TRANSFORMING RANK: HOW ARCHITECTURE NAVIGATES THE SPECTRAL PATHOLOGIES OF DEPTH

Anonymous authors
Paper under double-blind review

ABSTRACT

We study how each component of the Transformer feedforward block preserves rank across depth at initialization. We reinterpret skip connections and normalization as mechanisms for preserving gradient rank as well as magnitude across depth, since the operations that make a network expressive also reduce its rank. Skip connections route the gradient *around* the residual branch, where rank is lost, rather than along the long gradient paths that encourage the layers to compose. Across a combinatorial sweep of residual architectures that vary the normalization placement, initialization and branch scales, depth, branch shape and activation, we show that the effective rank of the input–output Jacobian is one function of the mean number of branches a gradient path passes through, which is determined by the branch-to-skip ratio. Deep residual networks therefore trade off rank collapse against ensemble-like behavior; preserving rank makes them ensemble-like by necessity. Normalization placement controls the branch-to-skip ratio across depth, which explains why rank collapses for Post-Norm but decays only slowly for Pre-Norm and unifies much of the normalization placement and depth scaling literature. The two-matrix structure uses additional parameters to preserve rank: the second matrix decorrelates a coherent mean spike that would grow across blocks with a single matrix and an uncentered activation, and expanding the width between the two matrices lets enough directions survive the activation to span the original space. In transformer language models, the rank at initialization follows the same function of path length, and most training failures coincide with rank collapse. Taken together, our results recast architecture design for deep networks as navigating an intrinsic tradeoff among rank collapse, ensemble-like behavior, and parameter count.

1 INTRODUCTION

The Transformer feedforward block has kept almost the same structure since the original Transformer paper (Vaswani et al., 2017): it normalizes the residual stream, expands the width with one matrix, applies an elementwise activation, and contracts back with a second matrix before adding the result to the stream. Only two parts of the block have changed in practice: the normalization has moved from after the residual addition (Post-Norm) to the branch input (Pre-Norm) (Nguyen & Salazar, 2019; Baevski & Auli, 2019), and the activation function has changed from ReLU to GELU (Hendrycks & Gimpel, 2016) to gated units (Shazeer, 2020). However, recent work shows that additional layers bring diminishing returns (Fahim & Karim, 2026; Liu et al., 2026; Sun et al., 2025), and current frontier models remain on the order of a hundred layers deep (Llama Team, 2024; DeepSeek-AI, 2024). In this work, we show that the architecture design of the feedforward block navigates an intrinsic tradeoff across depth between rank collapse and ensemble-like behavior.

In a plain MLP, the operations that make the network expressive also reduce its rank. The input–output Jacobian is a product of per-layer factors, and for random networks it keeps its rank at large depth only if the singular values of every factor are nearly equal (Saxe et al., 2014; Pennington et al., 2017). A nonlinearity spreads the singular values of each factor (Murray et al., 2022), so outside of dynamical isometry every layer loses effective rank. Across many layers this compounds into rank collapse, where the gradients and representations span only a few directions (Feng et al., 2022).

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.

