^{\ell\uparrow}(0)g_{q,a\mathfrak{s}b}^{\ell\sharp}.
\end{aligned} \tag{70}$$

The backward residual field is evaluated at $\ell+1$, and the residual projections already include their branch coefficient. After introducing these constraints and their Fourier fields, the coefficients of the individual Gaussian entries are

$$\begin{aligned}
J_{abi}^\uparrow &= \sum_{z,\mathfrak{s}}\left[\frac{\widehat{\chi}_{a\mathfrak{s}b}^\sharp(z)h_{i\mathfrak{s}}^{\ell\flat}(z)}{\sqrt{N_\flat}} + (\beta_0\sqrt{N_\flat}/K)\sum_q \widehat{\xi}_{q,i\mathfrak{s}}^\flat(z)g_{q,a\mathfrak{s}b}^{\ell\sharp}(z)\right],\\
J_{iac}^\downarrow &= \sum_{z,\mathfrak{s}}\left[(\beta_0\sqrt{N_\flat}/K)\widehat{\chi}_{i\mathfrak{s}}^\flat(z)\psi_S(\boldsymbol{h}_a^{\ell\sharp}(z))_{\mathfrak{s}c} + \frac{1}{\sqrt{N_\flat}}\sum_q \widehat{\xi}_{q,a\mathfrak{s}c}^\sharp(z)g_{q,i\mathfrak{s}}^{(\ell+1)\flat}(z)\right].
\end{aligned} \tag{71}$$

Consequently the exact initial-weight average in this constrained integral is

$$\mathbb{E}_{\boldsymbol{W}^{\ell\uparrow}(0),\boldsymbol{W}^{\ell\downarrow}(0)} e^{-i\sum_{a,b,i} W^{\ell\uparrow}_{abi}(0)J^{\uparrow}_{abi} - i\sum_{i,a,c} W^{\ell\downarrow}_{iac}(0)J^{\downarrow}_{iac}} = \exp\left[-\frac{1}{2}\sum_{a,b,i}(J^{\uparrow}_{abi})^2 - \frac{1}{2}\sum_{i,a,c}(J^{\downarrow}_{iac})^2\right]. \tag{72}$$

The same weight is used at every token, so these squares contain double token sums, rather than a token Kronecker delta. For example, the forward down-projection term is

$$-\frac{(\beta_0\sqrt{N_\flat}/K)^2}{2}\sum_{\upsilon,\upsilon'}\left[\sum_i \widehat{\chi}^{\flat}_i(\upsilon)\widehat{\chi}^{\flat}_i(\upsilon')\right]\left[\sum_{a,c}\psi_S(\boldsymbol{h}^{\ell\sharp}_a(z))_{\mathfrak{s}c}\psi_S(\boldsymbol{h}^{\ell\sharp}_a(z'))_{\mathfrak{s}'c}\right]. \tag{73}$$

The mixed terms have coefficient $(\beta_0\sqrt{N_\flat}/K)/\sqrt{N_\flat} = \beta_0/K$; they generate the weight-reuse responses in (65). Summing the mixed weight term over residual coordinates gives $\beta_0 N_\flat/K = \beta_0\frac{N_\flat}{K}$; summing it over channels gives $\beta_0 N_\sharp/K = \beta_0/L$.

The training terms follow from synchronous SGD without taking a gradient-flow limit. In the mathematical weight coordinates, (77) gives the exact histories

$$\begin{aligned} W^{\ell\uparrow}_{abi}(t) &= W^{\ell\uparrow}_{abi}(0) + \frac{\eta_\uparrow\gamma_0\beta_0}{\sqrt{N_\flat}}\mathcal{S}_t\left[\sum_{\mathfrak{r}=1}^{S} g^{\ell\sharp}_{p,a\mathfrak{r}b}(z')h^{\ell\flat}_{i\mathfrak{r}}(z')\right], \\ W^{\ell\downarrow}_{iac}(t) &= W^{\ell\downarrow}_{iac}(0) + \frac{\eta_\downarrow\gamma_0\beta_0}{\sqrt{N_\flat}}\mathcal{S}_t\left[\sum_{\mathfrak{r}=1}^{S} g^{(\ell+1)\flat}_{p,i\mathfrak{r}}(z')\psi_S(\boldsymbol{h}^{\ell\sharp}_a(z'))_{\mathfrak{r}c}\right]. \end{aligned} \tag{74}$$

All fields inside $\mathcal{S}_t$ are evaluated before the corresponding simultaneous update, as specified in (24). Substituting these histories and decoupling the initial-weight integral gives (66)–(68).

B.5.3 Attention-specific structure

For masked softmax attention, split $\boldsymbol{h}^{\sharp\mathrm{A}}_{a,\mathfrak{s}}$ into $(\boldsymbol{q}_{a,\mathfrak{s}}, \boldsymbol{k}_{a,\mathfrak{s}}, \boldsymbol{v}_{a,\mathfrak{s}})$ with dimensions $(d_\mathrm{k}, d_\mathrm{k}, d_\mathrm{v})$ and define

$$\begin{aligned} p_{a,\mathfrak{s}\mathfrak{r}} &= \frac{M_{\mathfrak{s}\mathfrak{r}}\exp(\boldsymbol{q}^{\mathsf{T}}_{a,\mathfrak{s}}\boldsymbol{k}_{a,\mathfrak{r}}/\sqrt{d_\mathrm{k}})}{\sum_{\mathfrak{r}'=1}^{S} M_{\mathfrak{s}\mathfrak{r}'}\exp(\boldsymbol{q}^{\mathsf{T}}_{a,\mathfrak{s}}\boldsymbol{k}_{a,\mathfrak{r}'}/\sqrt{d_\mathrm{k}})}, \\ \boldsymbol{o}_{a,\mathfrak{s}} &= \sum_{\mathfrak{r}=1}^{S} p_{a,\mathfrak{s}\mathfrak{r}}\boldsymbol{v}_{a,\mathfrak{r}}. \end{aligned} \tag{75}$$

Here M is a fixed binary mask with at least one visible token per row. The hidden computation mixes tokens and head coordinates, while the residual coordinate is accessed only through the two linear maps. The count corresponding to $N_\sharp$ is the number of heads, not the head dimension. In a network alternating MLP and attention updates, the two populations have separate counts $N_{\sharp\mathrm{M}}, N_{\sharp\mathrm{A}}$ and budgets $K_\mathrm{M} = L_\mathrm{M}N_{\sharp\mathrm{M}}$, $K_\mathrm{A} = L_\mathrm{A}N_{\sharp\mathrm{A}}$, with $L_\mathrm{M}, L_\mathrm{A}$ counting their respective residual updates.

The backward variables use the same rescaling as in the MLP. Regard $\boldsymbol{h}^{\sharp\mathrm{A}}_a$ and $\boldsymbol{o}_a$ as arrays over tokens, and let $D\psi_S(\boldsymbol{h}^{\sharp\mathrm{A}}_a)^*$ be the adjoint Jacobian of the head computation. For output component q, set $\boldsymbol{g}^{\flat}_{q,\mathfrak{s}} = \gamma_0 N_\flat\,\partial f_q/\partial\boldsymbol{h}^{\flat}_{\mathfrak{s}}$ and $\boldsymbol{g}^{\sharp\mathrm{A}}_{q,a,\mathfrak{s}} = (\gamma_0 K_\mathrm{A}/\beta_0)\,\partial f_q/\partial\boldsymbol{h}^{\sharp\mathrm{A}}_{a,\mathfrak{s}}$. The chain rule gives

$$\begin{aligned} \boldsymbol{g}^{\sharp\mathrm{A}}_{q,a} &= \frac{1}{\sqrt{N_\flat}}D\psi_S(\boldsymbol{h}^{\sharp\mathrm{A}}_a)^*\big(((\boldsymbol{W}^{\downarrow}_a)^{\mathsf{T}}\boldsymbol{g}^{+\flat}_{q,\mathfrak{s}})^{S}_{\mathfrak{s}=1}\big), \\ \boldsymbol{g}^{\flat}_{q,\mathfrak{s}} &= \boldsymbol{g}^{+\flat}_{q,\mathfrak{s}} + \frac{\beta_0\sqrt{N_\flat}}{K_\mathrm{A}}\sum_{a=1}^{N_{\sharp\mathrm{A}}}(\boldsymbol{W}^{\uparrow}_a)^{\mathsf{T}}\boldsymbol{g}^{\sharp\mathrm{A}}_{q,a,\mathfrak{s}}. \end{aligned} \tag{76}$$

Thus the hidden derivative is the Jacobian of a finite-dimensional channel, in place of multiplication by the scalar ϕ'. The two parameter Jacobians are

$$\begin{aligned} \frac{\partial f_q}{\partial\boldsymbol{W}^{\uparrow}_a} &= \frac{\beta_0}{\gamma_0 K_\mathrm{A}\sqrt{N_\flat}}\sum_{\mathfrak{s}=1}^{S}\boldsymbol{g}^{\sharp\mathrm{A}}_{q,a,\mathfrak{s}}(\boldsymbol{h}^{\flat}_{\mathfrak{s}})^{\mathsf{T}}, \\ \frac{\partial f_q}{\partial\boldsymbol{W}^{\downarrow}_a} &= \frac{\beta_0}{\gamma_0 K_\mathrm{A}\sqrt{N_\flat}}\sum_{\mathfrak{s}=1}^{S}\boldsymbol{g}^{+\flat}_{q,\mathfrak{s}}\boldsymbol{o}^{\mathsf{T}}_{a,\mathfrak{s}}. \end{aligned} \tag{77}$$

Token summation changes the finite internal contractions, while the size factors in the forward, backward, and parameter updates retain the MLP form.

At initialization, this replacement also specifies the covariance map. Index a finite collection of inputs and tokens by $z = (\boldsymbol{x}, \mathfrak{s})$, and write $C_{zz'} = N_\flat{}^{-1}(\boldsymbol{h}_z^\flat)^\mathsf{T}\boldsymbol{h}_{z'}^\flat$. Let Z be the Gaussian array with $\mathbb{E}[Z_{r,z}Z_{r',z'}] = \delta_{rr'}C_{zz'}$, where $r = 1, \ldots, d_\uparrow$, and define

$$[\mathcal{V}_\psi(C)]_{zz'} = \mathbb{E}\big[\psi_S(Z(\boldsymbol{x}))_\mathfrak{s}^\mathsf{T}\psi_S(Z(\boldsymbol{x}'))_{\mathfrak{s}'}\big]\,. \tag{78}$$

For the block increment $\delta\boldsymbol{h}_z^\flat$,

$$\mathbb{E}[\delta h_{i,z}^\flat \delta h_{j,z'}^\flat \mid \boldsymbol{h}^\flat] = \delta_{ij}\frac{\beta_0^2(N_\flat/K_\mathrm{A})}{L}[\mathcal{V}_\psi(C)]_{zz'}. \tag{79}$$

The continuous-depth initialization covariance therefore obeys $\partial_\tau C = \beta_0^2\alpha_\mathrm{A}\mathcal{V}_\psi(C)$ when $N_\flat/K_\mathrm{A} \to \alpha_\mathrm{A}$. The covariance map keeps the attention nonlinearity and token interactions; the scalar MLP map is its one-dimensional channel case.

This finite computation has the same residual rotation symmetry as the MLP example. For a common orthogonal $\boldsymbol{R} \in \mathbb{R}^{N_\flat \times N_\flat}$, transform every residual vector by $\boldsymbol{R}$, every up map by $\boldsymbol{W}_a^\uparrow \mapsto \boldsymbol{W}_a^\uparrow\boldsymbol{R}^\mathsf{T}$, and every down map by $\boldsymbol{W}_a^\downarrow \mapsto \boldsymbol{R}\boldsymbol{W}_a^\downarrow$; the input and output weights transform as $\boldsymbol{W}^0 \mapsto \boldsymbol{R}\boldsymbol{W}^0$ and $\boldsymbol{W}^L \mapsto \boldsymbol{W}^L\boldsymbol{R}^\mathsf{T}$. Each $\boldsymbol{h}_{a,\mathfrak{s}}^{\sharp\mathrm{A}}$, attention matrix, and output is unchanged. Gradients transform in the corresponding way by the chain rule. Independent isotropic Gaussian initialization, SGD with scalar rates for each parameter group, and orthogonally equivariant spectral updates preserve this symmetry. Consequently the residual Gaussianity argument applies to this structure whenever its mixed residual Gram matrices have deterministic limits. The Gaussian weight calculation above supplies the corresponding self-consistent training equations.

Normalization and shared parameters must respect the same conditions before that conclusion is used. A residual normalization $\mathcal{N}$ preserves the displayed symmetry if $\mathcal{N}(\boldsymbol{R}\boldsymbol{h}) = \boldsymbol{R}\mathcal{N}(\boldsymbol{h})$ for every orthogonal $\boldsymbol{R}$. RMS normalization without coordinatewise affine parameters, or with a scalar gain, has this property. Mean-subtracting LayerNorm and coordinatewise gains do not have full residual rotation equivariance. Tying input and output weights may still respect the rotation, but changes their joint initialization and training updates; it is not covered by the independent-endpoint convention. Growing token lengths or head dimensions also require a separate scaling analysis.

For spectral optimizers, the matrix on which the transform acts is an additional part of the specification. Packing all heads into a layer matrix gives projection dimensions proportional to $N_{\sharp\mathrm{A}}$ and $N_\flat$, consistent with the width-dependent spectral-sampling comparison in the MLP analysis. Applying the transform separately to each fixed-dimensional head preserves residual rotation equivariance, but does not enlarge that local spectral window when $N_{\sharp\mathrm{A}}$ grows. The MLP spectral-error estimate therefore cannot be transferred to per-head updates merely by replacing $N_\sharp$ with $N_{\sharp\mathrm{A}}$.

B.6 Spectral updates mix feature histories

For the spectral-clipping model of Muon, the explicit SGD learning histories in system (60) are replaced by feature-history contractions mixed by a matrix determined by their Gram matrices. Momentum enlarges these histories without changing the construction. The initial-weight integral and its source and reaction prefactors retain their form, with kernels and responses evaluated under the resulting Muon law.

A spectral update acts within the feature directions present in its gradient. For a fixed batch and finitely many training steps, collect these directions in two history matrices. Write

$$\begin{aligned} &D = s\overline{A}C\overline{B}^\mathsf{T}, \quad \overline{A} = A/\sqrt{p}, \qquad \overline{B} = B/\sqrt{q}, \\ &P = A^\mathsf{T}A/p, \quad Q = B^\mathsf{T}B/q, \quad T = P^{1/2}CQ^{1/2}, \end{aligned} \tag{80}$$

where $s > 0$, $A \in \mathbb{R}^{p\times r}$, $B \in \mathbb{R}^{q\times r}$ and $C \in \mathbb{R}^{r\times r}$. The normalized clipping map in (15) satisfies the exact identity

$$F_\lambda(D) = \overline{A}\mathcal{M}\overline{B}^\mathsf{T}, \qquad \mathcal{M} = P^{\dagger/2}F_{\lambda/s}(T)Q^{\dagger/2}. \tag{81}$$

Here $P^{\dagger/2}$ is the positive square root of the Moore–Penrose inverse. The formula includes singular Gram matrices. For an up-projection gradient, the columns of A, B are respectively $\boldsymbol{g}_q^{\ell\sharp}$ and $\boldsymbol{h}^{\ell\flat}$,

indexed by batch examples and output components. For a down-projection they are $\boldsymbol{g}_q^{(\ell+1)\flat}$ and $\phi(\boldsymbol{h}^{\ell\sharp})$. The diagonal entries of C are loss derivatives divided by batch size. Equation (22) gives, for both projections,

$$s = \frac{\beta_0\sqrt{N_\sharp}}{\gamma_0 K}, \qquad \lambda = \lambda_0\sqrt{N_\sharp}/K, \qquad \frac{\lambda}{s} = \frac{\lambda_0\gamma_0}{\beta_0}. \tag{82}$$

Thus the central spectral map has no remaining size factor. Linear momentum also has this representation: concatenating the feature columns from steps $0,\dots,t$ and using coefficient blocks $(1-\mu)\mu^{t-j}C_j$ gives $M_t = \mu M_{t-1} + (1-\mu)D_t$, with $M_{-1} = 0$. No independence of the trained histories is used in these identities.

On a fixed nonzero spectral support with eigenvalues bounded away from zero, concentration of the history Gram matrices and loss coefficients determines the limit of the mixing matrix $\mathcal{M}$. The finite identity itself does not require this restriction.

For bulk rates $\eta_M = \rho\sqrt{N_\sharp}$, the accumulated forward and backward histories take the following form. At update j, label the history columns by $z_a = (\boldsymbol{x}_a, t_a)$ and their output components by p_a; momentum includes columns from all steps through j. Suppressing the layer index only on the mixing matrices, define

$$\begin{aligned}
\boldsymbol{\mathcal{T}}_\uparrow(z) &= \sum_{j<t}\sum_{a,b} \boldsymbol{g}_{p_a}^{\ell\sharp}(z_a)(\mathcal{M}_{\uparrow,j})_{ab}\Phi^{\ell\flat}(z,z_b),\\
\boldsymbol{\mathcal{T}}_\downarrow(z) &= \sum_{j<t}\sum_{a,b} \boldsymbol{g}_{p_a}^{(\ell+1)\flat}(z_a)(\mathcal{M}_{\downarrow,j})_{ab}\Phi^{\ell\sharp}(z,z_b),\\
\boldsymbol{\mathcal{V}}_{\downarrow,q}(z) &= \sum_{j<t}\sum_{a,b} \phi(\boldsymbol{h}^{\ell\sharp}(z_b))(\mathcal{M}_{\downarrow,j})_{ab}G^{(\ell+1)\flat}_{qp_a}(z,z_a),\\
\boldsymbol{\mathcal{V}}_{\uparrow,q}(z) &= \sum_{j<t}\sum_{a,b} \boldsymbol{h}^{\ell\flat}(z_b)(\mathcal{M}_{\uparrow,j})_{ab}G^{\ell\sharp}_{qp_a}(z,z_a).
\end{aligned} \tag{83}$$

The kernels here are the empirical inner products, and all fields are evaluated in the trained network. With the initial-weight projections already defined, the four exact identities are

$$\begin{aligned}
\boldsymbol{h}^{\ell\sharp} &= \boldsymbol{\chi}^{\ell\sharp} - \rho_\uparrow\boldsymbol{\mathcal{T}}_\uparrow,\\
\boldsymbol{h}^{(\ell+1)\flat} - \boldsymbol{h}^{\ell\flat} &= \boldsymbol{\chi}^{(\ell+1)\flat} - \frac{\beta_0\rho_\downarrow}{L}\boldsymbol{\mathcal{T}}_\downarrow,\\
\boldsymbol{g}_q^{\ell\sharp} &= \phi'(\boldsymbol{h}^{\ell\sharp})\odot\left(\boldsymbol{\xi}_q^{\ell\sharp} - \rho_\downarrow\boldsymbol{\mathcal{V}}_{\downarrow,q}\right),\\
\boldsymbol{g}_q^{\ell\flat} - \boldsymbol{g}_q^{(\ell+1)\flat} &= \boldsymbol{\xi}_q^{\ell\flat} - \frac{\beta_0\rho_\uparrow}{L}\boldsymbol{\mathcal{V}}_{\uparrow,q}.
\end{aligned} \tag{84}$$

The minus signs reflect the positive loss gradients in the spectral factorization; $\mathcal{S}_t$ in the SGD formulas instead includes negative loss feedback.

The initial-weight integral (36) is unchanged: only the learning histories attached to its projection constraints differ. It gives the source covariances (55) and reaction prefactors (54), with kernels and responses evaluated under the Muon law. In (57)–(58), replace the four explicit SGD training terms by the corresponding contractions in (84), replacing empirical inner products and coordinate fields by their population versions. The responses differentiate this full computation, including the spectral update, while macroscopic kernels are fixed. Continuous depth replaces the residual $1/L$ terms by $d\tau$. For a nonlinear spectral map, its mixing coefficients depend on both hidden and residual history Gram matrices; the growing hidden population is therefore essential to this construction.

Trainable endpoints. The same construction replaces the two boundary histories in (60g). Let D_s^j be the positive loss-gradient matrix of the input or output map, obtained from (22) by the batch loss contraction, and $M_s^j = \mu M_{s-1}^j + (1-\mu)D_s^j$, with $M_{-1}^j = 0$. Using $\eta_{M,j} = \rho_j\sqrt{N_\flat}$ and $\lambda_j = \lambda_{0,j}/\sqrt{N_\flat}$ gives

$$\begin{aligned}
\boldsymbol{h}^{0\flat}(\boldsymbol{x},t) &= \frac{\boldsymbol{W}^0(0)\boldsymbol{x}}{\sqrt{D_{\mathrm{i}}}} - \frac{\eta_{M,\mathrm{in}}}{\sqrt{D_{\mathrm{i}}}}\sum_{s<t} F_{\lambda_{\mathrm{in}}}(M_s^{\mathrm{in}})\boldsymbol{x},\\
\boldsymbol{g}_q^{L\flat}(t) &= (\boldsymbol{W}^L(0))^{\mathsf{T}}\boldsymbol{e}_q - \eta_{M,\mathrm{out}}\sum_{s<t} F_{\lambda_{\mathrm{out}}}(M_s^{\mathrm{out}})^{\mathsf{T}}\boldsymbol{e}_q.
\end{aligned} \tag{85}$$

Here $\boldsymbol{e}_q$ is the qth output basis vector; $\mu = 0$ removes momentum. Applying (81) to these rectangular maps and replacing residual history Gram matrices by their population values gives the boundary laws. Task-coordinate sums remain finite. The readout identity in (60g) is unchanged.

Factorization and update scales. Indeed, polar decomposition gives $\overline{A} = J_A P^{1/2}$ and $\overline{B} = J_B Q^{1/2}$, with partial isometries J_A, J_B. A singular-value decomposition on the supports of P, Q gives $F_\lambda(D) = J_A F_{\lambda/s}(T) J_B^\mathsf{T}$. Substituting $J_A = \overline{A} P^{\dagger/2}$ and $J_B = \overline{B} Q^{\dagger/2}$ proves (81). In the linear spectral region the product reduces to D/λ.

For bulk Muon rates $\eta_M = \rho\sqrt{N_\sharp}$, put $k_B(v) = B^\mathsf{T} v/q$, using the input dimension q of the relevant projection. Holding the incoming field fixed gives

$$\begin{aligned} \Delta \boldsymbol{h}^{\ell\sharp}\big|_{\mathrm{up}} &= -\rho_\uparrow A_\uparrow \mathcal{M}_\uparrow k_{B_\uparrow}(\boldsymbol{h}^{\ell\flat}), \\ \Delta \boldsymbol{h}^{(\ell+1)\flat}\Big|_{\mathrm{down}} &= -\frac{\rho_\downarrow \beta_0}{L} A_\downarrow \mathcal{M}_\downarrow k_{B_\downarrow}(\phi(\boldsymbol{h}^{\ell\sharp})). \end{aligned} \tag{86}$$

B.7 Adam retains coordinate-pair moment histories

Adam replaces the explicit SGD learning histories in system (60) by population averages of normalized moment updates for residual–hidden coordinate pairs. The moment state is computed from each pair's gradient history before averaging over the source population. These contractions depend on the full history law, rather than only its second-moment kernels, and can be nonlinear in the remaining residual history.

Adam normalizes gradient entries separately. Write a parameter group's gradient as $D_t^j = d_j R_t^j$, with gradient-entry scale $d_j > 0$, and use a stabilizer $\epsilon_j = d_j \epsilon_0$, with $\epsilon_0 > 0$. The normalized moment states, initially zero, obey

$$\begin{aligned} M_t^j &= a_1 M_{t-1}^j + (1-a_1) R_t^j, & V_t^j &= a_2 V_{t-1}^j + (1-a_2)(R_t^j)^{\odot 2}, \\ \widehat{M}_t^j &= \frac{M_t^j}{1-a_1^{t+1}}, & \widehat{V}_t^j &= \frac{V_t^j}{1-a_2^{t+1}}, \\ \mathcal{U}_t^j &= \frac{\widehat{M}_t^j}{\sqrt{\widehat{V}_t^j} + \epsilon_0}, & W^j(t+1) &= W^j(t) - \eta_j^{\mathrm{W}} \mathcal{U}_t^j. \end{aligned} \tag{87}$$

The square, square root and division are entrywise. This is an exact change of gradient units, with fixed moment coefficients $a_1, a_2 \in [0, 1)$. For the bulk projections, $d_\uparrow = d_\downarrow = \beta_0/(\gamma_0 K\sqrt{N_\flat})$, and

$$\begin{aligned} (R_t^\uparrow)_{ai} &= \frac{1}{B_t} \sum_{(\boldsymbol{x},y)\in\mathcal{B}_t} \sum_q \partial_{f_q} \mathcal{L}_{\mathrm{ex}}\, g_{q,a}^{\ell\sharp}(\boldsymbol{x},t) h_i^{\ell\flat}(\boldsymbol{x},t), \\ (R_t^\downarrow)_{ia} &= \frac{1}{B_t} \sum_{(\boldsymbol{x},y)\in\mathcal{B}_t} \sum_q \partial_{f_q} \mathcal{L}_{\mathrm{ex}}\, g_{q,i}^{(\ell+1)\flat}(\boldsymbol{x},t) \phi(h_a^{\ell\sharp}(\boldsymbol{x},t)). \end{aligned} \tag{88}$$

The bulk rates are $\eta_j^{\mathrm{W}} = \rho_j/\sqrt{N_\flat}$.

The population variables for Adam include the moment state of a residual–hidden coordinate pair. For an up map, its normalized gradient is the batch average of $\boldsymbol{g}_q^{\ell\sharp} \boldsymbol{h}^{\ell\flat}$; for a down map it is the batch average of $\boldsymbol{g}_q^{(\ell+1)\flat} \phi(\boldsymbol{h}^{\ell\sharp})$, with the same loss derivatives as in (88). Applying (87) to these pair histories defines $\mathcal{U}_s^\uparrow(\sharp,\flat)$ and $\mathcal{U}_s^\downarrow(\flat,\sharp)$. Let a tilde denote a source-population cavity representative.

The population row and column averages are

$$\begin{aligned}
\mathcal{A}_{\uparrow}(z) &= \sum_{s<t} \mathbb{E}_{\widetilde{\flat}}[\mathcal{U}_s^{\uparrow}(\sharp,\widetilde{\flat})\widetilde{\boldsymbol{\hbar}}^{\ell\flat}(z)],\\
\mathcal{A}_{\downarrow}(z) &= \sum_{s<t} \mathbb{E}_{\widetilde{\sharp}}[\mathcal{U}_s^{\downarrow}(\flat,\widetilde{\sharp})\phi(\widetilde{\boldsymbol{\hbar}}^{\ell\sharp}(z))],\\
\mathcal{B}_{\downarrow,q}(z) &= \sum_{s<t} \mathbb{E}_{\widetilde{\flat}}[\mathcal{U}_s^{\downarrow}(\widetilde{\flat},\sharp)\widetilde{\boldsymbol{g}}_q^{(\ell+1)\flat}(z)],\\
\mathcal{B}_{\uparrow,q}(z) &= \sum_{s<t} \mathbb{E}_{\widetilde{\sharp}}[\mathcal{U}_s^{\uparrow}(\widetilde{\sharp},\flat)\widetilde{\boldsymbol{g}}_q^{\ell\sharp}(z)].
\end{aligned} \tag{89}$$

Each average holds the target-coordinate history fixed and averages the source histories with their cavity law. It is the limit of the corresponding empirical row or column average, rather than an independence assertion about trained coordinates. The training terms for $\boldsymbol{\hbar}^{\ell\sharp}$, $\boldsymbol{\hbar}^{(\ell+1)\flat} - \boldsymbol{\hbar}^{\ell\flat}$, the bracket multiplying $\phi'(\boldsymbol{\hbar}^{\ell\sharp})$ in $\boldsymbol{g}_q^{\ell\sharp}$, and $\boldsymbol{g}_q^{\ell\flat} - \boldsymbol{g}_q^{(\ell+1)\flat}$ become, respectively,

$$-\rho_{\uparrow}\mathcal{A}_{\uparrow}, \qquad -\frac{\beta_0\rho_{\downarrow}}{L}\mathcal{A}_{\downarrow}, \qquad -\rho_{\downarrow}\mathcal{B}_{\downarrow,q}, \qquad -\frac{\beta_0\rho_{\uparrow}}{L}\mathcal{B}_{\uparrow,q}. \tag{90}$$

The Gaussian integral over the initial weights has the same form as (36), with the same source and reaction prefactors. The optimizer dependence resides in the paired moment histories: their population contractions depend on the full representative-history law, not just its four second-moment kernels. Kernels and responses are evaluated self-consistently from this augmented Adam law. In continuous depth the residual factors $1/L$ become $d\tau$. Unlike the spectral history contractions, these pair averages are generally nonlinear functions of the remaining residual history. Gaussian initial sources therefore need not give Gaussian residual statistics after Adam updates.

Trainable endpoints. For the input and output maps, use the same moment recursion with $d_{\mathrm{in}} = 1/(\gamma_0 N_{\flat}\sqrt{D_{\mathrm{i}}})$, $d_{\mathrm{out}} = 1/(\gamma_0 N_{\flat})$, and $\eta_j^{\mathrm{W}} = \rho_j$. Their normalized gradients are

$$\begin{aligned}
(R_t^{\mathrm{in}})_{ik} &= \frac{1}{B_t}\sum_{(\boldsymbol{x},y)\in\mathcal{B}_t}\sum_{q}\partial_{f_q}\mathcal{L}_{\mathrm{ex}}\, g_{q,i}^{0\flat}(\boldsymbol{x},t)x_k,\\
(R_t^{\mathrm{out}})_{qi} &= \frac{1}{B_t}\sum_{(\boldsymbol{x},y)\in\mathcal{B}_t}\partial_{f_q}\mathcal{L}_{\mathrm{ex}}\, h_i^{L\flat}(\boldsymbol{x},t).
\end{aligned} \tag{91}$$

The resulting boundary histories are

$$\begin{aligned}
\boldsymbol{h}^{0\flat}(\boldsymbol{x},t) &= \frac{\boldsymbol{W}^0(0)\boldsymbol{x}}{\sqrt{D_{\mathrm{i}}}} - \frac{\rho_{\mathrm{in}}}{\sqrt{D_{\mathrm{i}}}}\sum_{s<t}\mathcal{U}_s^{\mathrm{in}}\boldsymbol{x},\\
\boldsymbol{g}_q^{L\flat}(t) &= (\boldsymbol{W}^L(0))^{\mathsf{T}}\boldsymbol{e}_q - \rho_{\mathrm{out}}\sum_{s<t}(\mathcal{U}_s^{\mathrm{out}})^{\mathsf{T}}\boldsymbol{e}_q.
\end{aligned} \tag{92}$$

For a representative residual coordinate, the input row and output column retain their finitely many task-coordinate moment states. These histories replace the boundary learning terms in (60g); its readout identity is unchanged.

Finite-network histories. The coefficients in (90) follow directly from the finite matrix updates. Accumulating the update matrices gives the exact forward histories

$$\begin{aligned}
\boldsymbol{h}^{\ell\sharp}(z) &= \frac{\boldsymbol{W}^{\ell\uparrow}(0)\boldsymbol{h}^{\ell\flat}(z)}{\sqrt{N_{\flat}}} - \frac{\rho_{\uparrow}}{N_{\flat}}\sum_{s<t}\mathcal{U}_s^{\ell\uparrow}\boldsymbol{h}^{\ell\flat}(z),\\
\boldsymbol{h}^{(\ell+1)\flat}(z) &= \boldsymbol{h}^{\ell\flat}(z) + \frac{\beta_0\sqrt{N_{\flat}}}{K}\boldsymbol{W}^{\ell\downarrow}(0)\phi(\boldsymbol{h}^{\ell\sharp}(z))\\
&\quad - \frac{\beta_0\rho_{\downarrow}}{K}\sum_{s<t}\mathcal{U}_s^{\ell\downarrow}\phi(\boldsymbol{h}^{\ell\sharp}(z)).
\end{aligned} \tag{93}$$

All query fields are evaluated in the trained network at $z = (\boldsymbol{x}, t)$. The backward histories follow from the transposed update matrices:

$$\begin{aligned} \boldsymbol{g}_q^{\ell\sharp}(z) &= \phi'(\boldsymbol{h}^{\ell\sharp}(z)) \odot \left[\boldsymbol{\xi}_q^{\ell\sharp}(z) - \frac{\rho_\downarrow}{N_\flat} \sum_{s<t} (\mathcal{U}_s^{\ell\downarrow})^\mathsf{T} \boldsymbol{g}_q^{(\ell+1)\flat}(z) \right], \\ \boldsymbol{g}_q^{\ell\flat}(z) - \boldsymbol{g}_q^{(\ell+1)\flat}(z) &= \boldsymbol{\xi}_q^{\ell\flat}(z) - \frac{\beta_0 \rho_\uparrow}{K} \sum_{s<t} (\mathcal{U}_s^{\ell\uparrow})^\mathsf{T} \boldsymbol{g}_q^{\ell\sharp}(z). \end{aligned} \tag{94}$$

B.8 Joint Gaussianity of residual histories

SGD and orthogonally equivariant spectral updates preserve a common rotation of the entire residual history. When mixed inner products concentrate at deterministic limits, any fixed finite collection of forward residual states and output Jacobians has a joint Gaussian limit across inputs, depths, output components, and training steps, even when hidden variables are non-Gaussian. Adam can break this property after one update while retaining self-averaging statistics. The population equations determine the mixed inner products; the rotation argument below identifies the joint law. Numerical illustrations then examine these distributions after learning in a nonlinear logic task and a complete GPT-2-style training run.

B.8.1 One rotation acts on the entire training history

Fix an orthogonal matrix $R \in O(N_\flat)$. Apply the same transformation to every layer:

$$\begin{aligned} \boldsymbol{W}^0 &\mapsto R\boldsymbol{W}^0, \quad \boldsymbol{W}^{\ell\uparrow} \mapsto \boldsymbol{W}^{\ell\uparrow} R^\mathsf{T}, \\ \boldsymbol{W}^{\ell\downarrow} &\mapsto R\boldsymbol{W}^{\ell\downarrow}, \quad \boldsymbol{W}^{L} \mapsto \boldsymbol{W}^{L} R^\mathsf{T}. \end{aligned} \tag{95}$$

For the MLP residual network, all hidden activations and outputs remain unchanged, while $\boldsymbol{h}^{\ell\flat} \mapsto R\boldsymbol{h}^{\ell\flat}$ and $\boldsymbol{g}_q^{\ell\flat} \mapsto R\boldsymbol{g}_q^{\ell\flat}$. In particular, an arbitrary hidden nonlinearity does not obstruct this symmetry: its input is unchanged. The identity skip connections require the same R at every depth.

Fix the data and batch sequence, independently of the residual coordinate system, and let the loss depend on the network through its output. The induced parameter transformation is orthogonal in the Frobenius inner product. Invariance of the loss therefore implies equivariance of its gradients. SGD with a scalar rate for each parameter group preserves (95), even when the rates differ between groups or change with training step.

The singular-value definition of F_λ gives

$$F_\lambda(QDP^\mathsf{T}) = QF_\lambda(D)P^\mathsf{T} \tag{96}$$

for compatible orthogonal Q, P. Thus the spectral update preserves the same residual rotation. The same identity holds for odd spectral polynomials and their finite compositions. Linear momentum also preserves it when the momentum state is transformed with the gradient. A threshold or scalar multiplier chosen from rotation-invariant quantities preserves equivariance. None of these statements requires the update to be linear or homogeneous as a function of the gradient.

Independent centered Gaussian weight matrices, with a common variance within each group, have an initialization law invariant under (95). Induction over training steps consequently gives a common rotational invariance of the whole residual history. The result applies when all four parameter groups use compatible updates; a hybrid method that applies coordinatewise Adam to the input or readout does not follow from this argument.

Residual-coordinate operations, such as learned diagonal normalization gains, require a separate symmetry check.

B.8.2 From deterministic mixed inner products to a joint Gaussian law

Collect r residual vectors from any fixed finite set of the labels above into an $N_\flat \times r$ matrix $Y_{N_\flat}$, and define

$$\Gamma_{N_\flat} = \frac{1}{N_\flat} Y_{N_\flat}^\mathsf{T} Y_{N_\flat}. \tag{97}$$

Suppose that $\Gamma_{N_\flat}$ converges in probability to a deterministic, finite, positive semidefinite matrix Γ. Then, for every fixed k, the first k rows of $Y_{N_\flat}$ converge jointly to k independent $\mathcal{N}(0,\Gamma)$ vectors.

To see this, conditional on $\Gamma_{N_\flat}$, rotational invariance makes $Y_{N_\flat}$ uniform on the orthogonal orbit with that Gram matrix. Let $Z_{N_\flat}$ be an independent $N_\flat \times r$ standard Gaussian matrix. For $N_\flat \geq r$, a representation of this conditional law is

$$Y_{N_\flat} \overset{d}{=} Z_{N_\flat}(Z_{N_\flat}^\mathsf{T} Z_{N_\flat})^{-1/2}(N_\flat \Gamma_{N_\flat})^{1/2} = Z_{N_\flat}\left(\frac{Z_{N_\flat}^\mathsf{T} Z_{N_\flat}}{N_\flat}\right)^{-1/2} \Gamma_{N_\flat}^{1/2}. \tag{98}$$

Only $Z_{N_\flat}^\mathsf{T} Z_{N_\flat}$ is inverted, so $\Gamma_{N_\flat}$ may be singular. The middle factor converges in probability to I_r, and $\Gamma_{N_\flat}^{1/2} \to \Gamma^{1/2}$. The first k rows of $Z_{N_\flat}$ are independent standard Gaussian vectors. Slutsky's theorem proves the stated joint limit.

The columns of $Y_{N_\flat}$ include the complete finite selection of histories from the beginning. Thus the proof establishes Gaussianity jointly across training steps, inputs, depth, and forward/backward variables. It does not add separately Gaussian terms and infer that their sum is Gaussian. Different entries within a row can be arbitrarily correlated; their correlations are the off-diagonal entries of Γ. The Gaussian matrix in (98) is a distributional representation, not noise injected during training.

A random limiting Gram matrix would instead give a Gaussian mixture. The deterministic-Gram condition is therefore substantive. The argument identifies finite-dimensional distributions; process convergence and growing training durations additionally require tightness and uniform estimates.

B.8.3 Adam can destroy residual Gaussianity in one step

Consider the zero-block network with scalar input 1, scalar target 1, and two trainable linear maps:

$$f_{N_\flat} = \frac{1}{N_\flat}\sum_{i=1}^{N_\flat} a_i w_i, \qquad \mathcal{L} = \frac{1}{2}(f_{N_\flat} - 1)^2, \qquad a_i(0), w_i(0) \overset{\text{iid}}{\sim} \mathcal{N}(0,1). \tag{99}$$

In the standard-Gaussian coordinates used here, $h_i^\flat = w_i$ and the normalized output gradient is $g_{1,i}^\flat = a_i$. Adam starts from zero moment estimates, uses bias correction, learning rate $\eta > 0$, and stabilizer $\epsilon_{N_\flat} = e/N_\flat$ with fixed $e \geq 0$. Regardless of its fixed momentum coefficients in $[0,1)$, the first update is the gradient divided by its absolute value plus $\epsilon_{N_\flat}$. Since $f_{N_\flat}(0) \to 0$, every fixed set of coordinates has the limit

$$\begin{aligned} w_i(1) &= w_i(0) + \eta\frac{a_i(0)}{|a_i(0)| + e}, \\ a_i(1) &= a_i(0) + \eta\frac{w_i(0)}{|w_i(0)| + e}. \end{aligned} \tag{100}$$

The common factor $f_{N_\flat}(0) - 1$ is included in the finite update before taking this limit; it is not replaced separately at every coordinate.

When $e = 0$, either marginal in (100) is

$$\tfrac{1}{2}\mathcal{N}(\eta, 1) + \tfrac{1}{2}\mathcal{N}(-\eta, 1). \tag{101}$$

Its variance is $1 + \eta^2$ and its fourth cumulant is $-2\eta^4$. For $e > 0$, the added term is an independent, bounded, nonconstant random variable. If its sum with the independent standard Gaussian were Gaussian, division of their characteristic functions would make this added term Gaussian as well, which is impossible. Thus a positive stabilizer does not restore Gaussianity in this scaling. The residual marginal itself fails to be Gaussian, so joint Gaussianity fails too.

This example still has self-averaging population statistics. At $e = 0$, for instance,

$$\frac{1}{N_\flat}\sum_i w_i(1)^2 \longrightarrow 1 + \eta^2, \qquad f_{N_\flat}(1) \longrightarrow 2\eta\sqrt{\frac{2}{\pi}}. \tag{102}$$

The common initial output tends to zero and the limiting coordinate pairs are functions of independent initial pairs. Non-Gaussianity here is not caused by a random limiting covariance.

To retain this update at large width, ϵ must follow the gradient scale. A fixed positive standard-coordinate ϵ would make the first update vanish. In code coordinates, the conversion rules give rate and stabilizer $\eta/N_\flat$ and e for the output code weight $a_i/N_\flat$; for the input code weight w_i, they remain η and $e/N_\flat$.

B.8.4 Numerical illustrations after learning

After learning AND, OR, and XOR, the nonlinear logic network displays approximately Gaussian joint clouds of residual activations and actual output Jacobians under SGD and the calibrated Taylor spectral update, whereas Adam produces a visibly non-Gaussian distribution (Fig. 10). The Jacobians include both the nonlinear branch and the skip path. Protocols and quantitative diagnostics are given in Appendix F.1.

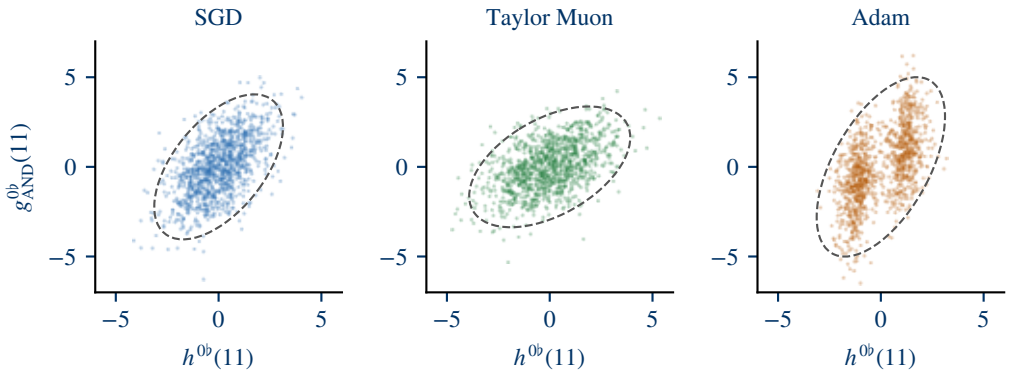


Figure 10: Residual activation–Jacobian pairs after learning the nonlinear logic task. Each axis is centered, with its original scale retained. Ellipses enclose 95% of the fitted Gaussian reference; no whitening is applied.

Approximately Gaussian residual activations and output Jacobians also persist through a complete GPT-2-style run using practical NS5 Muon on every matrix (Fig. 11). At an intermediate checkpoint, these residual shapes coexist with strongly non-Gaussian first-block pre-GELU activations (Fig. 3, right). Training and measurement details are given in Appendix I.3.

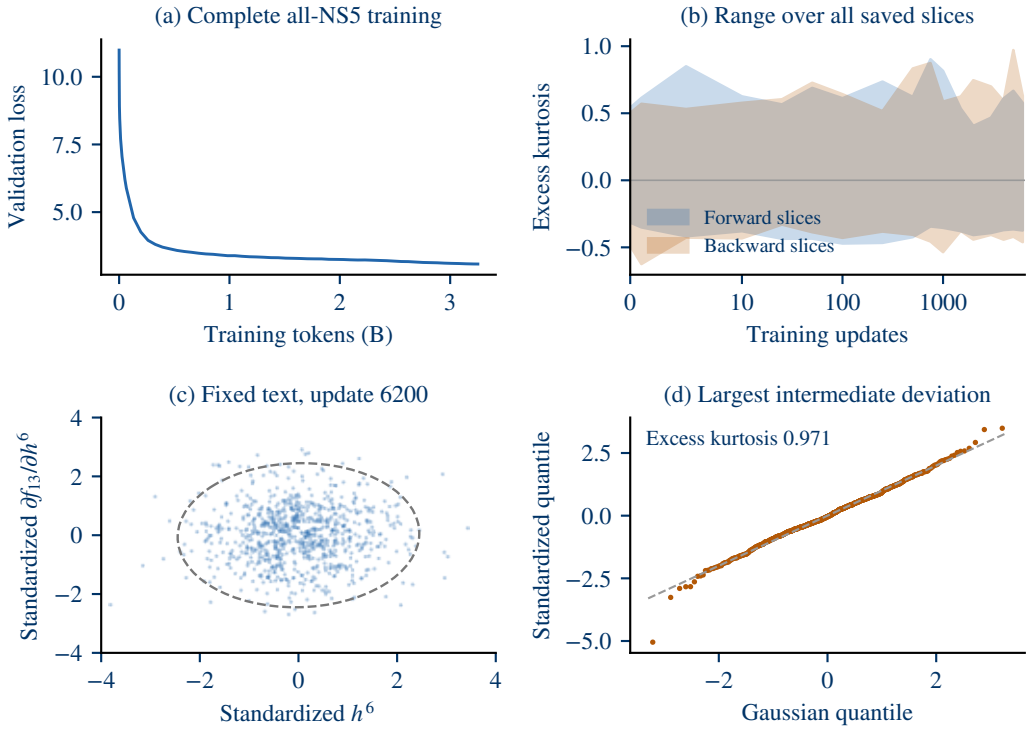


Figure 11: All-NS5 training and distributional diagnostics. Top: complete validation trajectory and excess-kurtosis ranges over the predefined slices at each measurement time; the bands are ranges, not confidence intervals. Bottom: the final forward/backward pair for the first predefined text, with a fitted Gaussian 95% ellipse, and quantiles for the retained backward slice with the largest intermediate excess kurtosis. Each plotted vector is standardized separately, without whitening.

C Zero- and one-step checkpoints

C.1 Initialization and the intrinsic ratio

Before training, the current block's matrices are independent of its incoming residual states. For a finite input collection, let $C_\ell(\boldsymbol{x},\boldsymbol{x}') = N_\flat^{-1}(\boldsymbol{h}^{\ell\flat}(\boldsymbol{x},0))^\top \boldsymbol{h}^{\ell\flat}(\boldsymbol{x}',0)$. Each up-projection row produces a Gaussian input vector of covariance C_ℓ. Define the nonlinearity's second-moment map by

$$[\mathcal{V}_\phi(C)]_{\boldsymbol{x},\boldsymbol{x}'} = \mathbb{E}[\phi(Z_{\boldsymbol{x}})\phi(Z_{\boldsymbol{x}'})], \qquad Z \sim \mathcal{N}(0,C). \tag{103}$$

If $\delta h_i^{\ell\flat}(\boldsymbol{x}) = h_i^{(\ell+1)\flat}(\boldsymbol{x},0) - h_i^{\ell\flat}(\boldsymbol{x},0)$, averaging over the independent current up- and down-projections gives

$$\begin{aligned}
\mathbb{E}[\delta h_i^{\ell\flat}(\boldsymbol{x}) \mid \boldsymbol{h}^{\ell\flat}] &= 0,\\
\mathbb{E}[\delta h_i^{\ell\flat}(\boldsymbol{x})\delta h_j^{\ell\flat}(\boldsymbol{x}') \mid \boldsymbol{h}^{\ell\flat}] &= \delta_{ij}\frac{\beta_0^2(N_\flat/K)}{L}[\mathcal{V}_\phi(C_\ell)]_{\boldsymbol{x},\boldsymbol{x}'},\\
\mathbb{E}[C_{\ell+1} \mid \boldsymbol{h}^{\ell\flat}] &= C_\ell + \frac{\beta_0^2(N_\flat/K)}{L}\mathcal{V}_\phi(C_\ell).
\end{aligned} \tag{104}$$

These finite-size identities retain the dependence of each block's conditional variance on its incoming state.

When empirical covariances concentrate, moments remain controlled, and the depth sums converge, the initialization covariance satisfies

$$\frac{dC(\tau)}{d\tau} = \beta_0^2\alpha\,\mathcal{V}_\phi(C(\tau)), \qquad C(0)_{\boldsymbol{x},\boldsymbol{x}'} = \frac{\boldsymbol{x}^\top\boldsymbol{x}'}{D_{\mathrm{i}}}. \tag{105}$$

The limiting residual process is centered Gaussian with independent increments over disjoint depth intervals. Its initial value is independent of these increments, and its covariance across both depth and inputs is

$$\begin{aligned}\mathbb{E}\Big[\boldsymbol{\hbar}^{\flat}(\tau;\boldsymbol{x},0)\boldsymbol{\hbar}^{\flat}(s;\boldsymbol{x}',0)\Big] &= C(\min\{\tau,s\})_{\boldsymbol{x},\boldsymbol{x}'},\\ \mathbb{E}[du(\tau;\boldsymbol{x},0)\,du(\tau;\boldsymbol{x}',0)] &= c\,[\mathcal{V}_\phi(C(\tau))]_{\boldsymbol{x},\boldsymbol{x}'}\,d\tau, \qquad c=\beta_0^2\alpha.\end{aligned} \tag{106}$$

At $t=0$, the learning histories and forward response kernels $A^{\flat}, A^{\sharp}$ vanish: no parameter update has yet coupled the forward fields to backward sources. Equation (60c) therefore reduces to $d\boldsymbol{\hbar}^{\flat}=du$, with the covariance above. Thus the deterministic covariance ODE describes the law of a random residual trajectory, not a deterministic trajectory.

For a fixed input, the diagonal $C_h(\tau;0)=C(\tau)_{\boldsymbol{x},\boldsymbol{x}}$ satisfies $\partial_\tau C_h = c\,\mathbb{E}[\phi(\sqrt{C_h}Z)^2]$, where $Z\sim\mathcal{N}(0,1)$. Unit pre-RMSNorm instead normalizes the branch input before the up-projection, leaving the residual stream itself unnormalized. With no gain and with the normalization epsilon set to zero, it replaces $C(\tau)$ in $\mathcal{V}_\phi$ by the correlation matrix $R(\tau)_{\boldsymbol{x},\boldsymbol{x}'}=C(\tau)_{\boldsymbol{x},\boldsymbol{x}'}/\sqrt{C(\tau)_{\boldsymbol{x},\boldsymbol{x}}C(\tau)_{\boldsymbol{x}',\boldsymbol{x}'}}$. Consequently $\partial_\tau C_h = c\,\mathbb{E}[\phi(Z)^2]$ is constant. With a nonzero normalization epsilon ε, the diagonal intensity is instead $c\,\mathbb{E}[\phi(\sqrt{C_h/(C_h+\varepsilon)}Z)^2]$. For ReLU, the zero-epsilon solutions are

$$C_h(\tau;0)=\begin{cases}C_h(0;0)e^{c\tau/2}, & \text{without RMSNorm},\\ C_h(0;0)+c\tau/2, & \text{with unit pre-RMSNorm}.\end{cases} \tag{107}$$

For $\phi(z)=z$, the factor $1/2$ is replaced by 1 in both solutions; in particular a normalized input without RMSNorm gives $C_h(\tau;0)=e^{c\tau}$. At fixed β_0, the random depth contribution disappears as $\alpha\to 0$.

C.2 Effective step and the linear checkpoint

C.2.1 Setting and parameter coordinates

As above, $K=N_\sharp L$. There is one deterministic input $\boldsymbol{x}\in\mathbb{R}^{D_{\mathrm{i}}}$ with $\|\boldsymbol{x}\|^2/D_{\mathrm{i}}=1$ and one scalar target $y=1$. At initialization,

$$\begin{aligned}\boldsymbol{h}^{0\flat}&=\boldsymbol{W}^0\boldsymbol{x}/\sqrt{D_{\mathrm{i}}}, \quad \boldsymbol{A}_\ell = I+\frac{\beta_0}{K}\boldsymbol{W}^{\ell\downarrow}\boldsymbol{W}^{\ell\uparrow},\\ \boldsymbol{h}^{(\ell+1)\flat}&=\boldsymbol{A}_\ell\boldsymbol{h}^{\ell\flat}, \qquad\qquad f_0=\frac{\boldsymbol{W}^L\boldsymbol{h}^{L\flat}}{\gamma_0 N_\flat}.\end{aligned} \tag{108}$$

Here $\boldsymbol{W}^{\ell\uparrow}\in\mathbb{R}^{N_\sharp\times N_\flat}$, $\boldsymbol{W}^{\ell\downarrow}\in\mathbb{R}^{N_\flat\times N_\sharp}$, and $\boldsymbol{W}^L\in\mathbb{R}^{1\times N_\flat}$. All entries are independent standard Gaussians. In particular, $\boldsymbol{h}^{0\flat}$ is a standard Gaussian vector independent of the residual weights and $(\boldsymbol{W}^L)^{\mathsf{T}}$. The input and readout remain frozen. All gradients below are evaluated at this same initial network.