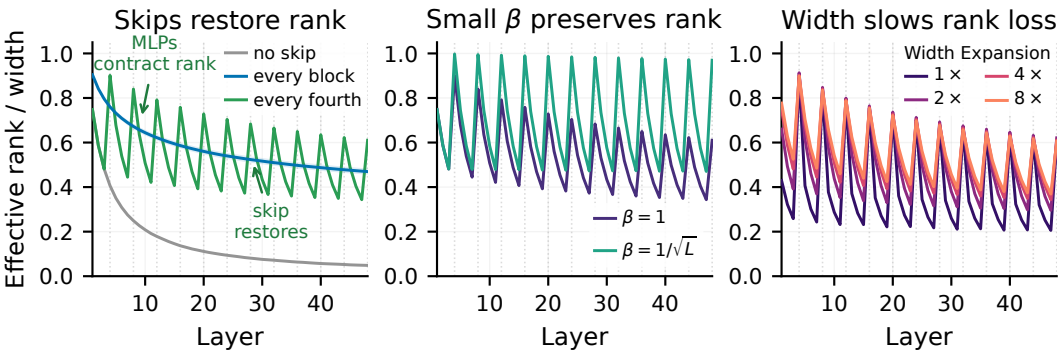

Figure 1: **Skip connections restore the effective rank that each feedforward block contracts; the branch scale β and the width expansion control how much survives with depth.** Effective rank of the Jacobian $\partial h_\ell / \partial h_0$ up to each layer ℓ, normalized by the width, for a Pre-Norm ReLU residual MLP ($L = 48$, $d = 256$, $\alpha = \beta = 1$, $4\times$ expansion unless stated; mean over 12 seeds). *Left:* a skip on every fourth block (green) shows rank contracting through the blocks and restored at each skip. *Center:* at $\beta = 1/\sqrt{L}$ the rank no longer decays. *Right:* wider branches raise the troughs (Section 5).

Skip connections and normalization layers are commonly understood as ways to keep magnitudes from vanishing or exploding with depth (Veit et al., 2016; Zhang & Sennrich, 2019); we reinterpret them as mechanisms for bypassing rank loss. Skip connections route the gradient around the residual branch, where rank is lost, through an identity path that is trivially full rank (Figure 1), and the *mean path length* $\langle k \rangle$, the average number of branches a gradient path passes through, determines how much rank survives. The placement of the normalization determines how $\langle k \rangle$ grows with depth. Every branch loses rank, so at initialization a deep residual network preserves rank only by sending most of its gradient along short paths through few blocks, as in an ensemble of shallow networks (Veit et al., 2016). Deep residual networks are therefore ensemble-like *by necessity*, which may limit how much their layers learn to compose. The other components of the block relax this tradeoff by losing less rank across and within the branches, at a cost in parameters.

We build from MLPs to the Transformer feedforward block one architecture component at a time and show that, at initialization, each component plays a distinct role in this same tradeoff:

- Skip connections route the gradient around the rank-losing branch, and the branch-to-skip ratio determines the mean path length $\langle k \rangle$ of the gradient: across 132 residual-network configurations that vary the normalization placement, initialization and branch scales and depth factorially, plus the branch shape and activation, the effective rank is one function of $\langle k \rangle$, and the rank each block loses depends on its branch-to-skip ratio even when the branch loses no rank on its own (Section 3).
- The normalization placement controls the branch-to-skip ratio across depth: for Post-Norm this ratio is constant and induces rank collapse, whereas for Pre-Norm this ratio decays, which preserves more rank but makes deeper layers contribute like an ensemble. This perspective unifies a wide range of results in the normalization-placement literature (Section 4).
- The two-matrix structure slows the collapse of the representation rank by decorrelating the branch means across layers; with a single matrix per branch, the branch means of an MLP align at $c = \mathbb{E}[\sigma]^2/\mathbb{E}[\sigma^2]$, and the representation collapses, in MLPs and in Pre-LN transformers (Section 5).
- The width expansion preserves the rank of the branch Jacobian by applying the activation in a wider space, which raises its effective rank for every activation we test (Section 5).
- In transformer language models (Section 6), the effective rank at initialization follows the same function of the path length as in residual MLPs. Within Post-LN, the path length predicts which networks train, whether it grows with depth or with the branch scale. Most training failures coincide with rank collapse.

2 Setup

We consider a residual network (He et al., 2016a) in which an input $h_0 \in \mathbb{R}^d$ is transformed by L residual blocks into h_L. Each residual block has the form

$$h_\ell = h_{\ell-1} + \beta\, f(h_{\ell-1}), \qquad \ell = 1, \ldots, L, \tag{1}$$

with branch scale β. The branch f may itself include a normalization, an elementwise activation σ, and one or two weight matrices; we write M for the number of branch matrices. By default the branch is the two-matrix feedforward block, $f(h) = W_{\text{down}}\,\sigma(W_{\text{up}}\,\text{Norm}(h))$, with $W_{\text{up}} \in \mathbb{R}^{m\times d}$ and $W_{\text{down}} \in \mathbb{R}^{d\times m}$, where m/d is the expansion ratio. Unless stated otherwise, the activation is ReLU, the normalization is RMSNorm in the Pre-Norm placement, β=1, and the weights are initialized as $\mathcal{N}(0, \alpha^2/\text{fan-in})$ with initialization scale $\alpha = 1$.