For fixed $\beta_0,\gamma_0>0$, write the SGD rate of each residual projection in standard-Gaussian coordinates as

$$\eta_{\mathrm{W}}=\eta_0\gamma_0^2K=\frac{\eta_{\mathrm{eff}}\gamma_0 K}{\beta_0}, \qquad \eta_{\mathrm{eff}}=\eta_0\gamma_0\beta_0. \tag{109}$$

The coefficient η_{eff} controls the direct local hidden-feature update. Both residual projections are updated simultaneously below, so their joint effect and its propagation through the network are retained. If the scale factors are instead absorbed into the stored matrices,

$$\mathtt{Wup}^\ell=\boldsymbol{W}^{\ell\uparrow}/\sqrt{N_\flat}, \quad \mathtt{Wdown}^\ell=\beta_0\sqrt{N_\flat}\,\boldsymbol{W}^{\ell\downarrow}/K, \tag{110}$$

the equivalent SGD rates are

$$\eta_{\mathtt{Wup}}=\eta_{\mathrm{eff}}\gamma_0 K/(\beta_0 N_\flat), \qquad \eta_{\mathtt{Wdown}}=\eta_{\mathrm{eff}}\gamma_0\beta_0 N_\flat/K. \tag{111}$$

These rates follow from the displayed up- and down-projection factors. Both endpoints remain fixed.

C.2.2 THE EXACT FINITE UPDATE

Define the backward vectors

$$\boldsymbol{g}_1^{L\flat} = (\boldsymbol{W}^L)^\mathsf{T}, \qquad \boldsymbol{g}_1^{\ell\flat} = \boldsymbol{A}_\ell^\top \boldsymbol{g}_1^{(\ell+1)\flat}, \qquad \delta = 1 - f_0. \tag{112}$$

For the loss $\mathcal{L} = \frac{1}{2}(f_0 - 1)^2$, direct differentiation gives

$$\begin{aligned} \nabla_{\boldsymbol{W}^{\ell\uparrow}} \mathcal{L} &= -\frac{\delta\beta_0}{\gamma_0 N_\flat K}((\boldsymbol{W}^{\ell\downarrow})^\top \boldsymbol{g}_1^{(\ell+1)\flat})(\boldsymbol{h}^{\ell\flat})^\top, \\ \nabla_{\boldsymbol{W}^{\ell\downarrow}} \mathcal{L} &= -\frac{\delta\beta_0}{\gamma_0 N_\flat K}\boldsymbol{g}_1^{(\ell+1)\flat}(\boldsymbol{W}^{\ell\uparrow}\boldsymbol{h}^{\ell\flat})^\top. \end{aligned} \tag{113}$$

The simultaneous parameter update is

$$\begin{aligned} \Delta\boldsymbol{W}^{\ell\uparrow} &= \frac{\eta_{\text{eff}}\delta}{N_\flat}((\boldsymbol{W}^{\ell\downarrow})^\top \boldsymbol{g}_1^{(\ell+1)\flat})(\boldsymbol{h}^{\ell\flat})^\top, \\ \Delta\boldsymbol{W}^{\ell\downarrow} &= \frac{\eta_{\text{eff}}\delta}{N_\flat}\boldsymbol{g}_1^{(\ell+1)\flat}(\boldsymbol{W}^{\ell\uparrow}\boldsymbol{h}^{\ell\flat})^\top, \\ \boldsymbol{A}_\ell^+ &= \boldsymbol{A}_\ell + \frac{\beta_0}{K}\left(\Delta\boldsymbol{W}^{\ell\downarrow}\boldsymbol{W}^{\ell\uparrow} + \boldsymbol{W}^{\ell\downarrow}\Delta\boldsymbol{W}^{\ell\uparrow} + \Delta\boldsymbol{W}^{\ell\downarrow}\Delta\boldsymbol{W}^{\ell\uparrow}\right), \\ (\boldsymbol{h}^{L\flat})^+ &= \boldsymbol{A}_{L-1}^+ \cdots \boldsymbol{A}_0^+ \boldsymbol{h}^{0\flat}. \end{aligned} \tag{114}$$

In particular, a finite SGD step includes the last, quadratic term and products of updates at different layers. Both are retained in the finite-network simulations.

For comparison, let $\boldsymbol{z}_0 = 0$ and define the linear response with the output error set to its limiting value $\delta = 1$:

$$\boldsymbol{z}_{\ell+1} = \boldsymbol{A}_\ell \boldsymbol{z}_\ell + \frac{\eta_{\text{eff}}\beta_0}{N_\flat K}\left[\|\boldsymbol{W}^{\ell\uparrow}\boldsymbol{h}^{\ell\flat}\|^2 \boldsymbol{g}_1^{(\ell+1)\flat} + \|\boldsymbol{h}^{\ell\flat}\|^2 \boldsymbol{W}^{\ell\downarrow}(\boldsymbol{W}^{\ell\downarrow})^\top \boldsymbol{g}_1^{(\ell+1)\flat}\right]. \tag{115}$$

This auxiliary response is not the definition of the finite training step. Its relation to (114) is justified below.

C.2.3 INITIALIZATION AND THE FINITE-STEP LIMIT

Set $s = 1 + \beta_0^2 N_\flat/(KL)$. Independence of the current block from its input gives the exact identities

$$\mathbb{E}\frac{\|\boldsymbol{h}^{\ell\flat}\|^2}{N_\flat} = s^\ell, \qquad \mathbb{E}f_0^2 = \frac{s^L}{\gamma_0^2 N_\flat}. \tag{116}$$

Thus $\delta \to 1$ as width grows, and the continuum initialization profile is $\exp(\beta_0^2 \alpha\tau)$. Conditioned on the residual weights and $\boldsymbol{h}^{0\flat}$, the vector $\boldsymbol{z}_\ell$ is linear in the independent readout $(\boldsymbol{W}^L)^\mathsf{T}$. It follows that $\mathbb{E}[(\boldsymbol{h}^{\ell\flat})^\top \boldsymbol{z}_\ell] = 0$ exactly.

The limiting statement first sends $N_\flat, N_\sharp \to \infty$ at each fixed L, with $N_\flat/(N_\sharp L) \to \alpha > 0$, and then sends $L \to \infty$. At fixed depth, the finite-step remainder is controlled as follows. Write $\widehat{\boldsymbol{W}}^{\ell\uparrow} = \boldsymbol{W}^{\ell\uparrow}/\sqrt{N_\flat}$, $\widehat{\boldsymbol{W}}^{\ell\downarrow} = \boldsymbol{W}^{\ell\downarrow}/\sqrt{N_\flat}$, $u = \boldsymbol{h}^{0\flat}/\sqrt{N_\flat}$, and $v = (\boldsymbol{W}^L)^\mathsf{T}/\sqrt{N_\flat}$. Their entries are independent Gaussians of variance $1/N_\flat$. At fixed L, the aspect ratio $N_\sharp/N_\flat$ stays in a compact subset of $(0, \infty)$, so all fixed-order moments of $\|\widehat{\boldsymbol{W}}^{\ell\uparrow}\|_{\text{op}}, \|\widehat{\boldsymbol{W}}^{\ell\downarrow}\|_{\text{op}}, \|u\|, \|v\|$ are uniformly bounded.

Expand the product in (114). Every term beyond (115) contains at least one normalized mixed contraction between the independent initial state and readout, of the form $N_\flat{}^{-1}(\boldsymbol{h}^{0\flat})^\top B(\boldsymbol{W}^L)^\mathsf{T}$, with the matrix B determined by the residual weights. Conditioned on those weights and averaging over both $\boldsymbol{h}^{0\flat}$ and $(\boldsymbol{W}^L)^\mathsf{T}$, its second moment is $N_\flat{}^{-2}\operatorname{tr}(BB^\top)$; its higher even moments have the corresponding Gaussian bounds. The normalized matrix factors in B have bounded operator-norm moments. Hölder's inequality therefore controls each higher insertion, including its dependent remaining factors, in normalized mean square by $C_L/N_\flat$. The number of terms is finite at fixed L. The same argument handles $\delta - 1 = -f_0$. Consequently, for $j \le L$,

$$\mathbb{E}\frac{1}{N_\flat}\|(\boldsymbol{h}^{j\flat})^+ - \boldsymbol{h}^{j\flat} - \boldsymbol{z}_j\|^2 \le \frac{C_L}{N_\flat}. \tag{117}$$

Here and below C_L may depend on $L, \alpha, \eta_{\text{eff}}, \beta_0, \gamma_0$, but not on $N_\flat, N_\sharp$ along the fixed-depth width limit.

To justify self-averaging over the weights as well as over the readout, apply the Gaussian Poincaré inequality to $\|\boldsymbol{h}^{j\flat}\|^2/N_\flat$, $\|\boldsymbol{z}_j\|^2/N_\flat$, and $(\boldsymbol{h}^{j\flat})^\top \boldsymbol{z}_j/N_\flat$. They are finite-degree scalar polynomials in $\widehat{\boldsymbol{W}}^{\ell\uparrow}, \widehat{\boldsymbol{W}}^{\ell\downarrow}, u, v$. Differentiating their vector and matrix products bounds the expected sum of squared vector gradients and squared Frobenius matrix gradients by C_L, using the norm-moment bounds above. Each independent Gaussian coordinate has variance $1/N_\flat$. Since the mixed energy has mean zero, this gives

$$\operatorname{Var}\left(\frac{\|\boldsymbol{h}^{j\flat}\|^2}{N_\flat}\right) + \operatorname{Var}\left(\frac{\|\boldsymbol{z}_j\|^2}{N_\flat}\right) + \mathbb{E}\left[\left(\frac{(\boldsymbol{h}^{j\flat})^\top \boldsymbol{z}_j}{N_\flat}\right)^2\right] \le \frac{C_L}{N_\flat}. \tag{118}$$

Equations (117) and (118) identify the limiting finite-step energy with the initialization energy plus the averaged tangent energy, in probability. This argument allows fixed, finite $\eta_{\text{eff}}, \beta_0, \gamma_0$; it is not a small-learning-rate expansion.

The finite-step reduction takes widths to infinity at fixed depth. The moment recursion below also has a direct joint width–depth limit; extending the remainder bound to that limit requires control of C_L as depth grows.

C.2.4 A CLOSED MOMENT CALCULATION

The second moment of (115) can be evaluated without discarding the factorization $\boldsymbol{W}^{\ell\downarrow}\boldsymbol{W}^{\ell\uparrow}$. Put $\varepsilon = \beta_0/K$, and define

$$\begin{aligned} q_\ell &= \|\boldsymbol{h}^{\ell\flat}\|^2/N_\flat, \qquad & r_\ell &= \|\boldsymbol{W}^{\ell\uparrow}\boldsymbol{h}^{\ell\flat}\|^2/N_\flat, \\ \boldsymbol{M}_0 &= 0, & \boldsymbol{M}_{\ell+1} &= \boldsymbol{A}_\ell \boldsymbol{M}_\ell \boldsymbol{A}_\ell^\top + \varepsilon(r_\ell I + q_\ell \boldsymbol{W}^{\ell\downarrow}(\boldsymbol{W}^{\ell\downarrow})^\top). \end{aligned} \tag{119}$$

If $\boldsymbol{F}_j = \boldsymbol{A}_{L-1}\cdots\boldsymbol{A}_j$, with $\boldsymbol{F}_L = I$, then $\boldsymbol{z}_j = \eta_{\text{eff}} \boldsymbol{M}_j \boldsymbol{F}_j^\top (\boldsymbol{W}^L)^\intercal$. The past and future are independent, so

$$\mathbb{E}\frac{\|\boldsymbol{z}_j\|^2}{N_\flat} = \eta_{\text{eff}}^2 s^{L-j} S_j, \qquad S_j = \mathbb{E}\frac{\operatorname{tr}(\boldsymbol{M}_j^2)}{N_\flat}. \tag{120}$$

To close its evolution, retain five scalar moments:

$$\begin{aligned} Q_j &= \mathbb{E} q_j^2, & P_j &= \mathbb{E}\left[q_j \operatorname{tr}(\boldsymbol{M}_j)/N_\flat\right], \qquad & T_j &= \mathbb{E}\left[(\operatorname{tr}(\boldsymbol{M}_j)/N_\flat)^2\right], \\ S_j &= \mathbb{E}\operatorname{tr}(\boldsymbol{M}_j^2)/N_\flat, & H_j &= \mathbb{E}[(\boldsymbol{h}^{j\flat})^\top \boldsymbol{M}_j \boldsymbol{h}^{j\flat}/N_\flat]. \end{aligned} \tag{121}$$

They start from $(Q_0, P_0, T_0, S_0, H_0) = (1 + 2/N_\flat, 0, 0, 0, 0)$. The factor $1 + 2/N_\flat$ is required for the Gaussian input embedding.

The five moments obey a closed finite-size recursion. Define

$$\begin{aligned} a_* &= 1 + 2\varepsilon^2 N_\sharp(N_\flat + 1) + \varepsilon^4 N_\flat N_\sharp[(N_\flat + 1)(N_\sharp + 1) + 2], \\ b_* &= 2\varepsilon^2 N_\sharp N_\flat + \varepsilon^4 {N_\flat}^2 N_\sharp(N_\flat + N_\sharp + 1), \\ d_* &= 1 + 2\varepsilon^2 N_\sharp N_\flat + \varepsilon^4({N_\flat}^2 {N_\sharp}^2 + 2N_\flat N_\sharp), \\ e_* &= 4\varepsilon^2 N_\sharp/N_\flat + 2\varepsilon^4 N_\sharp(N_\flat + N_\sharp + 1), \\ g_* &= 1 + 2\varepsilon^2 N_\sharp(N_\flat + 2) + \varepsilon^4 N_\sharp N_\flat(N_\sharp + 2)(N_\flat + 2). \end{aligned} \tag{122}$$

All quantities on the right below are evaluated at index j:

$$\begin{aligned} Q_{j+1} &= g_* Q_j, \\ P_{j+1} &= d_* P_j + e_* H_j + [2\varepsilon N_\sharp + 2\varepsilon^3 N_\sharp\{N_\flat(N_\sharp + 1) + 1\}]Q_j, \\ H_{j+1} &= a_* H_j + b_* P_j + [2\varepsilon N_\sharp + \varepsilon^3 N_\sharp N_\flat(N_\flat + 2N_\sharp + 3)]Q_j, \\ T_{j+1} &= d_* T_j + e_* S_j + [4\varepsilon N_\sharp + 4\varepsilon^3 N_\sharp(N_\flat N_\sharp + 1)]P_j \\ &\quad + 4\varepsilon^3 N_\sharp H_j + \varepsilon^2(4{N_\sharp}^2 + 2N_\sharp + 2N_\sharp/N_\flat)Q_j, \\ S_{j+1} &= a_* S_j + b_* T_j + [4\varepsilon N_\sharp + 2\varepsilon^3 N_\sharp N_\flat(N_\flat + 2N_\sharp + 1)]P_j \\ &\quad + 4\varepsilon^3 N_\sharp H_j + \varepsilon^2 N_\sharp(N_\flat + 4N_\sharp + 3)Q_j. \end{aligned} \tag{123}$$

These identities follow by conditioning on $\boldsymbol{h}^{j\flat}$, $\boldsymbol{M}_j$ and pairing the independent Gaussian entries of the current $\boldsymbol{W}^{j\uparrow}$ and $\boldsymbol{W}^{j\downarrow}$. For example, for fixed symmetric M,

$$\mathbb{E}\frac{\mathrm{tr}[(AMA^\top)^2]}{N_\flat} = a_*\frac{\mathrm{tr}(M^2)}{N_\flat} + b_*\left(\frac{\mathrm{tr}\,M}{N_\flat}\right)^2. \tag{124}$$

For one block, $S_1 = \varepsilon^2 N_\sharp(N_\flat + 4N_\sharp + 3)(1 + 2/N_\flat)$. The first trace moment provides another check:

$$\mathbb{E}\frac{\mathrm{tr}\,M_j}{N_\flat} = \frac{2\beta_0 j}{L}s^{j-1}. \tag{125}$$

An important fourth-order contribution is visible in b_*:

$$L\varepsilon^4 {N_\flat}^3 N_\sharp \longrightarrow \beta_0^4\alpha^3. \tag{126}$$

It survives the depth limit. Thus replacing $\boldsymbol{W}^{\ell\downarrow}\boldsymbol{W}^{\ell\uparrow}$ by a matrix with independent Gaussian entries and the same entrywise covariance would not preserve this calculation.

C.2.5 Solving the continuum moments

Let $c = \beta_0^2\alpha$ and $\tau = j/L$. Taking the limits of (123) yields the closed subsystem

$$\begin{aligned}
\dot{Q} &= 2cQ, & Q(0) &= 1,\\
\dot{P} &= 2cP + 2\beta_0 Q, & P(0) &= 0,\\
\dot{T} &= 2cT + 4\beta_0 P, & T(0) &= 0,\\
\dot{S} &= 2cS + c(2 + c\alpha)T + 2\beta_0(2 + c\alpha)P + cQ, & S(0) &= 0.
\end{aligned} \tag{127}$$

The moment H drops out of this subsystem in the limit. In fact these deterministic moment equations also follow along joint sequences with $N_\flat, L \to \infty$ and $N_\flat/K \to \alpha < \infty$, without requiring $N_\sharp \to \infty$. The nonnegative finite-size recursions give, by discrete Grönwall, $\sup_{j\le L}(Q_j + P_j + T_j + S_j + H_j) \le C$ for bounded $N_\flat/K$ and fixed β_0. The omitted coefficients vanish after multiplication by L, and the resulting bounded linear Euler recursion converges to (127).

The first three solutions are $Q = e^{2c\tau}$, $P = 2\beta_0\tau e^{2c\tau}$, and $T = 4\beta_0^2\tau^2 e^{2c\tau}$. Substituting them into the last equation gives

$$\frac{d}{d\tau}\left(e^{-2c\tau}S\right) = c + 4\beta_0^2(2 + c\alpha)\tau + 4c\beta_0^2(2 + c\alpha)\tau^2, \tag{128}$$

and hence

$$S(\tau) = e^{2c\tau}\left[c\tau + 2\beta_0^2(2 + c\alpha)\tau^2 + \frac{4}{3}c\beta_0^2(2 + c\alpha)\tau^3\right]. \tag{129}$$

Combining (120), (117), and (129) proves the general form of the iterated-limit finite-step result:

$$\begin{aligned}
C_h(\tau;0) &= e^{c\tau},\\
C_h(\tau;1) - C_h(\tau;0) &= \eta_{\text{eff}}^2 e^{c(1+\tau)}\left[c\tau + 2(2\beta_0^2 + c^2)\tau^2\left(1 + \frac{2c\tau}{3}\right)\right].
\end{aligned} \tag{130}$$

Here $\beta_0^2(2 + c\alpha) = 2\beta_0^2 + c^2$, so the expression is an exact regrouping of the moment solution. Setting $\beta_0 = 1$ and $\eta_{\text{eff}} = \gamma_0$ gives the baseline used in the numerical checks:

$$\begin{aligned}
C_h(\tau;0) &= e^{\alpha\tau},\\
C_h(\tau;1) &= e^{\alpha\tau} + \gamma_0^2 e^{\alpha(1+\tau)}\left[\alpha\tau + 2(\alpha^2 + 2)\tau^2 + \frac{4}{3}(\alpha^3 + 2\alpha)\tau^3\right].
\end{aligned} \tag{131}$$

Its continuous extension to $\alpha = 0$ is $C_h(\tau;0) = 1$ and $C_h(\tau;1) = 1 + 4\gamma_0^2\tau^2$. For general β_0, replacing α by $\beta_0^2\alpha$ in this baseline alone would miss the separate coupling dependence of the learning-induced term.

The calculation concerns feature motion, not a guarantee of a decrease in squared loss. For instance, (125) gives $f_1 \to 2\eta_{\text{eff}}\beta_0 e^c/\gamma_0$ in the same iterated limit. With $\beta_0 = 1$ and $\eta_{\text{eff}} = \gamma_0$, this overshoots the target. Training the input or readout, using multiple samples, or adding a nonlinearity changes the calculation.

C.2.6 THE SAME CHECKPOINT FROM THE DMFT

The population equations recover the full one-step formula directly. Keep the normalized-input, linear, frozen-endpoint setting above, with the same η_{eff} for both residual projections. The input is fixed; the second argument of each representative field is the training step $t = 0, 1$. At initialization the forward and backward processes are independent, with

$$\begin{aligned}\mathbb{E}[\boldsymbol{\hbar}^{\flat}(\tau;0)\boldsymbol{\hbar}^{\flat}(s;0)] &= e^{c\min(\tau,s)},\\ \mathbb{E}[\boldsymbol{g}_1^{\flat}(\tau;0)\boldsymbol{g}_1^{\flat}(s;0)] &= e^{c(1-\max(\tau,s))}.\end{aligned} \tag{132}$$

Thus the limiting initial output is zero and its loss feedback is one. The sources retain their correlations between training steps in (55); different forward/backward source families remain independent. Since $\boldsymbol{g}_1^{\sharp}(\tau;0) = r^{\sharp}(\tau;0)$ and $\boldsymbol{\hbar}^{\sharp}(\tau;0) = u^{\sharp}(\tau;0)$, the hidden cross moment satisfies $\Phi_{10}^{\sharp} = \Phi_{10}^{\flat}$. The residual cross moment then obeys $\partial_\tau \Phi_{10}^{\flat} = c\Phi_{10}^{\flat}$, with initial value one, hence $\Phi_{10}^{\flat} = e^{c\tau}$.

Let $A(\tau) = A^{\flat}(\tau; 1, 0)$. There is no forward response to current backward sources. The hidden equation and the residual equation at step one reduce to

$$\begin{aligned}\boldsymbol{\hbar}^{\sharp}(\tau;1) &= u^{\sharp}(\tau;1) + \lambda_h(\tau)\boldsymbol{g}_1^{\sharp}(\tau;0), \qquad \lambda_h = A_{10}^{\sharp} = \beta_0\alpha A + \eta_{\text{eff}}e^{c\tau},\\ d\boldsymbol{\hbar}^{\flat}(\tau;1) &= du(\tau;1) + k(\tau)\boldsymbol{g}_1^{\flat}(\tau;0)\,d\tau, \qquad k = cA + 2\beta_0\eta_{\text{eff}}e^{c\tau}.\end{aligned} \tag{133}$$

The two direct-learning terms in k come from the up-projection response and the down-projection learning history. A unit initial backward-source increment at depth τ changes $\boldsymbol{g}_1^{\flat}(s;0)$ by one for $s < \tau$, so $A(\tau) = \int_0^\tau k(s)\,ds$. Consequently

$$A(\tau) = 2\beta_0\eta_{\text{eff}}\tau e^{c\tau}, \quad \lambda_h(\tau) = \eta_{\text{eff}}e^{c\tau}(1 + 2c\tau). \tag{134}$$

Source independence gives $\Phi_{11}^{\sharp} = C_h(\tau;1) + \eta_{\text{eff}}^2 e^{c(1+\tau)}(1 + 2c\tau)^2$, whereas (132) gives $\mathbb{E}[\boldsymbol{\hbar}^{\flat}(\tau;1)\boldsymbol{g}_1^{\flat}(\tau;0)] = e^{c(1-\tau)}A(\tau) = 2\beta_0\eta_{\text{eff}}\tau e^{c}$. Thus the trained fields are correlated even though their source families are independent. The variance equation is

$$\begin{aligned}\partial_\tau C_h(\tau;1) &= cC_h(\tau;1) + \eta_{\text{eff}}^2 e^{c(1+\tau)}\left[c(1+2c\tau)^2 + 8\beta_0^2\tau(1+c\tau)\right],\\ C_h(0;1) &= 1.\end{aligned} \tag{135}$$

Integration recovers (130). The frozen readout also gives $f_1 = 2\beta_0\eta_{\text{eff}}e^{c}/\gamma_0$. This calculation retains a finite SGD step; it does not expand in η_{eff}.

C.2.7 COUPLING AND DIRECT LOCAL HIDDEN LEARNING

Fix $c > 0$, $\gamma_0 > 0$, and $\tau \in (0, 1]$. Within the limiting family, set $\alpha = c/\beta_0^2$ and allow the positive step multiplier to depend on β_0. Equation (130) gives

$$\frac{C_h(\tau;1) - C_h(\tau;0)}{(\eta_{\text{eff}}\beta_0)^2} \longrightarrow 4e^{c(1+\tau)}\tau^2\left(1 + \frac{2c\tau}{3}\right) \quad \text{as } \beta_0 \to \infty. \tag{136}$$

The limit is strictly positive. Keeping the first-step residual energy bounded therefore requires $\eta_{\text{eff}} = O(\beta_0^{-1})$. The output identity gives the same scale: $f_1 \to 2(\eta_{\text{eff}}\beta_0/\gamma_0)e^{c}$. These are comparisons between limits taken at fixed constants, without interchanging growing coupling with the width and depth limits.

For a block ℓ, hold the incoming residual vector $\boldsymbol{h}^{\ell\flat}$ and outgoing backward vector $\boldsymbol{g}_1^{(\ell+1)\flat}$ at their initial values. The linear hidden variables are $\boldsymbol{h}^{\ell\sharp} = \boldsymbol{W}^{\ell\uparrow}\boldsymbol{h}^{\ell\flat}/\sqrt{N_\flat}$ and $\boldsymbol{g}_1^{\ell\sharp} = (\boldsymbol{W}^{\ell\downarrow})^\top \boldsymbol{g}_1^{(\ell+1)\flat}/\sqrt{N_\flat}$. The parameter increments in (114) give the exact local changes

$$\begin{aligned}\Delta_{\text{loc}}\boldsymbol{h}^{\ell\sharp} &= \frac{\Delta\boldsymbol{W}^{\ell\uparrow}\boldsymbol{h}^{\ell\flat}}{\sqrt{N_\flat}} = \eta_{\text{eff}}\delta\,\frac{\|\boldsymbol{h}^{\ell\flat}\|^2}{N_\flat}\boldsymbol{g}_1^{\ell\sharp},\\ \Delta_{\text{loc}}\boldsymbol{g}_1^{\ell\sharp} &= \frac{\Delta(\boldsymbol{W}^{\ell\downarrow})^\top \boldsymbol{g}_1^{(\ell+1)\flat}}{\sqrt{N_\flat}} = \eta_{\text{eff}}\delta\,\frac{\|\boldsymbol{g}_1^{(\ell+1)\flat}\|^2}{N_\flat}\boldsymbol{h}^{\ell\sharp}.\end{aligned} \tag{137}$$

At initialization, the current block's matrices are independent of these neighboring vectors. For $\ell/L \to \tau$, their norm limits are $\|\boldsymbol{h}^{\ell\flat}\|^2/N_\flat \to e^{c\tau}$ and $\|\boldsymbol{g}_1^{(\ell+1)\flat}\|^2/N_\flat \to e^{c(1-\tau)}$. The backward

limit follows by applying the initialization calculation to the transposed blocks in reverse order. Since $\delta \to 1$, conditional Gaussian averaging over the hidden coordinates gives

$$\begin{aligned} N_\sharp{}^{-1}\|\Delta_{\mathrm{loc}}\boldsymbol{h}^{\ell\sharp}\|^2 &\longrightarrow \eta_{\mathrm{eff}}^2 e^{c(1+\tau)}, \\ N_\sharp{}^{-1}\|\Delta_{\mathrm{loc}}\boldsymbol{g}_1^{\ell\sharp}\|^2 &\longrightarrow \eta_{\mathrm{eff}}^2 e^{c(2-\tau)}. \end{aligned} \tag{138}$$

Both direct local changes therefore vanish in the bounded-energy large-β_0 family. The full hidden activation can still move through its changing input:

$$\boldsymbol{h}^{\ell\sharp}(1) - \boldsymbol{h}^{\ell\sharp}(0) = \Delta_{\mathrm{loc}}\boldsymbol{h}^{\ell\sharp} + \frac{(\boldsymbol{W}^{\ell\uparrow})^{+}[\boldsymbol{h}^{\ell\flat}(1) - \boldsymbol{h}^{\ell\flat}(0)]}{\sqrt{N_\flat}}. \tag{139}$$

The second term is not controlled by the vanishing local step alone. The backward hidden variable likewise receives changes through the updated downstream backward vector.

The coupling factor is also visible in a nonlinear block. Holding its incoming residual state fixed, the down-projection update on one scalar loss contributes

$$\Delta h_i^{(\ell+1)\flat}\Big|_{\mathrm{down}} = -\frac{\beta_0\eta_{\mathrm{eff}}}{L}\frac{\partial\mathcal{L}}{\partial f}\, g_i^{(\ell+1)\flat}\frac{1}{N_\sharp}\sum_{a=1}^{N_\sharp}\phi(h_a^{\ell\sharp})^2. \tag{140}$$

The linear checkpoint includes both projection updates and their propagation through all subsequent blocks.

C.2.8 Gaussian hidden preactivations along the SGD semi-freeze family

The same suppression of local learning also removes the non-Gaussian deformation of hidden preactivations in the regular SGD semi-freeze family. Take the population and depth limits at fixed β_0 first, then consider

$$\alpha = \frac{c}{\beta_0^2}, \qquad c > 0, \qquad \eta_{\mathrm{eff},\uparrow} = \eta_\uparrow\gamma_0\beta_0 = O(\beta_0^{-1}), \qquad \eta_{\mathrm{eff},\downarrow} = \eta_\downarrow\gamma_0\beta_0 = O(\beta_0^{-1}). \tag{141}$$

Fix a depth and a finite training horizon T. Let $\mathcal{X}$ contain all training and probe inputs, and set $\mathcal{Z} = \mathcal{X} \times \{0, \dots, T\}$. Work on a regular training window where the deterministic residual kernels, averaged responses, and loss feedback remain bounded along this family. The endpoint learning rates may remain fixed.

Stack the hidden fields on this collection into $\boldsymbol{\mathcal{h}}_\beta^\sharp, \boldsymbol{\mathcal{g}}_\beta^\sharp$. Equation (57) has the form

$$\begin{aligned} \boldsymbol{\mathcal{h}}_\beta^\sharp &= \boldsymbol{u}_\beta + \mathsf{E}_\beta\boldsymbol{\mathcal{g}}_\beta^\sharp, & \mathsf{E}_\beta &= \frac{c}{\beta_0}\mathsf{A}_\beta + \eta_{\mathrm{eff},\uparrow}\mathsf{S}_{\Phi,\beta}, \\ \boldsymbol{\mathcal{g}}_\beta^\sharp &= D_\phi(\boldsymbol{\mathcal{h}}_\beta^\sharp)[\boldsymbol{r}_\beta + \mathsf{F}_\beta\phi(\boldsymbol{\mathcal{h}}_\beta^\sharp)], & \mathsf{F}_\beta &= \frac{c}{\beta_0}\mathsf{B}_\beta + \eta_{\mathrm{eff},\downarrow}\mathsf{S}_{G,\beta}. \end{aligned} \tag{142}$$

Here $\mathsf{A}_\beta, \mathsf{B}_\beta$ collect the residual responses, $\mathsf{S}_{\Phi,\beta}, \mathsf{S}_{G,\beta}$ are the loss-weighted history maps, and D_ϕ is diagonal with the activation derivatives, repeated over output components. The centered Gaussian sources $\boldsymbol{u}_\beta, \boldsymbol{r}_\beta$ have the residual covariances in (55). In particular, the reaction term $(c/\beta_0)\mathsf{A}_\beta\boldsymbol{\mathcal{g}}_\beta^\sharp$ accounts for dependence between the initial up weights and their trained residual inputs. Freezing the up weights alone would not justify discarding that term.

Write $e_\beta = \|\mathsf{E}_\beta\|_{\mathrm{op}}$, $f_\beta = \|\mathsf{F}_\beta\|_{\mathrm{op}}$, $u_* = \|\boldsymbol{u}_\beta\|_{L^2}$, and $r_* = \|\boldsymbol{r}_\beta\|_{L^2}$. The regular-window bounds give $e_\beta + f_\beta = O_T(\beta_0^{-1})$ and $u_* + r_* = O_T(1)$. For an activation with bounded derivative $M = \|\phi'\|_\infty$, including ReLU, put $b = \|\phi(\mathbf{0})\|$. Pointwise, $\|\boldsymbol{\mathcal{h}}_\beta^\sharp\| \le \|\boldsymbol{u}_\beta\| + e_\beta\|\boldsymbol{\mathcal{g}}_\beta^\sharp\|$ and $\|\boldsymbol{\mathcal{g}}_\beta^\sharp\| \le M[\|\boldsymbol{r}_\beta\| + f_\beta(b + M\|\boldsymbol{\mathcal{h}}_\beta^\sharp\|)]$. Solving these inequalities before taking L^2 norms gives, for sufficiently large β_0,

$$\|\boldsymbol{\mathcal{g}}_\beta^\sharp\|_{L^2} \le \frac{M\{r_* + f_\beta(b + Mu_*)\}}{1 - M^2 e_\beta f_\beta} = O_T(1). \tag{143}$$

This controls the hidden field multiplying the small coefficients without an independence assumption. The fields and sources already share a probability space, so

$$\begin{aligned} W_2\Big(\mathcal{L}(\boldsymbol{\mathcal{h}}_\beta^\sharp), \mathcal{N}(0, \Sigma_\beta)\Big) &\le \|\boldsymbol{\mathcal{h}}_\beta^\sharp - \boldsymbol{u}_\beta\|_{L^2} \le e_\beta\|\boldsymbol{\mathcal{g}}_\beta^\sharp\|_{L^2} = O_T(\beta_0^{-1}), \\ (\Sigma_\beta)_{zz'} &= \Phi_\beta^\flat(z, z'). \end{aligned} \tag{144}$$

Here W_2 is the quadratic Wasserstein distance. If Σ_β converges, this gives a joint Gaussian limit over the selected inputs and training steps. Its covariance may change by order one during training; the preactivations need not remain at their initial values. The pointwise bound also controls every fixed L^p norm, so fixed-order moments approach those of the Gaussian source.

Pre-RMS normalization. For a residual branch using $\mathcal{N}(\boldsymbol{h}) = \boldsymbol{h}/\nu_{N_\flat}(\boldsymbol{h})$ with $\nu_{N_\flat}(\boldsymbol{h})^2 = \|\boldsymbol{h}\|^2/N_\flat + \epsilon$ and unit gain, the source covariance and the direct-learning history kernel are both

$$\Phi^{\mathcal{N}}_\beta(z,z') = \frac{\Phi^\flat_\beta(z,z')}{\nu_\beta(z)\nu_\beta(z')}, \qquad \nu_\beta(z)^2 = \Phi^\flat_\beta(z,z) + \epsilon. \tag{145}$$

The reaction also changes, because backpropagation through the up map $\boldsymbol{W}^{\ell\uparrow}$ contains the full normalization Jacobian:

$$\begin{aligned} D\mathcal{N}(\boldsymbol{h}) &= \frac{I}{\nu_{N_\flat}(\boldsymbol{h})} - \frac{\boldsymbol{h}\boldsymbol{h}^\mathsf{T}}{N_\flat\, \nu_{N_\flat}(\boldsymbol{h})^3}, \\ \boldsymbol{g}^\flat_q - \boldsymbol{g}^{+\flat}_q &= \frac{\beta_0\sqrt{N_\flat}}{K} D\mathcal{N}(\boldsymbol{h})(\boldsymbol{W}^{\ell\uparrow})^\mathsf{T}\boldsymbol{g}^\sharp_q. \end{aligned} \tag{146}$$

Let $\mathsf{T}_{p,N_\flat}(z,z')$ be the full response of $\boldsymbol{h}^\flat(z)$ to an additive perturbation of its residual backward variable at z', with population coefficients fixed. The raw initial up-transpose source enters that variable through $D\mathcal{N}(z')$, while the queried normalized input has derivative $D\mathcal{N}(z)$. The response replacing $A^\flat_p$ is therefore

$$A^R_{\beta,p}(z,z') = \lim_{N_\flat\to\infty} \frac{1}{N_\flat}\mathbb{E}\ \mathrm{tr}[D\mathcal{N}(z)\mathsf{T}_{p,N_\flat}(z,z')D\mathcal{N}(z')]\,. \tag{147}$$

This retains the rank-one term in both Jacobians. Contraction over an exposed initial up-weight entry contributes $(\beta_0/K)\mathbb{E}\,\mathrm{tr}[D\mathcal{N}(z)\mathsf{T}_{p,N_\flat}D\mathcal{N}(z')]$, so the population-counting factor remains $\beta_0 N_\flat/K$. Since $\|D\mathcal{N}\|_{\mathrm{op}} \le \nu_{N_\flat}^{-1}$, a lower bound $\nu_{N_\flat} \ge \nu_{\min} > 0$ gives

$$|A^R_{\beta,p}(z,z')| \le \nu_{\min}^{-2} \limsup_{N_\flat\to\infty} \left(\frac{1}{N_\flat}\mathbb{E}\|\mathsf{T}_{p,N_\flat}(z,z')\|^2_{\mathrm{F}}\right)^{1/2}. \tag{148}$$

Thus a bounded normalized response leaves the reaction of order $\beta_0\alpha = c/\beta_0$. A fixed positive ϵ bounds the inverse RMS factors directly; for $\epsilon = 0$ the regular window requires nonvanishing RMS scales and controlled inverse RMS factors. The hidden forward equation becomes

$$\boldsymbol{\hbar}^\sharp_\beta(z) = u^R_\beta(z) + \frac{c}{\beta_0}\sum_{p,z'} A^R_{\beta,p}(z,z')\boldsymbol{\mathcal{g}}^\sharp_{\beta,p}(z') + \eta_{\mathrm{eff},\uparrow}\mathcal{S}_t[\boldsymbol{\mathcal{g}}^\sharp_{\beta,p}(z')\Phi^{\mathcal{N}}_\beta(z,z')]. \tag{149}$$

Pre-RMS leaves the down-projection gradient and hidden backward identity unchanged. The second equation in (57) therefore retains its form: its Gaussian source covariance and direct history still use $G^\flat$, while $B^\flat$ is evaluated through the full normalized network. Writing that response as B^R_β gives $\mathsf{F}_\beta = (c/\beta_0)\mathsf{B}^R_\beta + \eta_{\mathrm{eff},\downarrow}\mathsf{S}_{G,\beta}$. The same moment bound and Gaussian approximation consequently apply with $\Sigma_\beta = \Phi^{\mathcal{N}}_\beta$. For unit RMS with $\epsilon = 0$, each limiting marginal has variance one, while its correlations across inputs and training steps can still evolve. With $\epsilon > 0$, the marginal variance is $\Phi^\flat_\beta(z,z)/(\Phi^\flat_\beta(z,z)+\epsilon)$.

This conclusion concerns preactivations $\boldsymbol{\hbar}^\sharp$, not $\phi(\boldsymbol{\hbar}^\sharp)$ or the hidden backward variable containing $\phi'(\boldsymbol{\hbar}^\sharp)$. Resolved non-Gaussianity in a fixed-input, fixed-depth preactivation slice of nonvanishing variance is a one-way diagnostic against having reached this regular semi-freeze description on the measured time scale. Approximate Gaussianity alone does not establish semi-freeze. The bound depends on the training window, so substantial deformation at finite coupling or after longer training is compatible with this limiting statement.

C.2.9 Finite-network illustrations

Figures 12 and 13 compare the full finite SGD step with the continuum prediction. At larger α, the discrepancies include both a finite-depth correction to the tangent energy and a higher-order remainder of the full update. The latter is measured by

$$\mathcal{R}_L = \frac{1}{N_\flat}\|\boldsymbol{h}^{L\flat}(1) - \boldsymbol{h}^{L\flat} - (1-f_0)\boldsymbol{z}_L\|^2. \tag{150}$$

This squared state remainder is not the total energy correction, which also contains cross terms and the random factor $1 - f_0$. Implementation checks, sampling conventions, and numerical tables are given in Appendix F.2.

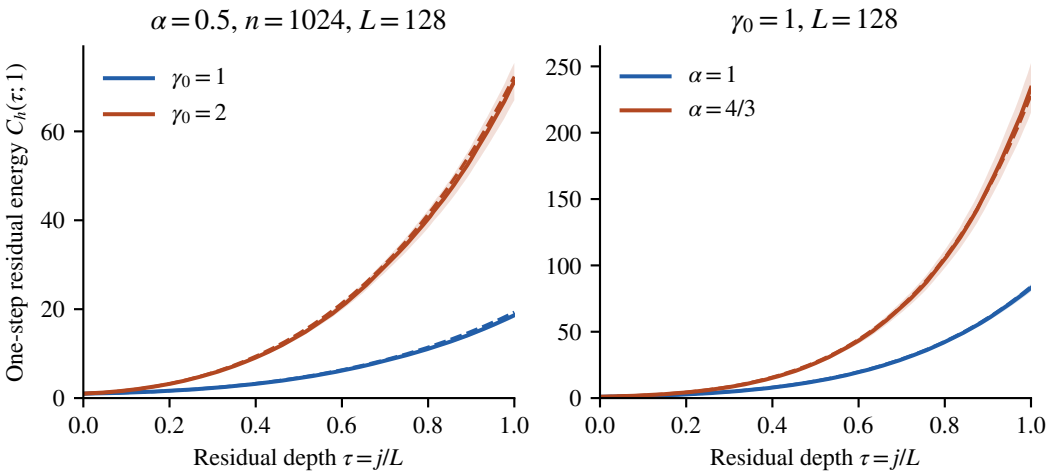


Figure 12: Solid curves: full finite-step means; dashed curves: (131). Shading is ± 2 Monte Carlo standard errors, not a simultaneous confidence band. All runs freeze both endpoints and update both residual projections.

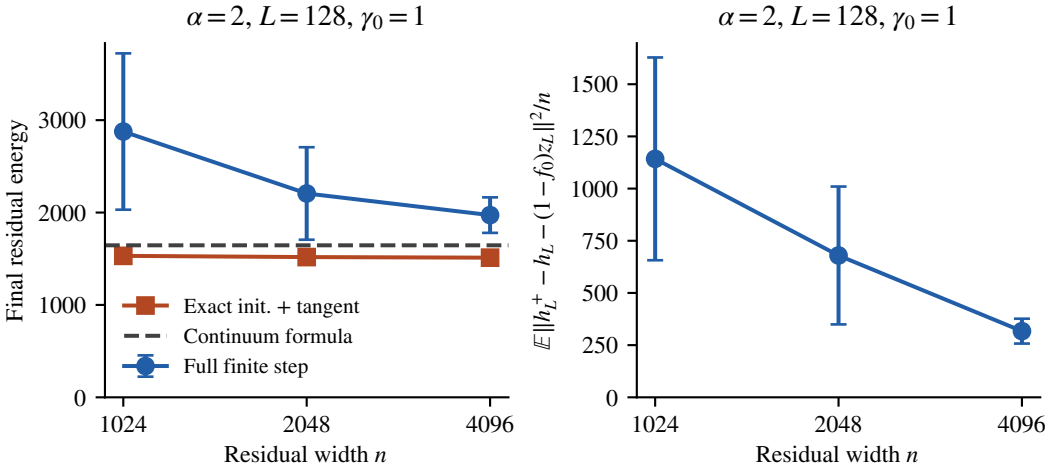


Figure 13: A stress test at $\alpha = 2$. Left: the actual finite-step energy, the exact initialization-plus-unit-error tangent energy, and the continuum formula. Right: the directly measured higher-order remainder (150). Error bars are ± 2 Monte Carlo standard errors. The tangent curve is a diagnostic, not a replacement training rule.

D FINITE-SIZE ERRORS OF MACROSCOPIC QUANTITIES

For a scalar observable O at a fixed input and training step, compare the finite model to the population value o_∞ of the same optimizer. Averaging over initialization separates mean bias from centered variability:

$$\mathbb{E}|O - o_\infty|^2 = |\mathbb{E}O - o_\infty|^2 + \mathrm{Var}(O). \tag{151}$$

Our size comparisons use $o_\infty(N_\flat/K)$ at the finite ratio $N_\flat/K$. Comparing to a different target α would also include $|o_\infty(N_\flat/K) - o_\infty(\alpha)|$.

D.1 HOW AVERAGING OVER DEPTH ENTERS

Consider a local scalar sampling error Z_ℓ at a constant population kernel, as in the spectral calculation below. If the two coordinate samples are independently redrawn for every block, then for $\overline{Z}_L = L^{-1}\sum_{\ell=0}^{L-1} Z_\ell$,

$$\mathbb{E}\overline{Z}_L = \mathbb{E}Z_0, \qquad \mathrm{Var}(\overline{Z}_L) = \mathrm{Var}(Z_0)/L. \tag{152}$$

The inverse-width mean bias survives; the centered standard deviation is reduced by $L^{-1/2}$. If the blocks share residual samples X and redraw only the hidden samples, conditional independence gives instead

$$\mathrm{Var}(\overline{Z}_L) = \mathrm{Var}\big(\mathbb{E}[Z_0 \mid X]\big) + \frac{1}{L}\mathbb{E}\big[\mathrm{Var}(Z_0 \mid X)\big]. \tag{153}$$

The shared residual randomness leaves a fluctuation floor.

The population saddle averages over $N_\flat$ complete residual histories and $N_\sharp$ hidden samples per block. Hidden kernels and responses enter the residual equations with weight $1/L$; for example, the forward-source covariance changes by $(\beta_0^2 N_\flat/(KL))\,\delta\Phi^{\ell\sharp}$. Residual sampling therefore retains its ${N_\flat}^{-1/2}$ scale, while local hidden sampling errors enter through a depth average. This averaging also applies directly to hidden observables integrated over depth, including weighted averages $L^{-1}\sum_\ell w(\ell/L)\widehat{F}_\ell$, where $\widehat{F}_\ell$ averages a channel observable within block ℓ and w is a fixed bounded weight. A single block has only the local ${N_\sharp}^{-1/2}$ sampling scale: when $N_\sharp = 1$, its hidden observable can remain random even as the depth-integrated observable concentrates. The Gaussian sources themselves are part of the limiting dynamics; finite-size error concerns their empirical statistics and the self-consistent coefficients.

These errors perturb both the population kernels and the loss feedback. Collect the kernels, responses, and outputs in Θ and write their closure as $\Theta = \mathcal{F}_L(\Theta)$. To first order in the empirical sampling defect ε_L,

$$(I - D\mathcal{F}_L(\Theta_*))\,\delta\Theta = \varepsilon_L. \tag{154}$$

The propagated contributions below include this full response. The source responses in (56) hold population kernels and loss feedback fixed; they enter this linearization but are not themselves the full macroscopic response.

For a trained network, the averaging condition must concern each block's propagated effect on the observable. In a sampling representation that separates the shared residual sample from the hidden samples, let $\mathcal{R}$ denote the shared information before the hidden samples are revealed. It is not the complete trained residual trajectory, which already depends on those samples. Write

$$b = \mathbb{E}O - o_\infty(N_\flat/K), \quad S = \mathbb{E}[O \mid \mathcal{R}] - \mathbb{E}O, \quad H = O - \mathbb{E}[O \mid \mathcal{R}]. \tag{155}$$

Conditional centering gives the exact identity $\mathbb{E}|O - o_\infty(N_\flat/K)|^2 = b^2 + \mathbb{E}S^2 + \mathbb{E}H^2$. Suppose the shared component has root-mean-square size $O({N_\flat}^{-1/2})$. For the hidden contribution, write $H = L^{-1}\sum_\ell \xi_\ell$ with $\mathbb{E}[\xi_\ell \mid \mathcal{R}] = 0$. A sufficient averaging condition is

$$\mathbb{E}\sum_{\ell,k=0}^{L-1}\big|\mathrm{Cov}(\xi_\ell, \xi_k \mid \mathcal{R})\big| \le CL/N_\sharp. \tag{156}$$

It gives $\mathbb{E}H^2 \le C/(N_\sharp L)$, separately from the shared ${N_\flat}^{-1/2}$ contribution. The variables ξ_ℓ include propagation through the trained network, with any random response weight included before centering. Thus (156) is a condition on the propagated fluctuations, not independence of trained layers.

D.2 Spectral bias from a finite hidden population

The spectral comparison uses scalar actions against coupled test functions. Sampling the kernel and both test functions on the same coordinates gives a common observable at every size, even though the matrices themselves act on different spaces. This calculation isolates the bias introduced by a finite spectral window.

D.2.1 An exact coupled comparison

Consider a real finite-rank kernel

$$k(y,x)=\sum_{j=1}^{r} s_j\psi_j(y)\varphi_j(x), \qquad S=\operatorname{diag}(s_1,\dots,s_r), \tag{157}$$

where $s_j>0$ and the two function systems are orthonormal in their respective probability measures. Sample $y_1,\dots,y_{N_\sharp}$ and $x_1,\dots,x_{N_\flat}$ independently and set

$$\begin{aligned} Y_{aj}&=\psi_j(y_a), \qquad X_{ij}=\varphi_j(x_i),\\ P&=Y^{\mathsf T}Y/N_\sharp, \qquad Q=X^{\mathsf T}X/N_\flat, \quad M=YSX^{\mathsf T}/\sqrt{N_\sharp N_\flat}. \end{aligned} \tag{158}$$

For fixed test coefficients $v,w\in\mathbb{R}^r$, let $v_{N_\sharp}=Yv/\sqrt{N_\sharp}$ and $w_{N_\flat}=Xw/\sqrt{N_\flat}$. Write F_c for the singular-value map $f_c(s)=\min(s/c,1)$ with fixed $c>0$. Its empirical action and population target are

$$A=v_{N_\sharp}^{\mathsf T}F_c(M)w_{N_\flat}, \qquad a_0=v^{\mathsf T}F_c(S)w. \tag{159}$$

Polar factorization gives the exact identity

$$A=v^{\mathsf T}P^{1/2}F_c(P^{1/2}SQ^{1/2})Q^{1/2}w. \tag{160}$$

For singular Gram matrices the same identity follows by using partial isometries. The square roots outside F_c are not inverse square roots.

The population-transformed kernel can also be evaluated at the same sampling points:

$$B=v^{\mathsf T}PF_c(S)Qw, \qquad Z=A-B. \tag{161}$$

Independence of the two sample sets gives $\mathbb{E}B=a_0$, hence $\mathbb{E}Z=\mathbb{E}A-a_0$. This paired difference isolates the additional spectral-processing bias. Its variance need not equal the variance of the physical action A.

If the sampled and population spectra both lie in the linear region, then, for every realization,

$$A=v^{\mathsf T}PSQw/c=B, \qquad Z=0. \tag{162}$$

The condition must hold for both spectra; a population singular value below the threshold does not prohibit a finite sample from crossing it. In update coordinates the same check is $-\eta_M F_\lambda(D)=-(\eta_M/\lambda)D$. Thus the effective SGD rate is η_M/λ. Above threshold each update singular value is η_M; this is amplitude normalization, not equality with Adam's update direction.

D.2.2 Mean bias and centered fluctuations have different orders

For fixed rank, bounded feature functions, and a spectrum separated from c, the scalar map

$$\mathcal{A}(P,Q)=v^{\mathsf T}P^{1/2}F_c(P^{1/2}SQ^{1/2})Q^{1/2}w$$

has three bounded derivatives near $(P,Q)=(I,I)$. Repeated singular values away from the clipping threshold are allowed. Let $\Xi_\sharp(y)=\psi(y)\psi(y)^{\mathsf T}-I$ and $\Xi_\flat(x)=\varphi(x)\varphi(x)^{\mathsf T}-I$. A second-order expansion and independence give

$$\begin{aligned} \mathbb{E}Z&=\frac{B_\sharp}{N_\sharp}+\frac{B_\flat}{N_\flat}+O({N_\sharp}^{-3/2}+{N_\flat}^{-3/2}),\\ B_\sharp&=\tfrac12\mathbb{E}\,D^2_{PP}\mathcal{A}(I,I)[\Xi_\sharp,\Xi_\sharp],\\ B_\flat&=\tfrac12\mathbb{E}\,D^2_{QQ}\mathcal{A}(I,I)[\Xi_\flat,\Xi_\flat]. \end{aligned} \tag{163}$$

Centering eliminates the first-order terms; independence eliminates the mixed second-order mean. The remaining coefficients depend on the spectrum, clip, tests, and sampling law, and can vanish. For bounded fixed-dimensional sample covariances, $\mathbb{E}\|P - I\|^3 = O({N_\sharp}^{-3/2})$ and $\mathbb{E}\|P - I\|^4 = O({N_\sharp}^{-2})$, and likewise for Q. The third moments control the local Taylor remainder. The fourth moments bound the probability of leaving the smooth neighborhood by $O({N_\sharp}^{-2} + {N_\flat}^{-2})$; bounded features control the observable and polynomial on that event. This proves the displayed remainder for the scalar action.