We track the rank, across depth, of three objects. The first is the input–output Jacobian $\partial h_L/\partial h_0$, an ordered product over the L blocks,

$$\frac{\partial h_L}{\partial h_0} = (I + \beta J_L)(I + \beta J_{L-1}) \cdots (I + \beta J_1), \tag{2}$$

where $J_\ell := \partial f(h_{\ell-1})/\partial h_{\ell-1}$ is the branch Jacobian of block ℓ. Since $\nabla_{h_0}\mathcal{L} = (\partial h_L/\partial h_0)^\top \nabla_{h_L}\mathcal{L}$ for a loss $\mathcal{L}$, we also refer to its rank as the gradient rank. The second is the forward residual representation $Z_\ell \in \mathbb{R}^{n\times d}$, whose rows are the stream h_ℓ for n probe inputs (Section 5). The third is the Jacobian of the branch without its normalization, written J_f (Section 5).

We measure the effective rank (Roy & Vetterli, 2007) rather than the exact rank, since a Jacobian can stay full rank while its singular values spread apart across depth. For a matrix A with singular values $\sigma_1 \geq \cdots \geq \sigma_N$, the effective rank is the exponential of the spectral entropy, $\text{erank}(A) = \exp(-\sum_k p_k \ln p_k)$ with $p_k = \sigma_k/\sum_j \sigma_j$, which we report normalized by the width d; it equals 1 when all singular values are equal and drops toward 0 as the spectrum collapses in a few directions. We evaluate the Jacobians at initialization at a random Gaussian input and report means over samples of the weights and the input unless stated otherwise.

3 Skip Connections and Rank Restoration

Skip connections route the gradient around the residual branch, where rank is lost, through an identity path that is trivially full rank. When the skips are spaced across several blocks, the rank traces a *sawtooth*, decreasing through the MLP blocks and recovering at each skip connection (Figure 1, left). This creates a tradeoff: if all of the gradient passed through the branches, the rank would collapse, and if it all passed through the skips, the network would act like an ensemble of individual layers rather than a deep network whose layers compose.

The branch has two parameters that control its scale, the branch scale β and the initialization scale α of its weight matrices. We define the branch-to-skip ratio r_ℓ of each block as the ratio of the Frobenius norms of the two terms in its Jacobian,

$$\frac{\partial h_\ell}{\partial h_{\ell-1}} = \underbrace{I}_{\text{skip}} + \underbrace{\beta J_\ell}_{\text{branch}}, \qquad r_\ell \equiv \frac{\|\beta J_\ell\|_F}{\|I\|_F}. \tag{3}$$

At initialization and unit stream scale, $r_\ell^2 = p\,\beta^2\alpha^{2M}$ in expectation, where $p = \mathbb{E}[\sigma'(z)^2]$ ($p = 1/2$ for ReLU). For positively homogeneous activations and a branch without output normalization, networks with the same $\beta\alpha^M$ compute the same function at initialization (Appendix A.4).

At a matched branch-to-skip ratio, the rank of a residual network barely depends on its branch (Appendix A, Table 1). Without skip connections, orthogonal linear layers keep full rank at any depth, whereas standard tanh and ReLU layers keep only 0.05–0.11 of the width after 48 layers. With skip connections at $r = 0.3$, the same five branches all keep 0.39–0.40 of the width after 48 blocks, including the orthogonal branch. The block $I + rB$ loses rank even when the branch does not: for an orthogonal B, the singular values of $I + rB$ spread over $[1 - r, 1 + r]$, and this spread compounds across blocks. Since the skip routes the gradient around each branch, it is the block as a whole that must preserve rank. A linear network with skew-symmetric branches is the one exception, since each block is then orthogonal to first order in r (Appendix A).