For the centered first-order part of Z, define

$$\xi_\sharp(y) = D_P(\mathcal{A} - B)(I, I)[\Xi_\sharp(y)], \qquad \xi_\flat(x) = D_Q(\mathcal{A} - B)(I, I)[\Xi_\flat(x)].$$

The same expansion yields

$$\mathrm{Var}(Z) = \frac{\mathrm{Var}(\xi_\sharp)}{N_\sharp} + \frac{\mathrm{Var}(\xi_\flat)}{N_\flat} + o({N_\sharp}^{-1} + {N_\flat}^{-1}). \tag{164}$$

Thus centered fluctuations and mean bias generally have different width powers. Either leading coefficient can vanish for particular spectra or tests.

D.2.3 A CONTINUOUS-KERNEL CASE WITH AN INFINITE SPECTRAL TAIL

The inverse-width bias is not restricted to kernels of small finite rank. Here is a one-sided version that retains an infinite tail. Assume $\sum_j s_j^2 < \infty$ and $\sup_{j,y} |\psi_j(y)| < \infty$, sample the hidden side, and keep the residual side at its population value. For a fixed mode j with $s_j > 0$, write

$$u(y) = (s_k \psi_k(y))_{k \ge 1}, \qquad S_{N_\sharp} = \frac{1}{N_\sharp} \sum_{a=1}^{N_\sharp} u(y_a) \otimes u(y_a), \qquad \mathbb{E} S_{N_\sharp} = S^2. \tag{165}$$

The coupled mode action is exactly $A_j = g(S_{N_\sharp})_{jj}/s_j$, where $g(t) = \min(t/c, \sqrt{t})$. Suppose c^2 is outside the spectrum of S^2. With repeated arguments interpreted by continuity, the second divided difference gives

$$\begin{aligned} B_j &= s_j \sum_k s_k^2 \,\mathrm{Var}(\psi_j \psi_k)\, g^{[2]}(s_j^2, s_k^2, s_j^2), \\ \mathbb{E} A_j - f_c(s_j) &= \frac{B_j}{N_\sharp} + O({N_\sharp}^{-3/2}). \end{aligned} \tag{166}$$

This follows by expanding g at S^2. Indeed, $\mathbb{E}[(S_{N_\sharp} - S^2)_{jk}^2] = s_j^2 s_k^2 \,\mathrm{Var}(\psi_j \psi_k)/N_\sharp$. The second-order contribution to $g(S_{N_\sharp})_{jj}$ is the sum of these variances times $g^{[2]}(s_j^2, s_k^2, s_j^2)$. The series converges absolutely because the features are bounded, the test mode and threshold are fixed, and $\sum_k s_k^2 < \infty$.

For completeness, smooth functional calculus is available here even though g has a kink at c^2. A fixed contour Γ enclosing the finitely many eigenvalues above c^2 gives, near S^2,

$$g(T) = T/c + \frac{1}{2\pi i} \oint_\Gamma (\sqrt{z} - z/c)(zI - T)^{-1}\, dz. \tag{167}$$

The contour lies away from zero and from the spectrum. The resolvent expansion has a third-order operator-norm remainder. The bounded Hilbert-space random vector $u(y)$ gives $\mathbb{E}\|S_{N_\sharp} - S^2\|_{\mathrm{HS}}^3 = O({N_\sharp}^{-3/2})$ and fourth moment $O({N_\sharp}^{-2})$. As above, this controls the local remainder and the exceptional event. No infinite empirical Gram matrix is assumed to approach the identity in operator norm. This proves the one-sided, fixed-test statement, not a uniform two-sided infinite-rank bound.

A threshold control. For a rank-one kernel, take its two singular functions as the tests and put $p = {N_\sharp}^{-1} \sum_a \psi(y_a)^2$, $q = {N_\flat}^{-1} \sum_i \varphi(x_i)^2$, $z = pq$, and $r = s/c$. Equations (159) and (161) reduce exactly to

$$A = \min(rz, \sqrt{z}), \qquad B = \min(r, 1)z. \tag{168}$$

Write $v_\sharp = \mathrm{Var}(\psi^2)$, $v_\flat = \mathrm{Var}(\varphi^2)$, and $V = v_\sharp/N_\sharp + v_\flat/N_\flat$. In a strictly saturated region $r > 1$, bounded features and a fixed threshold gap give

$$\mathbb{E}Z = -\frac{v_\sharp}{8N_\sharp} - \frac{v_\flat}{8N_\flat} + o({N_\sharp}^{-1} + {N_\flat}^{-1}). \tag{169}$$

At the exact threshold $r = 1$, however, $Z = -(z - \sqrt{z})_+$. Its leading term is $-(z-1)_+/2$. The scalar central limit theorem, with convergence of the corresponding first absolute moments, gives

$$\mathbb{E}Z = -\frac{\sqrt{V}}{2\sqrt{2\pi}} + o(\sqrt{V}). \tag{170}$$

At the hard-clipping kink this central-limit scaling replaces the smooth inverse-width expansion.

The sampled-kernel experiments in Appendix F.3 test the smooth bias, centered fluctuations, and threshold control.

D.3 Combining the finite-size contributions

The depth error concerns population statistics. At initialization, for example, the finite-depth variance recursion is the explicit Euler step $C_{\ell+1} = C_\ell + (c/L)\mathbb{E}[\phi(\sqrt{C_\ell}Z)^2]$ for the unnormalized equation in (3). Its local error is $O(L^{-2})$, giving $O(L^{-1})$ over the unit depth interval when the response remains bounded. For trained networks, the weak depth expansion applies to the self-consistent kernels and responses in (57)–(59), retaining the discrete training histories and forward/backward boundary coupling. A local remainder of order L^{-2} and bounded propagation give the same depth scale.

Sampling bias comes from the second response of the closed population equations. Eliminate the local hidden moments in favor of their sampling measures μ_ℓ. These measures describe complete channel histories, including the coordinate-pair moment states for Adam. For SGD and entrywise Adam, nonlinearities and optimizer states act on individual channels or coordinate pairs before hidden averaging. At fixed population coefficients, the resulting hidden moments enter both the residual drift and source covariance linearly, with weight $\Delta\tau = 1/L$. The local weak operator on the augmented forward/backward histories therefore has the expansion

$$\begin{aligned} \mathcal{T}_\ell(\mu_\ell) &= I + \Delta\tau\, \mathcal{L}_\ell(\mu_\ell) + O(\Delta\tau^2), \\ D^2_{\mu_\ell}\mathcal{L}_\ell &= 0, \qquad D^2_{\mu_\ell}\mathcal{T}_\ell = O(\Delta\tau^2). \end{aligned} \tag{171}$$

The expansion uses the covariance, not a particular square root of the Gaussian noise. Two hidden-sampling insertions therefore carry two depth weights. Their propagation includes changes in the kernels and loss feedback through (154). This counts the effect of a slice on the full history; it does not require a small pointwise fluctuation of a fixed-width hidden block.

At second order in this population-response expansion, centered sampling leaves the same-sample contractions

$$\mathbb{E}O - O_{\mathrm{pop},L} = \frac{b_\flat}{N_\flat} + \frac{1}{N_\sharp L^2}\sum_{\ell=0}^{L-1} b_\ell + \text{higher-order terms}, \tag{172}$$

where $O_{\mathrm{pop},L}$ is the finite-depth population value. The coefficients include the response of the observable itself and the contraction of a sampling fluctuation with the response derivative computed from that same sample; the latter need not vanish. Bounded propagated coefficients give the sampling-bias scales ${N_\flat}^{-1}$ and $(N_\sharp L)^{-1}$. Together with the depth error, these give $|b| \lesssim L^{-1} + {N_\flat}^{-1} + K^{-1}$ at the retained orders. Combining this mean bias with the two variance contributions yields

$$E_{\mathrm{SGD}} \lesssim \frac{1}{\sqrt{N_\flat}} + \frac{1}{L} + \frac{1}{\sqrt{N_\sharp L}}. \tag{173}$$

The depth term estimates the mean of the selected statistic. It is distinct from a trajectorywise strong-error estimate.

For Adam, include its first and second moment states and express the stabilizer in each group's gradient units. When these dynamics satisfy the same regularity, propagation, and covariance conditions,

$$E_{\mathrm{Adam}} \lesssim \frac{1}{\sqrt{N_\flat}} + \frac{1}{L} + \frac{1}{\sqrt{N_\sharp L}}. \tag{174}$$

These conditions concern Adam's own dynamics. Gaussianity is a separate question from the control of these errors.

Spectral Muon instead transforms an empirical hidden Gram matrix before its contribution is averaged over depth. Its local operator is already nonlinear in μ_ℓ at order $\Delta\tau$, so the same-sample contraction contributes $\Delta\tau/N_\sharp$ per slice and $1/N_\sharp$ after summing over depth. The smooth local calculation (163) identifies a mean bias of order ${N_\sharp}^{-1} + {N_\flat}^{-1}$. When the propagated spectral contributions retain this mean order, their depth average also has this order. Their conditionally centered parts enter H, while fluctuations of their conditional means enter S. Combining these terms with the same depth and covariance control gives

$$E_{\mathrm{Muon}} \lesssim \frac{1}{\sqrt{N_\flat}} + \frac{1}{L} + \frac{1}{\sqrt{N_\sharp L}} + \frac{1}{N_\sharp}. \tag{175}$$

The residual-side bias $1/N_\flat$ is covered by $1/\sqrt{N_\flat}$, and

$$\frac{1}{\sqrt{N_\sharp L}} \le \frac{1}{2}\left(\frac{1}{N_\sharp} + \frac{1}{L}\right). \tag{176}$$

This gives the three-term Muon estimate in the main text. The additional spectral bias vanishes in the fully linear case (162).

D.4 Regrouping the same hidden population

The main-text pairing example preserves the channel budget exactly. For even L, put ${N_\sharp}' = 2N_\sharp$ and $L' = L/2$, so ${N_\sharp}'L' = K$, and stack the projection matrices as

$$\widetilde{\boldsymbol{W}}^{r\uparrow} = \begin{pmatrix} \boldsymbol{W}^{2r\uparrow} \\ \boldsymbol{W}^{(2r+1)\uparrow} \end{pmatrix}, \quad \widetilde{\boldsymbol{W}}^{r\downarrow} = \begin{pmatrix} \boldsymbol{W}^{2r\downarrow} & \boldsymbol{W}^{(2r+1)\downarrow} \end{pmatrix}. \tag{177}$$

Both hidden populations now read the residual state at the start of the pair, and their outputs add with the unchanged coefficient $\beta_0\sqrt{N_\flat}/K$. This is exactly a block of width $2N_\sharp$. Compared with the original sequential network, the second population reads the state one residual update earlier, as in the main-text Euler interpretation. SGD and entrywise Adam updates commute with stacking the paired computation, since their code-coordinate rates and gradient units depend on $N_\flat, K$; corresponding Adam moments are stacked as well. A nonlinear spectral transform acts on each whole block and need not commute with stacking. Its finite-$N_\sharp$ bias is the additional contribution identified above.

The population coefficients give the same Euler interpretation. Replacing $(N_\sharp, L)$ by $(2N_\sharp, L/2)$ leaves $N_\flat/K$ and the hidden reaction coefficient $\beta_0 N_\flat/K$ unchanged. The residual drift coefficients and source covariances double, as required for a doubled depth step. For example, source independence gives

$$\begin{aligned} \mathrm{Cov}(u^{(2r+1)\flat} + u^{(2r+2)\flat}) &= \frac{\beta_0^2 N_\flat}{KL}(\Phi^{2r\sharp} + \Phi^{(2r+1)\sharp}) \\ &= \frac{\beta_0^2 N_\flat}{KL'}\Phi^{2r\sharp} + O(L^{-2}), \qquad L' = L/2, \end{aligned} \tag{178}$$

where the last equality uses regular depth variation of the population kernels. The backward-source covariance obeys the same calculation with $G^\sharp$. The comparison concerns the Gaussian sources; trained residual increments also contain the learning feedback.

Consider two regroupings with the same $N_\flat, K$, optimizer, data, training step, and limiting coefficients. Let their depths be L_1, L_2, with hidden widths $N_{\sharp,j} = K/L_j$, and compare the same macroscopic statistic O_j in each network. Both approximate the common value $o_\infty(N_\flat/K)$. For any coupling of their initializations, including reuse of the same ordered hidden computations, the triangle inequality gives

$$\begin{aligned} \|O_1 - O_2\|_2 &\le \|O_1 - o_\infty(N_\flat/K)\|_2 + \|O_2 - o_\infty(N_\flat/K)\|_2 \\ &\lesssim {N_\flat}^{-1/2} + K^{-1/2} + L_1^{-1} + L_2^{-1} \qquad \text{(SGD or Adam)}. \end{aligned} \tag{179}$$

For smooth spectral Muon, add $N_{\sharp,1}^{-1} + N_{\sharp,2}^{-1}$. This bound does not require or assert trajectory-wise agreement of individual residual coordinates. It compares macroscopic statistics within one optimizer, and does not exploit any additional cancellation from coupling the two finite networks.

E HALF COPY: A CORRELATED-INITIALIZATION RECIPE

Half Copy restores an initial residual-variance scale by correlating nearby channels, while keeping the network geometry, coupling constants, and learning-rate scalings unchanged. The construction has two choices: how many channels share a parent, and how strongly their down-projections are correlated. The four-channel example in Eq. 7 is one instance.

E.1 LAYER-LOCAL GROUPS AND GAUSSIAN WEIGHTS

Consider one channel family in Eq. 13, with total population K and a desired variance multiplier $s \geq 1$. For fixed coupling, preserving an anchor's coefficient $\beta_0^2 N_\flat/K$ requires

$$s = \frac{(N_\flat/K)_{\text{anchor}}}{N_\flat/K}. \tag{180}$$

Growing both widths and depth by R therefore gives $s = R$. Input and readout initialization are left unchanged.

Partition consecutive channels within each layer into disjoint groups $\mathcal{G}_g$ of integer sizes c_g, and let $p(a)$ identify the group containing channel a. In mathematical-weight coordinates, initialize

$$\begin{aligned} \boldsymbol{W}_a^{\uparrow} &= \boldsymbol{U}_{p(a)}, \\ \boldsymbol{W}_a^{\downarrow} &= \sqrt{\rho_{p(a)}}\,\boldsymbol{Z}_{p(a)} + \sqrt{1-\rho_{p(a)}}\,\boldsymbol{E}_a, \qquad 0 \leq \rho_g \leq 1. \end{aligned} \tag{181}$$

Parent arrays $\boldsymbol{U}_g, \boldsymbol{Z}_g$ and innovations $\boldsymbol{E}_a$ have independent standard Gaussian entries and are mutually independent. Different groups and layers use independent draws. For an MLP channel, these arrays are an up-projection row and down-projection columns; their code weights absorb the coefficients in Eq. 1. Each entry retains its standard Gaussian marginal.

E.2 MATCHING VARIANCE WITH INTEGER GROUPS

Channels sharing a parent see the same incoming state and initially produce the same activation $A_g(\boldsymbol{x})$. For one residual coordinate, their summed down-weight has variance

$$\operatorname{Var}\left(\sum_{a\in\mathcal{G}_g} W_{ia}^{\downarrow}\right) = c_g\big[1 + (c_g - 1)\rho_g\big]. \tag{182}$$

Thus, conditional on the incoming states and up-projections, the initial residual increment $\Delta h_i^\flat$ has covariance

$$\begin{aligned} &\operatorname{Cov}_{\downarrow}\big(\Delta h_i^\flat(\boldsymbol{x}), \Delta h_i^\flat(\boldsymbol{x}')\big) \\ &\quad = \frac{\beta_0^2 N_\flat}{K^2} \sum_{g \text{ in layer}} c_g\big[1 + (c_g - 1)\rho_g\big] A_g(\boldsymbol{x}) A_g(\boldsymbol{x}'). \end{aligned} \tag{183}$$

Choosing

$$\rho_g = \frac{s-1}{c_g - 1}, \qquad c_g \geq s, \quad c_g > 1, \tag{184}$$

makes every group's variance multiplier exactly s. After averaging over the up-projection parents, this multiplies the branch covariance by s without changing the single-channel activation law. The residual variance coefficient becomes $\beta_0^2(N_\flat/K)s$. For $s = 1$, use $\rho_g = 0$; singleton groups need no correlation.

Integer sizes need not divide the layer width. For a layer with m channels, choose a number G of parent groups, set $c = \lfloor m/G \rfloor$ and $r = m - Gc$, and use r groups of size $c + 1$ and $G - r$ groups of size c. Choose $1 \leq G \leq \lfloor m/\lceil s \rceil \rfloor$, so that $c \geq s$, then apply Eq. 184 separately to the two sizes. This matches the target exactly despite the uneven partition. Taking $c > s$ keeps $\rho_g < 1$ and leaves independent innovations; for example, group sizes near λs with fixed $\lambda > 1$ leave $1 - \rho_g$ of order one. If a layer has fewer than s channels, this layer-local construction cannot supply the requested enhancement.

Alternatively, fixing ρ first makes the ideal common group size $c = 1 + (s-1)/\rho$. At $\rho = 1/3$, this is $c = 3s - 2$: $c = 4$ when $s = 2$, and approximately $3R$ under conventional growth. For balanced integer groups with fixed ρ, the realized layerwise multiplier is

$$\chi_\ell = 1 + \rho \, \frac{\sum_{g \text{ in layer}} c_g(c_g - 1)}{m}. \tag{185}$$

The CIFAR and Tiny Shakespeare growth experiments use this fixed-ρ version, recording the remaining integer mismatch rather than retuning ρ. Their task-specific choices are given in Appendices G.2 and H.

E.3 Attention, locality, and learning

For attention, group whole heads. Copy the Q, K, and V up-maps within each group, using independent parents for the three maps, and apply Eq. 181 to their output-projection matrices. The grouped heads then compute identical attention outputs at initialization. In Eq. 183, products of scalar activations become inner products of the head outputs, with the head-dimension normalization already included in ϕ_{A}. MLP units and attention heads have separate populations and groups; tokens and coordinates within a head are not grouped as extra channels.

Under conventional growth, group sizes are $\Theta(R)$ while $K = \Theta(R^2)$. The total number of independent parent groups therefore grows as $K/R = \Theta(N_\flat)$, and each group's fraction of the ordered hidden population vanishes. Groups stay within a single layer, so correlated copies read the same residual state; copying across layers would not obey the same finite-network covariance identity.

Copying is an initialization operation, not parameter tying: materialize every matrix separately and use separate optimizer states with the rates of Table 1. When $\rho_g < 1$, distinct down-projections give the copied up-projections distinct backward signals, allowing them to separate from the first update (Appendix F.4). The matching calculation fixes the initial variance contribution; it does not identify the subsequent learning dynamics with those of independently initialized fixed-ratio networks.

F Controlled checks of theoretical predictions

These controlled experiments test residual distribution shapes, the one-step checkpoint, spectral sampling error, and the separation of initially copied features. Protocols and quantitative records are collected here, with the distributional and checkpoint figures alongside the theoretical arguments.

F.1 A nonlinear logic task with actual backward propagation

The inputs are $00, 01, 10, 11$, in that order; the output coordinates are AND, OR, and XOR. Put $n = N_\flat = N_\sharp$ and augment the input as $x' = (x_1/\sqrt{2}, x_2/\sqrt{2}, 1)$. The stored matrices compute

$$\begin{aligned} h &= Ex', & z &= Uh/\sqrt{n}, \\ H &= h + V\tanh(z)/\sqrt{n}, & f &= A^{\mathsf{T}}H/n. \end{aligned} \tag{186}$$

The three columns of E have independent Gaussian entries of standard deviation 0.01; U, V, A have standard Gaussian entries. Thus the constant input implements a trained input bias. There are no other biases, normalization, dropout, or weight decay. The objective is half the mean squared error over the twelve targets; binary predictions use threshold $1/2$. Widths are $64, 128, 256, 512, 1024$, with seeds 100001–100008. Each optimizer runs 1000 full-batch updates in CPU float64. Optimizers share their initial arrays, and widths use nested prefixes within each seed. All parameter groups update simultaneously.

For explicit optimizer units, let Γ be the stored loss gradient multiplied by n for E, A, or by $n\sqrt{n}$ for U, V. SGD applies rate 0.1 to Γ, with an additional $1/\sqrt{n}$ for U, V. Adam uses rate 0.03 and the same group factors, $(\beta_1, \beta_2) = (0.9, 0.999)$, bias correction, and $\epsilon = 10^{-8}$ in these normalized-gradient coordinates.

The spectral comparison uses the odd polynomial $p(s) = 15s/8 - 5s^3/4 + 3s^5/8$, composed five times:

$$F(\Gamma) = \frac{\lambda_n}{(15/8)^5} p^{\circ 5}(\Gamma/\lambda_n). \tag{187}$$

The matrix polynomial acts on singular values, preserving singular vectors. Its outer learning rate is four times the SGD rate above. For each group, the fixed bound is 1.25 times the maximum SGD singular value across all planned widths, seeds, and updates, after dividing spectra by $\sqrt{n}$ for E, A and by n for U, V. The resulting dimensionless bounds, in this order, are 0.7609752890, 0.1501675784, 0.06461736165, and 0.08052997111. They are multiplied back by $\sqrt{n}$ or n to obtain λ_n. The largest observed singular-value/bound ratio is 0.839; bound violations would stop a run rather than change its recipe. Exact low-rank evaluation retains all four input contributions and performs no numerical rank truncation. The pooled calibration couples widths and seeds: equivariance of a fixed-bound update does not alone establish rotational invariance after conditioning on this entire calibration procedure.

The external gain was selected before adding the input bias, using learning progress and domain safety rather than Gaussianity. That earlier small-input control is informative: strong spectral compression at unit gain left all eight largest-width seeds short of solving the task, with mean loss 0.0428. Gain four resolved this slow-learning failure. A separate standard-input control changed the update direction by roughly 50–60% in the four parameter groups relative to the best scalar multiple of the same SGD gradient. Spectral processing therefore need not be an almost-linear update, but these differently calibrated methods are not a matched-cost optimizer benchmark.

We record h, H, A, and the actual rescaled Jacobian $g_q^{0\flat} = n\,\partial f_q/\partial h$ at updates $0, 1, 2, 4, 8, 16, 32, 64, 128, 256, 512, 1000$. The derivative includes the nonlinear block and the identity path; it excludes the loss derivative. For a chosen pair, each coordinate vector is centered and divided by its own empirical standard deviation within each run. With the resulting z_i, covariance C, and empirical average over residual coordinates, we calculate

$$\kappa_{ijkl} = \langle z_i z_j z_k z_l \rangle - C_{ij}C_{kl} - C_{ik}C_{jl} - C_{il}C_{jk},$$
$$Q_4 = \left(\frac{1}{16} \sum_{i,j,k,l=1}^{2} \kappa_{ijkl}^2 \right)^{1/2}. \tag{188}$$

We summarize eight run-level values, without pooling seeds or inputs. Residual coordinates are not treated as independent experimental replicates. The plotted scatter uses the predetermined seed 100001, width 1024, and update 1000; it is centered only, without whitening or rotation. If shown, covariance ellipses are Gaussian reference contours, not confidence regions for the observed samples.

All 120 runs across the five widths, eight seeds, and three optimizers classify all twelve targets correctly. The joint distributions are illustrated in Fig. 10 in Appendix B.8.4. Table 3 reports the RMS of standardized joint fourth cumulants across eight seeds, including forward pairs from two inputs.

Residual pair	SGD	Taylor	Adam
$h^{0\flat}(11), g^{0\flat}_{\mathrm{AND}}(11)$	$.0601 \pm .0293$	$.0687 \pm .0327$	$.6109 \pm .0172$
$h^{0\flat}(11), h^{0\flat}(00)$	$.0667 \pm .0216$	$.0712 \pm .0217$	$1.1086 \pm .1195$
$h^{1\flat}(11), h^{1\flat}(00)$	$.0790 \pm .0327$	$.0692 \pm .0303$	$.3654 \pm .0591$

Table 3: Joint fourth-cumulant RMS after 1000 updates at width 1024, reported as mean $\pm$ sample standard deviation across eight seeds.

The complementary two-layer linear control removes the block and input bias, using $h = E(x_1, x_2)/\sqrt{2}$, $f = A^{\mathsf{T}}h/n$, and the same loss, input standard deviation, optimizer units, eight seeds, and 1000 updates. Its widths are $256, 1024, 4096, 16384, 65536$. Both SGD and Adam approach the linear loss floor $1/12$ and fail XOR. At the largest width, their excess kurtoses for $h(11)$ are -0.0044 ± 0.0279 and -1.9412 ± 0.0015, respectively. SGD's variance grows from approximately 10^{-4} to 1.21, so its Gaussian appearance does not result from frozen features. An exact finite-dimensional SGD recurrence matches its trajectory to 1.09×10^{-14}. The separate one-step linear residual calculation, including its finite-width discrepancies, is checked in Appendix F.2.

Raw mixed moments across initializations. Gaussian-shaped coordinate clouds need not have deterministic covariance. We also examine raw mixed inner products in the same logic-task

archive, using all five widths and eight seeds. At each of updates 0 and 1000 we collect $h(00), h(11), H(00), H(11), g^{0\flat}_{\mathrm{AND}}(11), g^{0\flat}_{\mathrm{XOR}}(11), A_{\mathrm{AND}}, A_{\mathrm{XOR}}$, where h, H are the residuals before and after the block and A_q is the rescaled final-residual output Jacobian. Chronological concatenation gives $Y \in \mathbb{R}^{n\times 16}$. We retain $Y^{\mathsf{T}}Y/n$ and the coordinate means without centering, whitening, or run-specific rescaling. All 2,880 matching-time readout values $H^{\mathsf{T}}A/n$ reproduce the archived predictions exactly.

Across widths 64–1024, the RMS of the 64 entrywise seed standard deviations in the final-time Gram block decreases from 0.1376 to 0.0398 for SGD, 0.1462 to 0.0407 for Taylor, and 0.1925 to 0.0596 for Adam. The corresponding RMS magnitudes of the seed-mean entries change from 1.2431 to 1.2366, 1.2497 to 1.2285, and 1.6173 to 1.6325. Cross-time block dispersions also decrease over this range. These summaries do not imply monotonic improvement: for example, the SGD final-residual second-moment SD increases from 0.0727 at width 512 to 0.0809 at width 1024, while the aggregate final-time dispersion barely changes over that last doubling (0.0403 to 0.0398). The accompanying data retain all entries, coordinate means, and seed-paired adjacent-width differences; the predefined scalar summaries include forward/backward, cross-input, cross-output and cross-time pairs.

The observations support increasing reproducibility of these raw moments over the tested widths, including for non-Gaussian Adam features. They do not establish convergence to a deterministic Gram matrix, a concentration rate, or a whole-process limit. Standard deviations use seeds, not neurons, as replicates; widths are paired by their nested initializations. Taylor remains descriptive because its calibration pools these same widths and seeds.

F.2 Numerical verification of the linear checkpoint

The simulations implement (114) in NumPy float64 on a CPU, retaining all $L+1$ residual states. Each realization uses an independent Gaussian initialization and one simultaneous SGD step for the squared loss $\frac{1}{2}(f-1)^2$. All finite-step runs in this subsection use $\eta_{\mathrm{eff}} = \gamma_0$. The input embedding can be sampled directly as $\boldsymbol{h}^{0\flat} \sim \mathcal{N}(0, I_{N_\flat})$, which has exactly the same distribution as the frozen normalized Gaussian embedding in (108).

Implementation checks. PyTorch automatic differentiation verified both parameter gradients and the full updated forward pass. At $N_\flat = 128$, $N_\sharp = 8$, $L = 32$, and $\beta_0 = \gamma_0 = 1$, the maximum absolute gradient-update discrepancy was 7.3×10^{-16} and the final-state discrepancy was 7.2×10^{-15}. A second check at $(N_\flat, N_\sharp, L, \beta_0, \gamma_0) = (64, 8, 16, 0.7, 1.3)$ gave discrepancies below 7.8×10^{-16} and 3.2×10^{-15}, respectively. No input or readout gradient was applied.

The five-moment recursion was checked separately by constructing $\boldsymbol{M}_j$ directly in $50{,}000$ Gaussian realizations at $(N_\flat, N_\sharp, L, \beta_0) = (8, 3, 4, 0.7)$. All five terminal means were within 0.79 Monte Carlo standard errors of (123). Across the finite-network runs with at least 32 realizations, the measured tangent energies agreed with (120) to within 2.10 standard errors. These comparisons test the implementation and the finite-size moment identities separately.

Depth and shape conventions. Residual state $\boldsymbol{h}^{j\flat}$ is assigned depth j/L. The plots include both endpoints, $j = 0, \ldots, L$; a trajectory containing only $\boldsymbol{h}^{0\flat}, \ldots, \boldsymbol{h}^{(L-1)\flat}$ would end at $(L-1)/L$, not at 1. All theoretical predictions use the realized ratio $N_\flat/(N_\sharp L)$. For example, at $N_\flat = 1024, L = 128$, requested ratios $1.25, 1.5, 2.5$ yield $N_\sharp = 6, 5, 3$ after integer rounding, corresponding to actual ratios $4/3, 1.6, 8/3$. The large-ratio discrepancies reported below remain after accounting for the realized dimensions.

Finite-step measurements. Table 4 reports the energy after the full finite update. The uncertainty is one Monte Carlo standard error of the mean. All rows use $L = 128$, $\beta_0 = 1$, and $\eta_{\mathrm{eff}} = \gamma_0$. Figure 12 in Appendix C.2.9 shows representative complete depth profiles, including the actual ratio $4/3$. Its left panel uses 128 seeds per curve; the right uses 256, with $(N_\flat, N_\sharp) = (2048, 16)$ at $\alpha = 1$ and $(1024, 6)$ at $\alpha = 4/3$.

Separating two finite-size effects. At $\alpha = 2, L = 128, \gamma_0 = 1$, the exact tangent energies for $N_\flat = 1024, 2048, 4096$ are respectively $1523.90, 1510.75, 1504.24$, while the continuum tangent

$N_\flat$	$N_\sharp$	$N_\flat/K$	γ_0	Seeds	Full-step $C_h(1;1)$	Limit (131)
1024	16	0.5	1	128	18.63 ± 0.28	19.32
1024	16	0.5	2	128	71.21 ± 2.10	72.32
2048	16	1	1	256	83.06 ± 1.40	84.00
1024	6	$4/3$	1	256	233.83 ± 9.35	228.38
1024	4	2	1	128	2876.95 ± 423.02	1645.33
2048	8	2	1	256	2206.47 ± 250.25	1645.33
4096	16	2	1	256	1972.17 ± 96.10	1645.33

Table 4: Residual energy after one finite SGD step for the squared loss. The continuum formula is not asserted to be exact at these sizes. In particular, the large-α discrepancy remains visible.

contribution is 1637.94. Thus a depth correction is already present before the finite-step remainder is considered. At $N_\flat = 4096$, increasing L from 128 to 256 while keeping $\alpha = 2$ changes the exact tangent prediction to 1572.84, closer to its continuum value. These deterministic comparisons use the exact finite-size recurrence, not a fit to the Monte Carlo curves.

The full step has a further remainder, measured directly by (150). At the three widths above, its means are $1142.45,\ 679.59,\ 317.05$. This substantial contribution decreases over the tested width range; we do not infer a uniform asymptotic rate from three noisy points. Figure 13 displays both effects. The correction to residual energy is not equal to $\mathcal{R}_L$: cross terms and the random factor $1 - f_0$ also contribute.

For a milder depth check with $N_\flat = 1024, \alpha = 0.5, \gamma_0 = 1$, the exact tangent energies at $L = 64, 128, 256$ are $17.2195, 17.4740, 17.6032$; the continuum value is 17.6688. The corresponding sampled tangent means are $17.2566, 17.2774, 17.6069$. This comparison distinguishes deterministic finite-size corrections from the sampling fluctuations in a full-step depth sweep.

Finally, the general-β_0 formula was checked at $(N_\flat, N_\sharp, L, \beta_0, \gamma_0) = (1024, 16, 128, 0.7, 1.3)$ with 256 seeds. The full-step terminal energy was 8.479 ± 0.074, compared with 8.629 from (130). The finite-network mean remains slightly below the continuum prediction.

F.3 Coupled scalar tests of spectral sampling error

A sampled kernel isolates spectral processing from network dynamics. For independent uniform coordinates on $[0, 1]$, use $\psi_j(y) = \sqrt{2}\sin(2\pi jy)$, $\varphi_j(x) = \sqrt{2}\cos(2\pi jx)$, and $s_j = (1+(j/8)^2)^{-5/8}$, giving an infinite square-summable tail. At any finite spectral truncation, put $Y_{aj} = \psi_j(y_a)$ and $X_{ij} = \varphi_j(x_i)$. With $S = \mathrm{diag}(s_j)$ and m, n coordinates, set $M = YSX^\mathsf{T}/\sqrt{mn}$, $P = Y^\mathsf{T}Y/m$, and $Q = X^\mathsf{T}X/n$. Let F_c apply $s \mapsto \min(s/c, 1)$ to singular values. With coupled mode-one test vectors $v = Ye_1/\sqrt{m}$ and $w = Xe_1/\sqrt{n}$, compare the scalars

$$Z = v^\mathsf{T}F_c(M)w - e_1^\mathsf{T}PF_c(S)Qe_1. \tag{189}$$

Both terms use the same sampled coordinates. This comparison remains meaningful when matrix sizes change; it does not subtract unlike matrices.

At $c = 0.5$, the population spectrum has a positive gap from the clipping threshold. Sampling the hidden side with seeds 718000–718127, using nested prefixes at widths $512, 1024, 2048, 4096$ and keeping the residual side at its population value gives mean biases $-0.004095, -0.001977, -0.001001, -0.000509$. The analytic leading coefficient is -2.0323153, rather than a fitted amplitude. The fitted exponents of absolute bias and centered standard deviation are -1.001 and -0.480, respectively. Their 95% bootstrap intervals are $[-1.031, -0.969]$ and $[-0.562, -0.403]$, resampling seed identities jointly across widths. The mean estimator subtracts the zero-mean control variate $-\frac{1}{2}(P_{11} - 1)$; fluctuations use the original Z. The paired bootstrap uses 500 resamples with seed 230017. At fixed $m = 128, n = 256$, averaging $L = 1$–64 independent blocks in 64 independent sequences reduces centered dispersion with fitted exponent -0.539 in L, while leaving mean bias near -0.0244. Both coordinate samples are redrawn between blocks, unlike shared network coordinates.

Two controls delimit the inverse-width interpretation. In the entirely linear region, paired error and regrouping differences are at numerical roundoff (5.6×10^{-17} and 1.1×10^{-16}). At an exact

clipping kink, a rank-one example instead gives mean-bias exponent -0.501. Thus the smooth-region bias should not be extrapolated through the kink or equated with a universal trained-network error. The kernel is evaluated using a 2^{22}-point Fourier table with linear interpolation; refinement to 2^{24} changes the tested actions by at most 5.6×10^{-7}. Seeds, scalar statistics, and reference coefficients are retained in the accompanying data.

F.4 A CONTROL FOR SEPARATING INITIALLY COPIED FEATURES

A ReLU residual-MLP control uses $D_{\mathrm{i}} = 12$, $D_{\mathrm{o}} = 10$, $L = 3$, no biases or normalization, $\beta_0 = 2.25$, and $\gamma_0 = 1$. Float32 SGD on mean cross-entropy uses standard-coordinate rates $0.002N_{\flat}$ for input/readout and $0.002K$ for both branch matrices. With $N_{\flat} = 32, N_{\sharp} = 8$, nine standard-normal inputs and labels $0, \ldots, 8$, groups of four identical up-projections receive down-projection correlation $1/3$; parameters remain untied. After subtracting each group's mean, the up-gradient RMS is 8.98095×10^{-3}; one update raises the corresponding up-weight RMS from zero to 4.31086×10^{-4}.

A separate exact-duplication check copies both projections twice within each block ($N_{\flat} = 16, N_{\sharp} = 4 \to 8$), with the compensating doubling of K and branch learning rates. On seven Gaussian inputs and labels $0, \ldots, 6$, logits match within 9.54×10^{-7} after three updates. Both use initialization seed 7319; input seeds are 73190091 and 73190090, respectively. The first check demonstrates separation of initially copied features. The second checks compensated exact duplication, not equivalence between correlated and independent initialization laws.

G CIFAR-10 RESIDUAL-MLP EXPERIMENTS

The CIFAR-10 (Krizhevsky, 2009) experiments below separate convergence, scaling recipes, and learning-rate selection. They use different normalization and learning-rate settings, specified for each comparison.

G.1 CIFAR-10 GROWTH AND COUPLED GROUPING

The network takes 3072 flattened pixel values mapped to $[-1, 1]$ and returns ten logits. It uses the four-projection parametrization of (1), ReLU, Gaussian initialization, and cross-entropy loss. There is no augmentation, bias, normalization, momentum, weight decay, or learning-rate schedule. In standard-Gaussian coordinates, SGD uses $\eta_0 \gamma_0^2 N_{\flat}$ for input/readout and $\eta_0 \gamma_0^2 K$ for up/down matrices, with $\eta_0 = 0.02$, $\gamma_0 = 1$, and $\beta_0 = 2.25$. Training uses PyTorch (Paszke et al., 2019) with MPS float32, batch size 64, and ten epochs. Keeping the final batch of 16 gives 782 updates per epoch. The implementation specifies a common shuffled minibatch sequence; initialization seeds are separate from data order. The older checkpoints do not retain individual batch-index arrays or per-run source revisions.

The growth family uses $N_{\flat} = q^2$ and $N_{\sharp} = L = q$. The tested values are $q = 8, 10, \ldots, 24$ and $q = 28, 32, 40, 64, 80, 96$, with 32 distinct initialization seeds per size, listed in the data export. Logit RMSE is computed on the first ten test images per class, over all 100×10 output entries. Each model is compared with the mean output of the 32 $q = 96$ models at the same epoch; distances are then averaged. Figure 5 shows twice the standard error of these run-level distances, conditional on this finite reference. The $q = 96$ self-reference is excluded when fitting size dependence. Its individual-to-mean RMSE is 0.129 at epoch one and 1.272 at epoch ten, which measures finite reference dispersion, not an asymptotic error floor. Full training/test losses and accuracies use all 50,000/10,000 images, rather than this probe. The kernel panel uses the final residual states on the first ten training images per class. Their uncentered Gram matrix is normalized to unit diagonal, giving a cosine kernel. Each run's entrywise kernel RMSE is measured against the mean of the 32 $q = 96$ kernels at the same epoch. The means and twice-standard-error bars are computed across runs, as for the output errors.

Unweighted least-squares fits of log error against $\log q$ over all 14 plotted sizes ($q = 8$–80) give output/kernel slopes of $-0.808/-0.820$ at epoch one and $-0.289/-0.320$ at epoch ten. Over the larger half of these sizes, $q \in \{22, 24, 28, 32, 40, 64, 80\}$, the corresponding slopes are $-0.948/-$

Table 5: Adjacent regroupings of each coupled pool. All selected cases are shown; epoch zero records the initial disagreement.

		$18 \to 36$		$36 \to 72$		$72 \to 144$	
q	Epoch	R	S	R	S	R	S
24	0	0.0173	0.0023	0.0136	0.0002	0.0106	0.0016
24	1	0.2480	0.0628	0.2073	-0.0209	0.1866	0.0369
24	10	1.1092	-0.1466	1.0145	0.1718	0.9123	-0.1930
36	0	0.0124	-0.0005	0.0100	0.0002	0.0082	-0.0010
36	1	0.2118	0.0927	0.1550	-0.0097	0.1498	-0.0211
36	10	1.4441	0.1931	1.3345	0.2075	1.3254	-0.0408

0.896 and $-0.230/-0.262$. These are descriptive fits to finite-reference errors. The q^{-1} guides in Fig. 5 indicate the theoretical exponent; their vertical offsets are chosen for readability.

The grouping study fixes $N_\flat = K = q^2$ for $q = 24, 36$, using all 21 and 25 divisor depths, respectively. Both families use seed 101234. Each family shares its input/output matrices and one ordered pool of hidden weights, partitioned into consecutive groups of size K/L. It has one coupled realization, not an ensemble of independent depths. Changing the grouping changes the residual states at which subsequent groups are evaluated. The $L = K$ model is the finest finite reference; its zero distance to itself is omitted. Only $q = 24$ appears in Fig. 5 (right); both sizes are retained in the numerical supplement. At fixed $N_\flat$ and K, the population-sampling terms in Theoretical Result 3 do not decrease with depth, so this comparison is not assigned a pure L^{-1} reference slope. Early output agreement improves towards this reference, but later agreement can flatten or be nonmonotone. Likewise, raw residual Grams retain amplitude changes removed by cosine normalization. At epoch one, $q = 36$, the mean raw Gram diagonal at $L = 1, 36, 72, 1296$ is $2.56, 64.84, 66.41, 65.37$. Thus neither every added layer nor every diagnostic improves monotonically.

Direct comparisons of adjacent groupings. Distances to one reference do not determine how close two other models are to each other. We therefore also compare the saved raw logits $F_{q,L,e} \in \mathbb{R}^{100\times 10}$ directly at $L = 18, 36, 72, 144$ and epochs $e = 0, 1, 10$ for both sizes. These choices were fixed before this additional extraction from the completed runs. For $\Delta F = F_{q,2L,e} - F_{q,L,e}$, we report

$$R = \left(\frac{1}{1000} \sum_{a=1}^{100} \sum_{c=1}^{10} \Delta F_{a,c}^2 \right)^{1/2}, \qquad S = \frac{1}{100} \sum_{a=1}^{100} \left(\Delta F_{a,y_a} - \frac{1}{9} \sum_{c \neq y_a} \Delta F_{a,c} \right), \tag{190}$$

where y_a is the saved test-probe label. The signed contrast S is a fixed linear test of the output change; its sign does not determine a change in accuracy or cross-entropy. The raw-logit discrepancy R also counts offsets common to all classes.

The three successive RMS discrepancies decrease within each size and epoch, but their scale grows substantially during training (Table 5). These absolute differences do not separate changes in logit amplitude from sensitivity to grouping. At epoch ten, $q = 36$ has larger discrepancies than $q = 24$, and its final refinement makes little further difference to their magnitude. The signed contrasts change direction repeatedly, so a small average contrast cannot stand in for agreement of all outputs. These comparisons describe local sensitivity to grouping without choosing the finest partition as a limiting output. They remain one realization per size, with dependent pairs, and do not estimate an ensemble bias, centered fluctuations, or a convergence rate. The data export retains all 24 output arrays and 18 difference arrays.

G.2 CIFAR-10 GROWTH RECIPES FROM A COMMON ANCHOR

The CIFAR panel in Fig. 6 uses the same flattened inputs and ten-class cross-entropy as the convergence experiment, but uses non-affine RMSNorm (Zhang & Sennrich, 2019) before each ReLU

branch: $\widehat{h} = h/\sqrt{\|h\|^2/N_\flat + 10^{-6}}$. The identity path is unchanged; there are no biases, augmentation, momentum, weight decay, or learning-rate schedule. Its anchor has $(N_\flat, K, L) = (181, 362, 16)$, $\beta_0 = 0.5625$, $\gamma_0 = 1$, and endpoint/residual learning-rate multipliers $2.56/1.28$ in the stored-Gaussian convention: the actual SGD rates are $\eta_{\text{end}}\gamma_0^2 N_\flat$ for input/readout and $\eta_{\text{res}}\gamma_0^2 K$ for up/down matrices. There is no additional 0.02 factor in this experiment. Three seeds, 100001–100003, share the minibatch order. Training uses MPS float32 and batches of 64, dropping the last batch. Each run completes five epochs (3,905 SGD updates); the final cross-entropy is evaluated on a fixed 10,000-example training subset. Figure 6 reports the mean and sample standard deviation across these three seeds.

Write the anchor dimensions as (N_0, K_0, L_0). At budget index j, the ideal ODE geometry is $(N_0 2^{j/3}, K_0 2^{2j/3}, L_0 2^{j/3})$, whereas the fixed-ratio geometry is $(N_0 2^{j/2}, K_0 2^{j/2}, L_0 2^{j/4})$. Both have residual-block parameter budget $2N_0 K_0 2^j$. Integer widths are distributed across layers to realize the recorded total K; comparisons use actual budgets, not stage names. The early ODE/semi-freeze/Half Copy point has $(N_\flat, K, L) = (362, 1448, 32)$, whereas the early fixed-ratio point is $(724, 1448, 32)$, with twice the block parameters. The subsequent six stages likewise use different budget indices for the two geometries; exact dimensions and indices accompany the data. The extra largest fixed-ratio point is omitted from the main panel but included in the full-range Fig. 14. Total trainable parameters are $2N_\flat K + 3082N_\flat$. ODE and the fixed-ratio recipe keep their rate multipliers fixed. Semi-freeze uses the ODE geometry, multiplies β_0 by $\sqrt{R_\alpha}$, and divides only the residual rate multiplier by R_α, where $R_\alpha = (N_0/K_0)/(N_\flat/K)$; endpoint rates remain fixed. Half Copy keeps the ODE geometry and rates and uses the layer-local construction in Appendix E. Its up-projections are identical within each group; down-projections have correlation $1/3$, and all parameters remain independently trainable. With group sizes c_g across all layers, the initial variance enhancement is $1 + \sum_g c_g(c_g - 1)/(3K)$. The integer layer-local partitions target R_α; the supplement records their realized enhancement and group sizes. The older recipe that also reduces endpoint rates is not used in this figure.

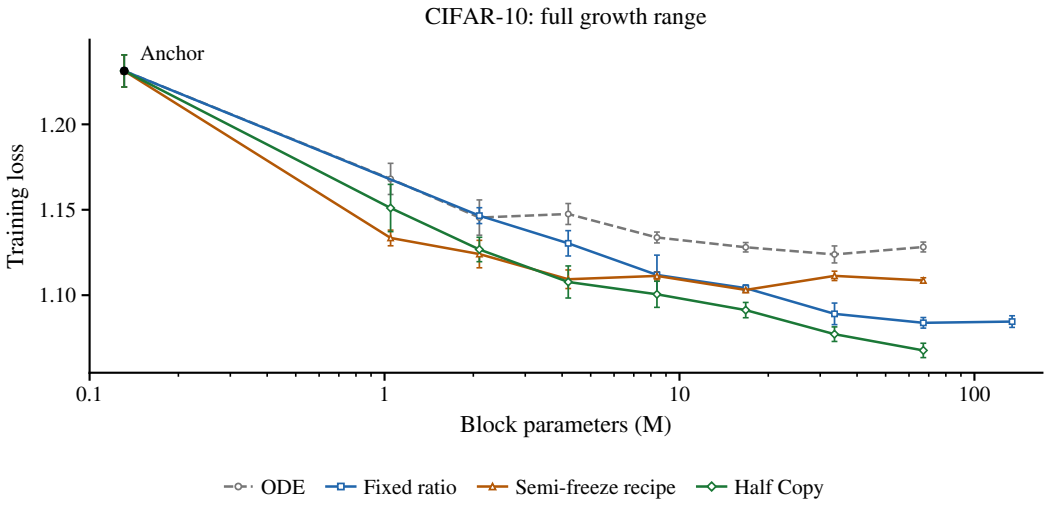


Figure 14: CIFAR-10 growth recipes over the full measured range, including the shared anchor and early points. Final training cross-entropy follows five SGD epochs; bars are sample standard deviations over three seeds. All recipes use their actual block parameter counts, including the extra largest fixed-ratio point omitted from Fig. 6.

The full-range curves illustrate the reversal of an initially favorable semi-freeze comparison; an initial advantage is not required. In a separate common-anchor comparison with $\beta_0 = 0.28125$ and endpoint/residual multipliers $1.28/1.28$, semi-freeze has higher four-seed mean training loss than the fixed-ratio recipe at all twelve measured budgets $j = 3, \ldots, 14$. The endpoint pairs are 1.19993 versus 1.19249 and 1.22033 versus 1.14292, respectively. This record combines three audited historical seeds with a fourth seed; one individual seed at the smallest budget slightly favors semi-freeze.

Learning-rate transfer and local retuning. A separate 114-run scan tests the older anchor $\beta_0 = 0.28125$, $\gamma_0 = 1$, with $\eta_{\text{end}} = \eta_{\text{res}} = 1.28m$. It uses the same five-epoch pre-RMS protocol, three seeds, and budget indices $j = 0, 7, 14$, but does not reuse the plotted growth runs. The candidate multipliers are $m = 2^{k/2}$ for $k = -2, \ldots, 2$ at $j = 0$, $k = 0, \ldots, 4$ at $j = 7$, and $k = 0, \ldots, 5$ at $j = 14$. The anchor is shared by ODE, fixed-ratio, and Half Copy recipes. All selection uses final training loss.

The anchor-selected multiplier is $\sqrt{2}$. The grid minimizers at $j = 7$ are $2\sqrt{2}, 2, 2$ for ODE, fixed ratios, and Half Copy, respectively; at $j = 14$ they are $2\sqrt{2}, 2\sqrt{2}, 2$. All lie inside their search windows. Transferring $\sqrt{2}$ instead of retuning costs 0.008–0.016 in mean training cross-entropy across the six larger-scale comparisons. Local retuning preserves the ordering Half Copy, fixed ratios, ODE from lower to higher loss at both sizes (Fig. 15). These finite grids assess transfer accuracy, not a continuous optimal learning rate; their anchor and endpoint-rate convention differ from Fig. 6.

The figure also includes an earlier coarse scan of ODE, fixed ratios, and semi-freeze at the same anchor, seeds, budgets, and five-epoch protocol. It uses $m = 2^k$, $k = -4, \ldots, 3$, with 168 physical runs on A100; the 114-run refinement above was performed on MPS. The figure shows one curve per recipe, using the refined measurements where grid points overlap and the earlier scan elsewhere, without pooling seeds. For semi-freeze, m scales the endpoint rate and the prescribed reduced residual rate together. Within the coarse scan, its grid minimum moves from 1 at the anchor to 4 at $j = 7$ and 8 at $j = 14$, where it is still at the search boundary. Transferring its anchor-selected $m = 1$ costs 0.052 and 0.073 in mean training loss relative to these measured minima. Thus semi-freeze provides a case where the prescribed rate scaling requires appreciably more retuning.

H Tiny Shakespeare Transformer experiments

H.1 Model, training, and evaluation

The growth experiments use a causal character-level Transformer (Vaswani et al., 2017) on the Tiny Shakespeare text distributed with `char-rnn` (Karpathy, 2015), with a 65-character vocabulary and a contiguous $90/10$ split. This series evaluates only training windows. Each of three seeds, 100001–100003, specifies initialization, training-window sampling, and fixed evaluation windows. The training and evaluation streams use seed offsets $+1$ and $+2$. The same seed identifies paired streams across recipes and sizes. Consequently, variation across seeds includes data sampling as well as initialization; it is not initialization-only uncertainty. Each run uses 5,000 Adam updates, effective batch size 32, sequence length 128, and float32 arithmetic, processing 20,480,000 training positions. Loss is evaluated on 16 fixed batches of 32 training sequences at step 1 and every 500 updates. All uncertainty reported here is sample standard deviation over three seeds, not standard error.

Write $n = N_{\text{b}}$, let $N_{\text{b},\ell}$ denote the MLP width and $N_{\text{h},\ell}$ the head count in block ℓ, and set

$$K_{\text{b}} = \sum_{\ell=0}^{L-1} N_{\text{b},\ell}, \qquad K_{\text{h}} = \sum_{\ell=0}^{L-1} N_{\text{h},\ell}.$$

The head dimension is $d_{\text{h}} = 16$. Each block applies causal attention and then a GELU MLP (Hendrycks & Gimpel, 2016), with a separate non-affine pre-RMSNorm (Zhang & Sennrich, 2019) before each branch, using $\epsilon = 10^{-6}$. Queries and keys use RoPE (Su et al., 2024), with frequencies 2^{-j}, $j = 0, \ldots, 7$. Attention logits are divided by $\tau\sqrt{d_{\text{h}}}$, with $\tau = 1$. There are no biases, dropout, weight decay, gradient clipping, or learning-rate schedule. The token embedding and output weights are untied. A final non-affine RMSNorm precedes the linear readout. For stored standard-Gaussian weights, all up-maps have factor $1/\sqrt{n}$, the MLP residual branch has factor $\beta_{\text{b}}\sqrt{n}/K_{\text{b}}$, the attention output branch has factor $\beta_{\text{h}}\sqrt{n}/(K_{\text{h}}\sqrt{d_{\text{h}}})$, and the readout has factor $1/(\gamma n)$. Here $\gamma = 1$; d_{h} is kept separate from the head-count ratio.

The core and total trainable parameter counts are

$$P_{\text{core}} = 2nK_{\text{b}} + 4d_{\text{h}}nK_{\text{h}}, \qquad P_{\text{total}} = P_{\text{core}} + 2(65)n.$$

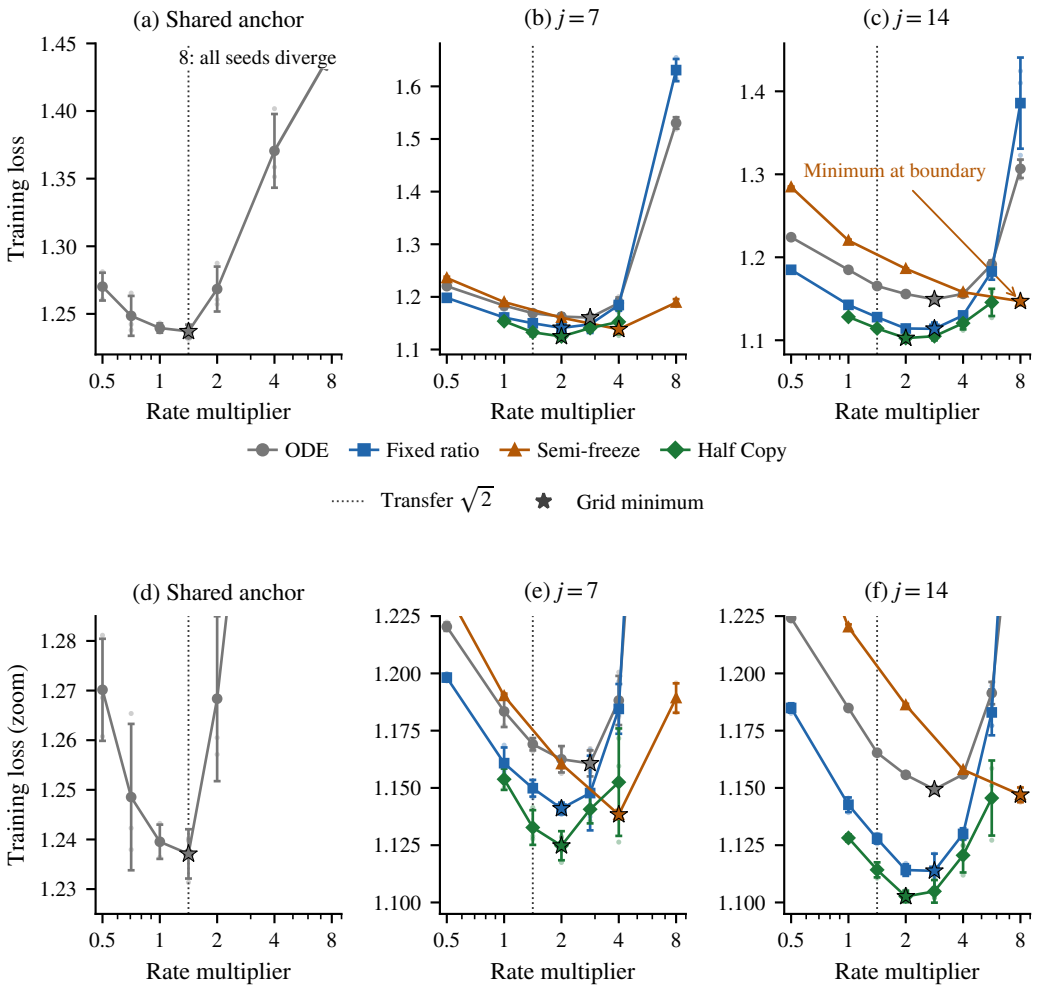


Figure 15: CIFAR-10 learning-rate scans, shown for $m \geq 1/2$. Top: scan overview. Bottom: the same data with tighter vertical limits to resolve the basin bottoms; all six panels use identical horizontal axes. Colors identify recipes. Small points show three seeds; bars show sample standard deviations after five epochs. Stars mark grid minima; the semi-freeze minimum at $j = 14$ is not bracketed. The dotted line marks the anchor-selected $\sqrt{2}$. The shared anchor is drawn once. The anchor arrow at $m = 8$ marks three divergent runs, not a finite loss.

Parameter counts below are realized counts, not nominal targets. The anchor has $(n, N_{\mathrm{b}}, N_{\mathrm{h}}, L) = (128, 256, 4, 4)$, giving $K_{\mathrm{b}} = 1024$, $K_{\mathrm{h}} = 16$, and $P_{\mathrm{core}} = 393216$. Its branch coefficients are $\beta_{\mathrm{b}} = 1.6771809884309676$ and $\beta_{\mathrm{h}} = 0.35617040537656264$.

H.2 Calibration and eight optimizer groups

Initial branch energy provides a calibration observable without requiring a diversity score. Let $\Delta_{\mathrm{b},\ell}$ and $\Delta_{\mathrm{h},\ell}$ be the MLP and attention additions to the residual stream at one token. We average the following quantities over tokens and fixed training windows:

$$Q_{\mathrm{b}} = \sum_{\ell} \frac{\|\Delta_{\mathrm{b},\ell}\|^2}{n}, \quad Q_{\mathrm{h}} = \sum_{\ell} \frac{\|\Delta_{\mathrm{h},\ell}\|^2}{n}, \quad Q = Q_{\mathrm{b}} + Q_{\mathrm{h}}, \quad S_0 = \frac{\|h^0\|^2}{n}.$$

The anchor calibration uses five seeds and eight batches, with no weight updates. A target $Q/S_0 = 0.3$ gives measured mean 0.325306 and an attention-to-MLP energy ratio of 1.16695. A stronger

Parameter group	Stored-weight Adam rate	Gradient unit s_j
Token embedding	$\eta_{\mathrm{in}}\gamma$	$1/(\gamma n)$
Readout	$\eta_{\mathrm{out}}\gamma$	$1/(\gamma n)$
MLP up	$\eta_{\mathrm{bulk}}\gamma\beta_{\mathrm{b}}/\sqrt{n}$	$\beta_{\mathrm{b}}/(\gamma K_{\mathrm{b}}\sqrt{n})$
MLP down	$\eta_{\mathrm{bulk}}\gamma\beta_{\mathrm{b}}/\sqrt{n}$	$\beta_{\mathrm{b}}/(\gamma K_{\mathrm{b}}\sqrt{n})$
Attention Q	$\eta_{\mathrm{bulk}}\gamma\beta_{\mathrm{h}}/\sqrt{n}$	$\beta_{\mathrm{h}}/(\gamma K_{\mathrm{h}}\sqrt{n})$
Attention K	$\eta_{\mathrm{bulk}}\gamma\beta_{\mathrm{h}}/\sqrt{n}$	$\beta_{\mathrm{h}}/(\gamma K_{\mathrm{h}}\sqrt{n})$
Attention V	$\eta_{\mathrm{bulk}}\gamma\beta_{\mathrm{h}}/\sqrt{n}$	$\beta_{\mathrm{h}}/(\gamma K_{\mathrm{h}}\sqrt{n})$
Attention output	$\eta_{\mathrm{bulk}}\gamma\beta_{\mathrm{h}}/\sqrt{n}$	$\beta_{\mathrm{h}}/(\gamma K_{\mathrm{h}}\sqrt{n})$

Table 6: The eight Adam groups. The compensated-coupling recipe changes the bulk constants and branch coefficients as specified below.

target of 3 gives 5.02973 and a branch ratio of 2.30879, failing the prescribed balance check. These tests select a usable energy scale, not a performance optimum. A subsequent training-loss search adjusts bulk, input, and output rates. The selected anchor has five-seed loss 1.269145 ± 0.010729; none of the six tested axial neighbors passes the local improvement criterion. The endpoint search reaches an output-rate boundary, so this is not a global hyperparameter optimum.

All experiments in the comparison use Adam (Kingma & Ba, 2015) with moments $(0.9, 0.999)$ and its usual bias correction. Table 6 specifies the eight groups in stored Gaussian coordinates. The constants are $\eta_{\mathrm{in}} = 0.064$, $\eta_{\mathrm{out}} = 0.256$, and $\eta_{\mathrm{bulk}} = 0.38159268702343885$. For each group, the denominator constant is transported as $\epsilon_j = 10^{-8} s_j / s_{j,0}$, where $s_{j,0}$ is its anchor value. There is no additional factor $128/n$ beyond this ratio. This deterministic transport is part of the recipe; it does not assert that every gradient coordinate has identical scaled statistics.

The ratios $\epsilon_j/\sqrt{\widehat{v}_j}$ are recorded at updates 1, 100, 1000, and 5000 on up to 8192 stratified coordinates per group. This check motivated replacing an earlier fixed-ϵ continuation; its results are not pooled with the transported-ϵ series here. Earlier development runs also used learned positional embeddings and held-out evaluation; they are a different protocol.

H.3 Growth recipes and correlated initialization

The main growth series uses $R = 4, 5, 6.25, 8, 10, 12.75$. The ODE route has $(n, N_{\mathrm{b}}, N_{\mathrm{h}}, L) = (128R, 256R, 4R, 4R)$, with fixed branch and rate constants. Thus $n/K_{\mathrm{b}} = 1/(8R)$ and $n/K_{\mathrm{h}} = 8/R$. For the fixed-ratio route, let q_R be the nearest integer to $16R^{3/2}$ and choose $(n, K_{\mathrm{b}}, K_{\mathrm{h}}, L) = (8q_R, 64q_R, q_R, 4R)$. Balanced floor/ceiling allocations preserve these totals when per-block counts are not integers. This keeps $(n/K_{\mathrm{b}}, n/K_{\mathrm{h}}) = (1/8, 8)$. Compensated coupling keeps the ODE geometry but multiplies both branch coefficients by $\sqrt{R}$ and divides the six bulk η constants by R; endpoint constants stay fixed. Its initialization compensates the falling $\beta^2 n/K$ scale. These experiments do not directly measure whether individual features freeze, and we make no such inference from the recipe's construction.

Half Copy keeps the ODE geometry, branch coefficients, and rates, using the construction in Appendix E. The growth series fixes $\rho = 1/3$ and uses layer-local balanced groups targeting $\chi = R$ separately for MLP units and attention heads. For each family, its reported enhancement is the channel-weighted average of Eq. 185 over layers. Integer group sizes cause small departures: at $R = 6.25$, the realized uniform attention enhancement is 6.1712.

Figure 16 also includes the common $R = 1$ anchor and three earlier $R = 2$ points, all with the same three seeds and 5,000-update protocol. At $R = 2$, ODE and semi-freeze use $(n, K_{\mathrm{b}}, K_{\mathrm{h}}, L) = (256, 4096, 64, 8)$; the fixed-ratio point uses $(512, 4096, 64, 8)$. Thus this earlier fixed-ratio point has twice their block parameters; it is plotted at its actual budget, not treated as an equal-budget instance of the allocation above. There is no $R = 2$ Half Copy endpoint. The anchor's initial Q is reconstructed using the pinned original model code and evaluation windows, without training.

All 72 runs in this six-scale, four-recipe, three-seed comparison complete 5,000 updates. The initial-energy comparison includes the anchor and early points as well as the five main scales through $R = 10$; we do not supply an initial-energy value for $R = 12.75$. Initial Q is a sum of squared increments, not the squared total residual displacement. For all branch increments collected as Δ_j,

		ODE / compensated / Half Copy			Fixed ratios		
R	L	n	K_{h}	P_{core} (M)	n	K_{h}	P_{core} (M)
4	16	512	256	25.165824	1024	128	25.165824
5	20	640	400	49.152000	1432	179	49.214976
6.25	25	800	625	96.000000	2000	250	96.000000
8	32	1024	1024	201.326592	2896	362	201.283584
10	40	1280	1600	393.216000	4048	506	393.271296
12.75	51	1632	2601	815.007744	5824	728	814.055424

Table 7: Actual growth geometry and core parameter counts. Every row has $K_{\mathrm{b}} = 64K_{\mathrm{h}}$. Add $130n$ for total parameters; the two geometry routes therefore need not have identical total parameter counts even when their core counts match.

$\|\sum_j \Delta_j\|^2/n = Q + \sum_{j\neq k}\langle\Delta_j, \Delta_k\rangle/n$. Accordingly, Q does not measure cross-layer covariance or all sources of feature diversity.

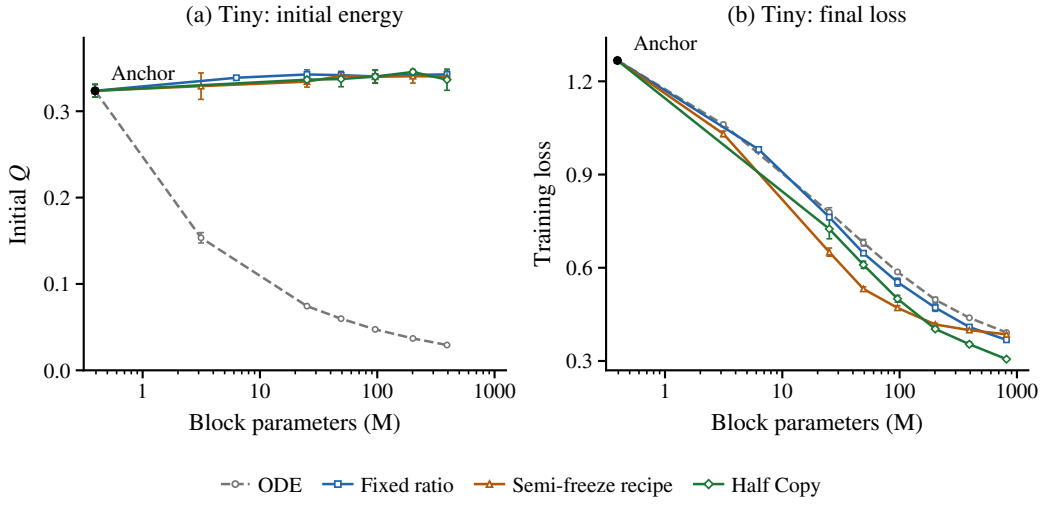


Figure 16: Tiny Shakespeare growth recipes over the full measured range, including the shared anchor and available early points. Left: initial summed squared residual increments Q. Right: training loss after 5,000 Adam updates. Bars are sample standard deviations over three seeds. Half Copy has no $R = 2$ run; losses include $R = 12.75$, where initial Q was not measured. Figure 6 focuses on $R \geq 4$.

H.4 Cross-task parameter savings at comparable training loss

The parameter savings quoted in the Introduction compare the final three-seed mean training losses in Fig. 6, after five SGD epochs for CIFAR-10 and 5,000 Adam updates for Tiny Shakespeare. We count all trainable parameters, including input and output maps: $P = 2N_{\flat}K + 3082N_{\flat}$ for CIFAR-10 and $P = P_{\mathrm{core}} + 130n$ for Tiny Shakespeare. For an ODE reference with count P_{ODE}, the saving is $1 - P_{\mathrm{alt}}/P_{\mathrm{ODE}}$, where the alternative matches or improves its mean loss under the same task-specific training schedule.

For Tiny Shakespeare SDE, the matching sizes lie between sampled networks. With neighboring counts $P_a < P_b$ and mean losses $\bar{\ell}_a > \bar{\ell}_{\mathrm{ODE}} > \bar{\ell}_b$, we interpolate locally:

$$w = \frac{\log \bar{\ell}_a - \log \bar{\ell}_{\mathrm{ODE}}}{\log \bar{\ell}_a - \log \bar{\ell}_b}, \quad P_{\mathrm{alt}} = P_a^{1-w} P_b^{w}. \tag{191}$$

No extrapolation is used. At ODE reference scales $R = 10$ and 12.75, this gives savings of 28.1% and 34.7%, respectively. Interpolating loss linearly against either $\log P$ or P instead gives 27.2% or 23.1% at $R = 10$, and 34.1% or 29.6% at $R = 12.75$. For CIFAR-10, a direct comparison needs no

R	$L = 1$	$L = 2$	$L = 3$	$L = 4$
4	0.426116	0.595115	0.677284	0.689741
8	0.257038	0.339469	—	0.382781

Table 8: Final fixed-training-window cross-entropy in the separate low-depth scan. The dash denotes no completed endpoint in this evidence set.

Recipe	$L = 40$	$L = 4$	$R = 12.75,\ L = 51$
ODE route	0.4389 ± 0.0003	0.4085 ± 0.0067	0.3916 ± 0.0010
Fixed ratios	0.4092 ± 0.0058	0.3704 ± 0.0085	0.3680 ± 0.0025
Compensated coupling	0.3993 ± 0.0054	0.3688 ± 0.0059	0.3857 ± 0.0025
Half Copy	0.3539 ± 0.0069	0.3362 ± 0.0082	0.3059 ± 0.0038

Table 9: Final training loss, mean and sample standard deviation over three seeds. The first two columns are paired at $R = 10$; the last column is a larger growth experiment, not part of that pair.

interpolation: SDE uses 12.850M parameters with mean loss 1.1119, versus ODE's 19.581M and 1.1281, a 34.4% reduction.

Half Copy also admits direct comparisons on both tasks. On CIFAR-10, 10.618M parameters give loss 1.1005, below the same 19.581M ODE reference, using 45.8% fewer parameters. On Tiny Shakespeare, Half Copy at $R = 10$ uses 393.382M parameters with loss 0.3539, compared with 815.220M and 0.3916 for ODE at $R = 12.75$, using 51.7% fewer parameters. All three paired seeds favor the smaller network in each of these direct comparisons, including the CIFAR SDE comparison. The introductory 20% and 40% figures round down these estimates and comparisons at the specified larger reference sizes. They describe final mean training loss under these schedules, not confidence bounds or guarantees at every size or seed; for example, one paired Tiny SDE estimate at $R = 10$ is 18.8% under linear loss–parameter interpolation.

H.5 Width–depth allocation at fixed capacity

One to four blocks at fixed capacity. A separate Half Copy scan probes shallower allocations at two fixed budgets. For $R = 4$ and 8, respectively, $(n, K_{\mathrm{b}}, K_{\mathrm{h}}) = (512, 16384, 256)$ and $(1024, 65536, 1024)$, with head dimension 16. The total parameter counts are 25,232,384 and 201,459,712. All runs use the same Tiny Shakespeare split and Adam recipe described above: 5,000 updates, sequence length 128, effective batch 32, and seed 100001. Final cross-entropy averages 512 fixed training sequences, not held-out data. MLP units and attention heads are allocated evenly across blocks, up to integer rounding. This scan rebuilds the original layer-local correlated groups at every depth, with $\rho = 1/3$. It therefore changes their joint initialization law as well as depth. Among the seven completed points, fewer blocks give lower training loss at each budget; each point uses a single seed. The original 16-block model at $R = 4$ uses the same layer-local rule, parameter budget, and training protocol. Its seed-100001 loss is 0.724028, completing the $1, 2, 4, 16$-layer comparison in Fig. 8. These four plotted values evaluate the terminal model on the fixed training windows; they are not individual optimization-minibatch losses.

Paired four-block and deeper controls. For each recipe at $R = 4, 5, 6.25, 8, 10$, a four-block model preserves $n, K_{\mathrm{b}}, K_{\mathrm{h}}$, both parameter counts, and the training protocol of its original $L = 4R$ counterpart. All 60 paired comparisons yield lower final training loss with four blocks (Fig. 17). Table 9 gives the largest paired scale. The networks are initialized and trained separately. For Half Copy, regrouping can also move correlated groups across layer boundaries; its comparison does not isolate depth from all changes in the joint initialization law. The other three recipes retain independent Gaussian initialization.

Additional deeper allocations at $R = 4, 5, 6.25$ comprise 36 completed runs. At $R = 5$, increasing depth from 20 to 100 raises mean loss from 0.6805 to 0.6889 for the ODE route, from 0.6469 to 0.6555 for fixed ratios, from 0.5319 to 0.5526 for compensated coupling, and from 0.6096 to 0.6324 for Half Copy. The fixed-ratio deeper model here has 1.12% more core parameters due to rounding;

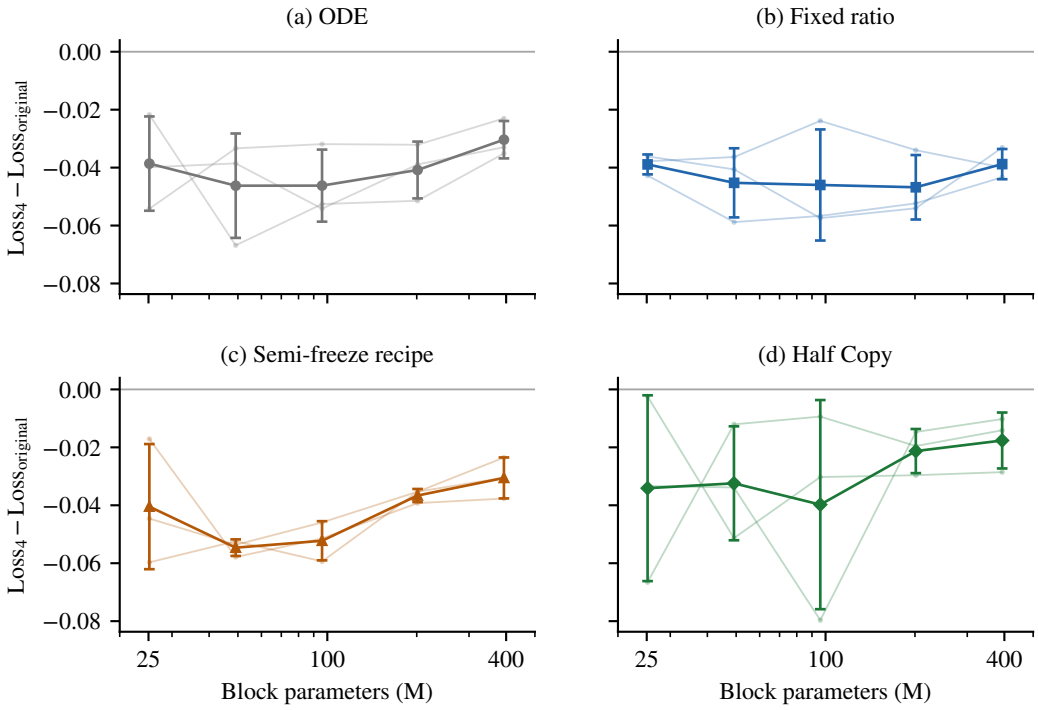


Figure 17: Paired depth comparisons on Tiny Shakespeare at five budgets ($R = 4, 5, 6.25, 8, 10$). Each value is four-block minus original-depth final training loss. Faint lines show the three paired seeds; thick lines and bars show their mean and sample standard deviation. All sixty differences are negative. The separate $R = 12.75$ extension is not included.

at $R = 6.25$ the discrepancy is -1.59%. The direction is not universal: compensated coupling at $R = 4$ gives 0.6486 for the deeper allocation and 0.6503 for the original. These finite controls support a task- and recipe-specific depth effect, not a universal ordering by depth.

H.6 Depth-integrated residual kernels

Depth-integrated residual kernels supply a separate diagnostic of the trained networks. These measurements use the 5,000-update checkpoint: the integration is over network depth, not training time. On 16 common training texts at token positions 31, 63, 95, and 127, let H^ℓ be the $64 \times n$ residual matrix after block ℓ and before the next normalization. For each seed we form the raw kernels

$$\Phi^\ell = \frac{H^\ell (H^\ell)^\mathsf{T}}{n}, \qquad \overline{\Phi} = \frac{1}{L}\sum_{\ell=1}^{L} \Phi^\ell.$$

No diagonal normalization or division by branch strength is applied. The input kernel is recorded separately. Across the three seeds, relative Frobenius dispersion of $\overline{\Phi}$ decreases from 14.9% to 10.1% between $R = 4$ and $R = 10$ for the ODE route, from 15.0% to 10.6% for fixed ratios, from 13.4% to 8.9% for compensated coupling, and from 17.3% to 15.3% for Half Copy. Here relative dispersion is $\sqrt{\mathbb{E}_s\|\overline{\Phi}_s - \mathbb{E}_s\overline{\Phi}_s\|_F^2}/\|\mathbb{E}_s\overline{\Phi}_s\|_F$, with the numerator estimated using sample variance. Cross-layer covariance accounts for approximately 93–95% of the entrywise variance at $R = 10$. Layers cannot be treated as independent replications, and three seeds do not establish a fluctuation exponent. Figure 18 also retains kernel amplitude through the mean diagonal. Half Copy has the lowest training loss among these recipes at $R = 10$ but the largest relative kernel dispersion; the two statistics therefore give different recipe rankings. Across-seed dispersion is distinct from random variation along depth within a network.

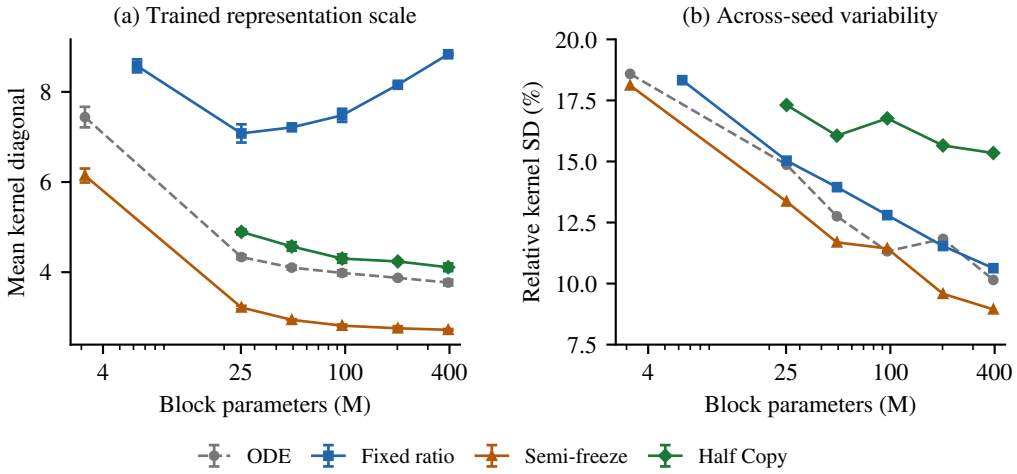


Figure 18: Raw residual kernels after 5,000 Tiny Shakespeare updates. Left: mean diagonal of the depth-integrated kernel, with three-seed sample standard deviations. Right: relative Frobenius dispersion across those seeds. For the integer grouping used in this experimental series, Half Copy measurements begin at $R = 4$.

R	Seed	Recipe	Uniform	Width allocation	Depth mesh
8	100001	ODE	0.506104	0.477521	—
8	100001	Half Copy	0.397821	0.380109	—
8	100002	ODE	—	0.483269	0.488810
8	100002	Half Copy	—	0.402959	0.388330
10	100001	ODE	—	0.432979	0.435738
10	100001	Half Copy	—	0.339870	0.337943

Table 10: Exploratory nonuniform allocations, each trained for 5,000 updates. Dashes indicate comparisons not included here, not failed training. The second $R = 8$ seed contributed to allocation selection.

H.7 Exploratory nonuniform allocations

Nonuniform width and nonuniform depth meshes implement different finite networks. The width experiment keeps $n, L, K_{\mathrm{b}}, K_{\mathrm{h}}$ fixed while allocating more MLP units and heads to selected depths. Writing $d_\ell = N_{\mathrm{h},\ell}/(K_{\mathrm{h}}/L)$, with MLP counts proportional to head counts, the local residual branch and its gradient-unit epsilon are multiplied by $1/d_\ell$. Stored-weight learning rates are unchanged. The depth-mesh experiment instead keeps per-block capacity fixed and uses intervals δ_ℓ summing to one, multiplying the same branch and epsilon factors by $L\delta_\ell$. For Half Copy, maintaining the corresponding initialization variance can exceed what layer-local head groups can supply. At $R = 8$, the finest requested interval requires enhancement 27.6283, whereas 32 heads with $\rho = 1/3$ allow at most $1 + 31/3 = 11.3333$ within one layer. The implemented mesh therefore allows groups across two or three adjacent layers and matches the weighted covariance over each such group of layers; MLP groups remain layer-local. This exploratory construction differs from the uniformly spaced, layer-local growth recipes above. Table 10 reports completed training losses; each row is a single seed, not a three-seed estimate.

These allocations were guided by measured activity profiles, not an identified SDE drift. A centered-window estimator initially used to interpret those profiles failed an Ornstein–Uhlenbeck control: it returned approximately -0.840 for a window-averaged squared drift of 0.854. Its near-correct constant-drift control therefore did not validate that interpretation. The trained models and losses remain valid finite experiments, but the allocation rule is an exploratory heuristic. Neither these selected runs nor the branch-energy profiles establish a general benefit from moving channels toward any particular depth.

I GPT-2 PROTOCOLS AND RESIDUAL MEASUREMENTS

The AdamW shape and energy experiments and the all-NS5 Muon experiment use distinct architectures and training protocols, detailed below. They are not a matched optimizer comparison.

Sources of the theoretical illustration. Figure 3 combines selected examples from these two experiments. On the left, coordinate 6 of the twenty-four-stage AdamW model is tracked at all 25 residual boundaries for validation text 0, token 1023, at initialization and after 15,000 updates. On the right, the all-NS5 model at update 4500 pairs validation texts 3 and 6 at block 1, token 1023. All 768 post-block residual coordinates and all 3072 pre-GELU hidden coordinates are retained. Each marginal is centered and standardized separately, without rotation, whitening, or tail removal. The examples were selected for qualitative illustration; the systematic distributional measurements below retain the predefined texts and slices. The covariance-matched Gaussian ellipses are 95% contours, not confidence intervals or contours forced to enclose 95% of the data.

I.1 ADAMW MODELS AND FINITE REGROUPING

The base implementation of GPT-2 (Radford et al., 2019) is `build-nanogpt` (Karpathy, 2024), commit `6104ab1b53920f6e2159749676073ff7d815c1fa`. It has twelve attention–MLP blocks with pre-LayerNorm (Ba et al., 2016), residual width 768, twelve heads of dimension 64, MLP width 3072, approximate GELU, learned positional embeddings, and a final LayerNorm. Biases and affine LayerNorm parameters are retained. The token embedding and output matrix are tied; the vocabulary is padded to 50,304 entries. There is no dropout. All five constructions have 124,475,904 parameters.

The one-, two-, and four-stage constructions group 12, 6, and 3 consecutive original blocks, respectively. Within each group, the attention branches, each with its own LayerNorm, read the same incoming state. Their outputs are summed and added once. The MLP branches then read this updated state and their outputs are summed and added once. No averaging coefficient is introduced. The twenty-four-stage model instead divides each original block into two serial attention–MLP stages, each with six heads and 1536 MLP units. Matrix slices retain their original parameters; the stages share their parent block's LayerNorm parameters, and each contributes half its original down-projection bias. Up-projection biases are partitioned. Parameter tensors, optimizer groups, and numerical initial arrays are unchanged, although the initial functions differ.

All runs use seed 1337 and retain the initialization and optimizer settings of Karpathy (2024). Embeddings and ordinary linear weights have Gaussian standard deviation 0.02; attention and MLP down-projections use $0.02/\sqrt{24}$ in every construction. Biases begin at zero and LayerNorm gains at one. Fused AdamW (Loshchilov & Hutter, 2019) uses $(\beta_1, \beta_2) = (0.9, 0.95)$, $\epsilon = 10^{-8}$, and weight decay 0.1 on matrices and embeddings, with no decay on biases or LayerNorm. The gradient norm is clipped at 1. Learning rate increases linearly over 715 updates to 6×10^{-4}, then follows the original cosine schedule toward 6×10^{-5}. Each update accumulates eight microbatches of 64 sequences of length 1024: 524,288 tokens. The one-, twelve-, and twenty-four-stage runs use one A100-SXM4-80GB. The two-stage run resumes from its 10,341-update A100 checkpoint on an H100; the four-stage run uses an H100 throughout. All use PyTorch 2.8.0, the CUDA 12.8 build, and bfloat16 autocasting, without `torch.compile`.

The data are `HuggingFaceFW/fineweb-edu` (Lozhkov et al., 2024), configuration `sample-10BT`, revision `87f09149ef4734204d70ed1d046ddc9ca3f2b8f9`. Ordered preprocessing uses the GPT-2 tokenizer (`tiktoken` 0.11.0), inserting an end-of-text token before each document. The first 100-million-token shard is held out; the remaining 99 contain 9,853,989,344 training tokens. Identical shard checksums and sequential loaders give the same training sequence, cycling when it is exhausted. Each run completes 19,073 updates, consuming 9,999,745,024 tokens. Validation precedes the final update, so the reported endpoint is after 19,072 updates and 9,999,220,736 tokens. Every evaluation resets to the same held-out prefix and averages twenty microbatches, or 1,310,720 tokens.

Figure 8 instead reports training loss, averaged over the final 100 optimization updates in every construction (zero-based log steps 18,973–19,072). Each logged value averages the eight gradient-accumulation microbatches before the corresponding parameter update; the plotted mean therefore

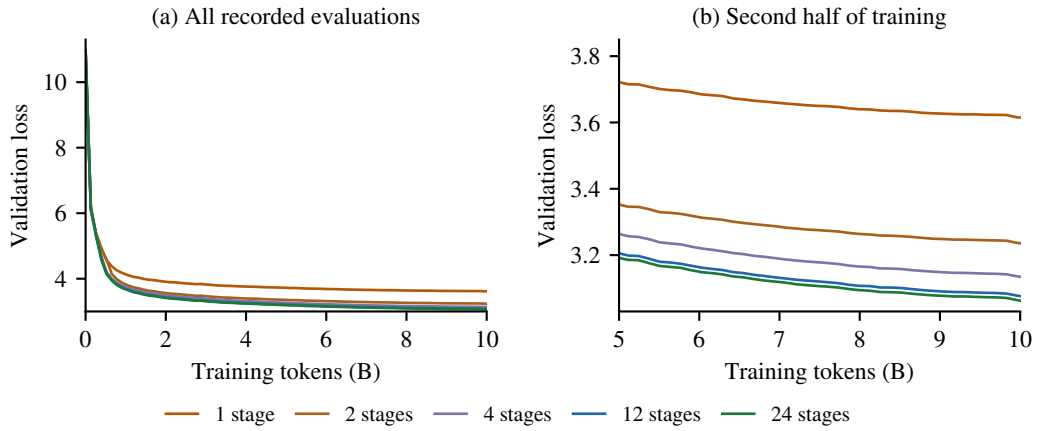


Figure 19: Complete validation trajectories for the five AdamW depth allocations, all with 124,475,904 parameters and seed 1337. Left: all 78 recorded evaluations per model, including initialization. Right: an enlarged view of the second half of training. Lines connect the saved evaluations without smoothing; values retain the four-decimal precision of the logs.

covers 52,428,800 training tokens along the last 100 updates, rather than evaluating one frozen terminal model. For 1, 2, 4, 12, and 24 stages, these means are respectively 3.588404, 3.219410, 3.120185, 3.062130, and 3.049850. Both panels thus use training data, but the Tiny Shakespeare panel uses the fixed-window terminal evaluation described in Appendix H.5; their estimators and tokenizations differ. The held-out metrics in Table 11 retain their original validation definition. Figure 19 shows all recorded validation evaluations, including initialization, with a second panel resolving the later training differences.

HellaSwag uses all 10,042 validation questions without demonstrations (Zellers et al., 2019). The candidate ending with the lowest mean autoregressive token loss over its completion is selected. The CE and HellaSwag orderings of the twelve- and twenty-four-stage models differ; neither single-seed ordering establishes a significant advantage.

Construction	Stages	Validation CE	HellaSwag
Parallel regrouping	1	3.6148	0.2808
Parallel regrouping	2	3.2359	0.2927
Parallel regrouping	4	3.1347	0.2942
Original	12	3.0764	0.3091
Split blocks	24	3.0627	0.3061

Table 11: Completed GPT-2 comparisons at the common evaluation endpoint.

I.2 Energy in common residual coordinates

We evaluate four held-out 1024-token sequences beginning at offsets 0, one million, two million, and three million in the validation shard. Forward hooks record the true residual after every complete attention–MLP stage, before the next normalization, in CPU float32. The twenty-four-stage model has 24 measured stages despite retaining twelve parameter containers. Initialization is reconstructed in the original PyTorch environment and its parameter-array hash checked against the training initialization.

For each token, use $N_\flat = 768$ and $\Delta \boldsymbol{h}_\ell = \boldsymbol{h}_{\ell+1} - \boldsymbol{h}_\ell$. Define $E_0 = \|\boldsymbol{h}_0\|^2/N_\flat$, $Q = \sum_\ell \|\Delta \boldsymbol{h}_\ell\|^2/N_\flat$, and $C = \|\sum_\ell \Delta \boldsymbol{h}_\ell\|^2/N_\flat - Q$. The full accounting is

$$\frac{\|\boldsymbol{h}_L\|^2}{N_\flat} = E_0 + Q + C + \frac{2\langle \boldsymbol{h}_0, \sum_\ell \Delta \boldsymbol{h}_\ell \rangle}{N_\flat}. \tag{192}$$

We compute the quadratic quantities token by token before averaging. Write $B_0 = 2\langle \boldsymbol{h}_0, \sum_\ell \Delta \boldsymbol{h}_\ell \rangle / N_\flat$; the identity is $E_L = E_0 + Q + C + B_0$. The 4096 tokens are descriptive probes, not independent replicates.

Stages	Updates	E_0	Q	C	B_0
12	0	0.000801	0.08099	−0.001308	−0.00000351
12	15,000	0.001385	7.84837	1.85463	0.006261
24	0	0.000801	0.08406	0.000121	0.00000726
24	15,000	0.001394	6.40892	3.91831	0.006519
24	19,072	0.001338	5.63515	2.70808	0.005173

Table 12: Mean energy components in the same residual coordinates. The full-precision components satisfy the accounting identity to less than 10^{-13} in the saved aggregates.

Positive C records aggregate alignment across depth; Q includes both directed and random increments. We do not identify either with an independently estimated SDE coefficient. Additional saved profiles at updates 1000, 2000, and 3000 describe the development of early and late activity. Local-window comparisons use equal normalized depth widths, for example two of twelve versus four of twenty-four stages. Earlier drift/diffusion labels refer to finite-window proxies requiring extra local assumptions, not direct drift/diffusion identification.

Layer pairs and observation granularity. Motivated by the layer-increment angle measurements of Liu et al. (2026), we retain both increment magnitudes and all layer-pair inner products. On the same saved trajectories, form the uncentered increment Gram $G_{\ell m} = \langle \Delta \boldsymbol{h}_\ell, \Delta \boldsymbol{h}_m \rangle / N_\flat$ for each token. Then $Q = \operatorname{tr} G$ and $C = \sum_{\ell \neq m} G_{\ell m}$. At update 15,000, the mean adjacent-increment cosine is similar for twelve and twenty-four stages, 0.1764 and 0.1838, whereas C/Q is 0.2363 and 0.6114. Adjacent pairs contribute 2.63068 and 2.84889 to C; more distant pairs contribute -0.77605 and 1.06942. Local alignment alone therefore does not determine the accumulated cross-depth contribution.

Combining consecutive observed increments into six equal depth windows gives the following accounting at the same checkpoints:

Stages	Stages per window	Q_{window}	C_{window}	D
12	2	9.76888	−0.06589	9.70300
24	4	10.51574	−0.18851	10.32723

Here $D = \|\boldsymbol{h}_L - \boldsymbol{h}_0\|^2 / N_\flat = Q + C$ is unchanged. Within-window cross terms move into Q_{window}, leaving C_{window} negative in both cases. This is a rebinning of fixed trajectories, not a sliding intervention on the network. It illustrates why the observation granularity must accompany Q/C comparisons. Figure 20 shows the full lag contributions and their cumulative sums, alongside the change in C/D under these observation windows.

I.3 A COMPLETE RUN WITH NS5 MUON ON EVERY MATRIX

This experiment uses a different GPT-2-style architecture, adapted from the `modded-nanogpt` record of 2024-10-10 (Jordan et al., 2024b) (commit `bc3a0c2d640d0d73dedaef87eae26148d2e32afb`). The model has 123,532,032 parameters: twelve blocks, width 768, twelve 64-dimensional heads, and MLP width 3072. It uses GELU, rotary positional embeddings, non-affine RMSNorm with stabilizer 10^{-6}, no biases or dropout, and a tied 50257×768 embedding/readout. Each attention residual branch is multiplied by $1/\sqrt{24}$; the MLP residual branch has multiplier one. Every matrix with fan-in d is independently Gaussian-initialized with standard deviation $1/\sqrt{3d}$, except that the tied endpoint matrix is initialized only once.

All 49 parameter matrices use Muon (Jordan et al., 2024a), with the five-step Newton–Schulz polynomial implementation from that record. If D_t is the current gradient, the momentum and Nesterov

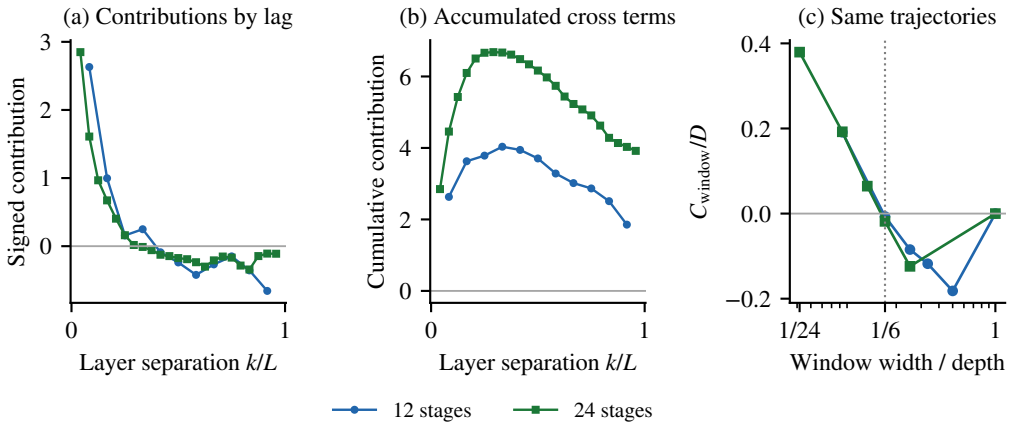


Figure 20: Cross-depth contributions at update 15,000 in the AdamW models. Left: the signed lag contribution $2\sum_{\ell} G_{\ell,\ell+k}$, averaged over the four probe texts and their tokens; these are magnitude-weighted inner products, not average cosines. Middle: its cumulative sum, ending at C. Right: regrouping the same increments into equal observation windows changes C_{window}/D while keeping squared residual displacement D fixed. The dotted line marks six windows, each spanning $1/6$ of the depth.

direction are $M_t = 0.95M_{t-1} + D_t$ and $G_t = D_t + 0.95M_t$, with zero initial momentum. Normalize $X_0 = G_t/(\|G_t\|_{\mathrm{F}} + 10^{-7})$ and apply five iterations

$$X_{j+1} = 3.4445X_j - 4.7750X_jX_j^{\mathsf{T}}X_j + 2.0315(X_jX_j^{\mathsf{T}})^2X_j. \tag{193}$$

The Frobenius norm is computed before conversion to bfloat16; the iteration uses bfloat16. Tall matrices are transposed during the iteration and restored afterward. For an $a \times b$ matrix, the parameter increment is $-\eta_t\sqrt{\max(a,b)}X_5$. The combined Q/K/V matrix is split into its three 768×768 parts before normalization and iteration; each uses multiplier $\sqrt{768}$. The tied endpoint receives its combined input/readout gradient once, with multiplier $\sqrt{50257}$. This NS5 map is distinct from both exact spectral clipping and the Taylor polynomial used in the logic experiment.

Base rates are 0.00036 for internal matrices and 0.00018 for the tied endpoint. For t completed updates, the next update multiplies both rates by $\min\{1, (6200 - t)/1800\}$, for $t = 0, \ldots, 6199$. There is no warmup, weight decay, gradient clipping, or Adam group. The endpoint rate was selected by validation loss in a prior 713-update, context-512 pilot and then fixed; Gaussianity plots did not enter selection. An additional seed was checked in that pilot, but the full run has only seed 1337.

Training uses 6,200 updates at 524,288 tokens per update, context 1024, and sixteen microbatches of 32 sequences. Parameters are float32; training autocasting is bfloat16. The run uses PyTorch 2.8.0/CUDA 12.8 on one H100 80GB, with `torch.compile`. Data use the GPT-2-tokenized FineWeb-Edu release (Lozhkov et al., 2024) `kjj0/finewebedu10B-gpt2`, pinned to revision `440d9d739f970c5d80492e12e81d63d60622ca30`. Thirty-three ordered training shards each contain 100 million tokens; a separate shard supplies validation. The cursor crosses shard boundaries without cycling. Validation uses a fixed prefix of 262,144 tokens, excluded from training. The final loss 3.083088636 is measured after all 6,200 updates. This corpus preparation and evaluation set are not assumed identical to those of the historical AdamW runs.

Distribution measurements. Before training, the measurement updates are fixed as

$$0, 1, 5, 10, 25, 50, 100, 250, 500, 750, 1000, 1500, 2000, 3000, 4000, 5000, 6200.$$

At each time, eight validation sequences start at offsets $0, 8192, \ldots, 57344$. We record positions 255 and 1023 at all 13 residual boundaries. Backward variables are the actual derivatives of token-1023 pre-softmax logits 13 and 262 with respect to each recorded residual, without a loss derivative or random projection. They are saved in raw Jacobian units; within-vector standardization removes any

scalar gradient convention from the shape diagnostic. Measurement uses float32 without autocasting. The last-boundary Jacobian is checked against the analytic derivative of RMSNorm followed by the tied linear readout; its maximum error at the final time is 3.73×10^{-9}.

Each time has 208 nondegenerate forward slices and 200 backward slices per logit; the final-boundary derivative at the earlier token is structurally zero. For each vector separately, we report its uncentered RMS, centered standardized third moment, and fourth moment minus three. Fixed token-1023 pairs are (h^1, h^6), (h^6, h^{12}), and $(h^6, \partial f_{13}/\partial h^6)$, using the same residual coordinate in both components for all eight texts. Gaussian 95% ellipses use the measured covariance and are descriptive references, not confidence regions. At the terminal time, the fractions outside these ellipses span 3.906–6.641%; the raw final-residual mean RMS is 22.152, compared with 0.724 at initialization. The maximum backward excess kurtosis of 0.9709 occurs at update 5000, sample 2, boundary 6, token 1023, logit 13, and is retained. Figure 11 in Appendix B.8.4 uses the first predefined text, without selection by Gaussian appearance. All predefined times and slices are retained in the measurement archive. The numerical supplement provides the complete time-course summaries and the illustrative raw vectors.

Internal MLP distributions. Additional measurements in the same seed-1337 model use the initial and update-4500 checkpoints, with the same eight texts and two token positions. Forward hooks record the pre-GELU up-projection, GELU output, MLP down-projection increment, and residual after each complete block, giving 192 separate slices per variable and time. Hidden vectors have 3072 coordinates; increments and residuals have 768. Measurements use CPU float32 without autocasting. Each vector is centered and standardized separately, without pooling coordinates across texts, positions, or blocks.

The first block develops strongly non-Gaussian preactivations: all sixteen slices at update 4500 have skewness 3.38–7.54 and excess kurtosis 29.16–78.75 (Table 13). This departure therefore precedes GELU. It varies substantially across depth: the median pre-GELU excess kurtosis over all 192 slices is 0.5121, with smaller deviations in middle blocks (Fig. 21). Approximately Gaussian residual shapes thus coexist with strongly non-Gaussian hidden variables in the same model. Internal measurements stop at update 4500, whereas the residual and Jacobian measurements above continue through update 6200.

	Initialization		Update 4500	
First-block variable	Skewness	Excess kurt.	Skewness	Excess kurt.
Pre-GELU	-0.0214	0.0256	5.5374	54.8062
Post-GELU	1.6722	3.0864	10.9190	138.0886
MLP increment	0.0006	0.0035	0.0572	-0.0393
Post-block residual	0.0081	0.0132	0.0374	-0.1160

Table 13: Internal-variable contrast in the all-NS5 model. Entries are medians over sixteen separately measured first-block slices, not uncertainty estimates; coordinates and slices are not assumed independent.

I.4 Exploratory nonuniform depth interventions

These historical AdamW runs keep the twelve- or twenty-four-stage parameter budget and multiply both residual branches, including their down-projection biases, by $c_\ell = L\Delta\tau_\ell$. The original learning-rate and decay schedule is retained; the corresponding block Adam stabilizer is $c_\ell 10^{-8}$, while endpoint stabilizers remain 10^{-8}. Half Copy (HC) changes initialization only: up-projections are copied within a group and down-projections combine a shared and independent Gaussian with correlation ρ, preserving marginal variances. Attention groups consist of complete heads, with matching Q/K/V rows and output columns. The parameters remain untied during training.

For the twenty-four-stage two-end grid, groups of 8, 12, and 4 stages use $c = (0.75, 1.25, 0.75)$ and pair HC at both ends with $\rho = 1/3$. Soft-head and strong-head variants use $c = (0.825, 1.2, 0.75)$ and $(0.675, 1.30, 0.75)$, respectively; their head correlations are $7/33$ and $13/27$, with tail correlation $1/3$. The twelve-stage mild-tail grid has groups of 3, 7, and 2 stages, $c = (1, 15/14, 0.75)$, and pair HC only in its last two stages. The fitted twelve-stage grid ranges from $c_1 = 1.4568$ to

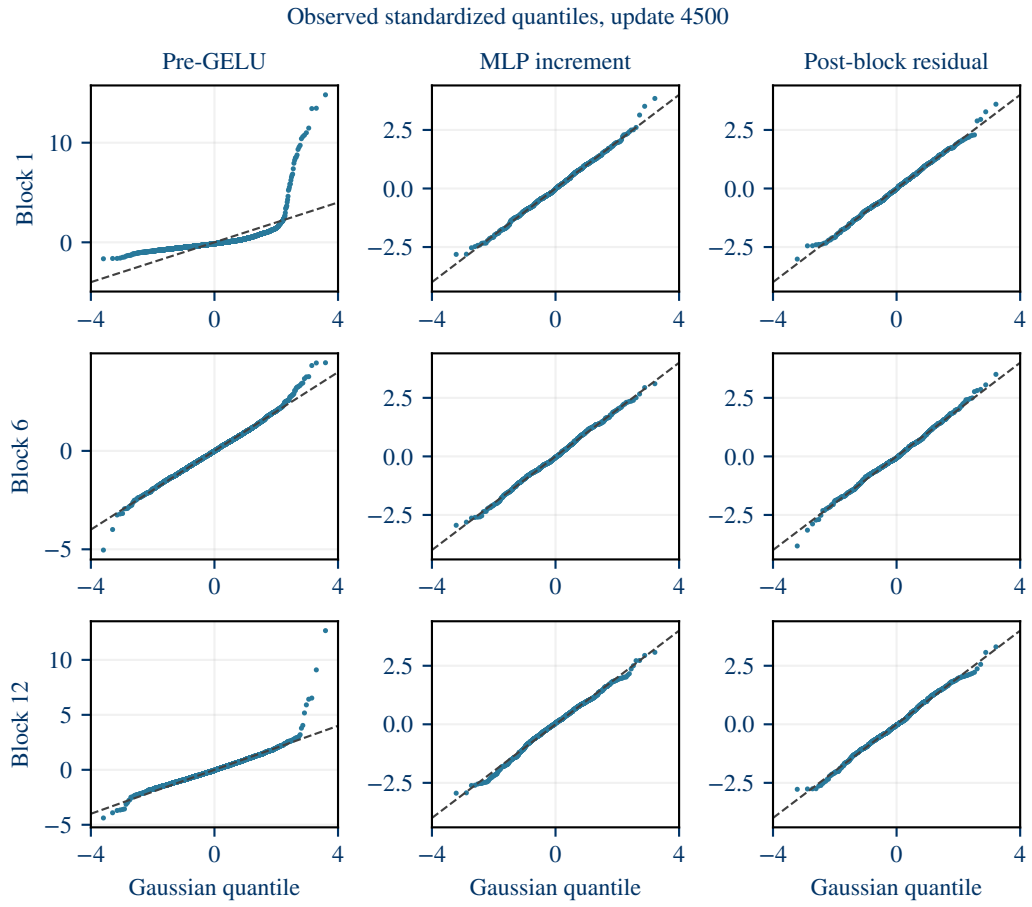


Figure 21: Internal distributions at update 4500 for the first fixed text, token 1023, and blocks 1, 6, and 12. Each panel standardizes one vector separately and retains its full tails; vertical ranges differ across panels. Preactivations have 3072 coordinates, increments and residuals 768. Dashed lines give the standard Gaussian reference.

Stages	Intervention	Validation CE	HellaSwag
12	Fitted tail, HC	3.0878	0.2996
24	Two ends, HC	3.0611	0.3089
12	Mild tail, HC	3.0741	0.3075
24	Soft head, HC	3.1034	0.3037
24	Strong head, HC	3.0981	0.3011

Table 14: All five completed adaptive candidates, seed 1337, evaluated at the same endpoint as the uniform baselines in Table 11. Favorable and unfavorable outcomes are shown.

$c_{12} = 0.3505$, with $\rho = 1/3$ and integer HC groups in the last five stages. Full-precision mesh and group-count data accompany the supplementary protocol. The no-HC twenty-four-stage control stopped at 1500 updates under its preset early-loss rule; the twelve-stage two-end run was paused at 5466. These incomplete runs supply no endpoint. The completed single-seed comparisons change full recipes and show no consistent adaptive-depth advantage.