Veit et al. (2016) decompose a residual network forward pass into 2^L paths that take either the skip or the branch at each block. We apply a similar expansion to the input–output Jacobian. Grouping the paths by their order k, the number of blocks at which a path takes the branch rather than the skip, gives

$$\frac{\partial h_L}{\partial h_0} = \sum_{k=0}^{L} \beta^k S_k, \qquad S_k = \sum_{1\leq \ell_1 < \cdots < \ell_k \leq L} J_{\ell_k} \cdots J_{\ell_1}, \tag{4}$$

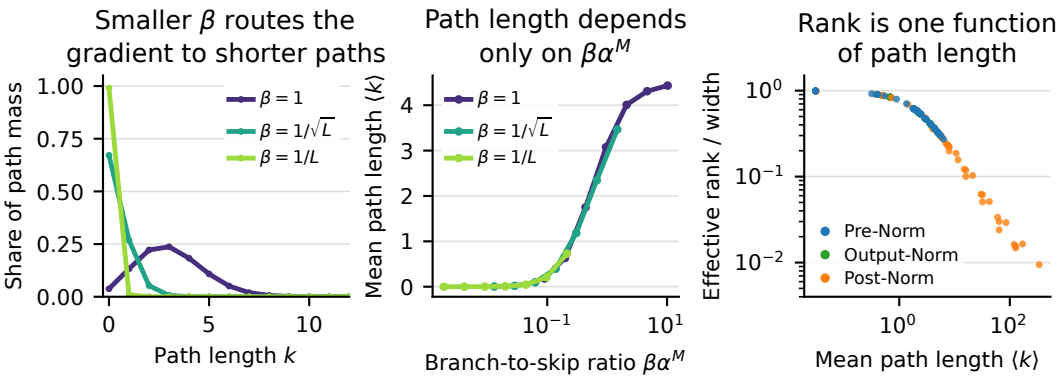

Figure 2: **Skip connections carry most of the gradient along short paths. Their mean length $\langle k \rangle$ depends only on $\beta\alpha^M$, and the effective rank is one function of $\langle k \rangle$.** Residual MLPs at initialization, $d = 256$. *Left:* share of the squared Jacobian magnitude m_k^2 in the order-k paths (Eq. 4) for a Pre-Norm ReLU MLP ($L = 48$, $4\times$ expansion, $\alpha \approx 1$; 8 seeds). *Center:* $\langle k \rangle$ (Eq. 5) against $\beta\alpha^M$ ($M = 2$), sweeping α at three values of β. *Right:* effective rank of the input–output Jacobian, normalized by width, for 132 configurations that vary the placement, α, β, the depth (12–384), the branch shape and the activation (3 seeds each).

where each order-k path carries weight $(\beta\alpha^M)^k$. To summarize where the magnitude of the Jacobian sits, we define the *mean path length*

$$\langle k \rangle = \frac{\sum_k k\, m_k^2}{\sum_k m_k^2}, \qquad m_k = \|\beta^k S_k\|_F, \tag{5}$$

which we compute exactly by dynamic programming over the blocks (Appendix E). Treating the blocks as independent gives $\langle k \rangle \approx \sum_\ell r_\ell^2/(1+r_\ell^2)$, the sum that Liu et al. (2020) use to bound output sensitivity to parameters. This approximation is within 0.3% of the exact value at the median over the runs of Figure 2, and within 1.5% in the worst run. The mean path length is the Jacobian counterpart of the path-order polynomial in the kernel analyses of Martens et al. (2021) and Zhang et al. (2022). A smaller branch scale shifts the magnitude onto shorter paths, with $\langle k \rangle = 3.1$, 0.40 and 0.01 for $\beta = 1$, $1/\sqrt{L}$ and $1/L$ at $L = 48$ (Figure 2, left), and $\langle k \rangle$ depends on β and α only through $\beta\alpha^M$ (Figure 2, center).

Across normalization placements, initialization and branch scales, depths, branch shapes and activations, the effective rank of the input–output Jacobian is one function of the mean path length $\langle k \rangle$ (Figure 2, right). The 132 configurations fall on one curve, from 0.99 of the width at $\langle k \rangle \approx 0.03$ to 0.01 at $\langle k \rangle \approx 340$. Note that this curve is an empirical finding rather than a consequence of Eq. 4, since a sum of paths need not inherit the rank loss of its individual paths.

From this perspective, downscaling the residual branch, as the depth-scaling literature often recommends, or initializing it at zero (Bachlechner et al., 2021; De & Smith, 2020; Zhang et al., 2019) avoids rank collapse by shortening the paths, possibly at the expense of composition. At $\beta = 1/\sqrt{L}$, the scaling proposed for stable gradients, feature learning and signal propagation (Hayou et al., 2021; Yang et al., 2024; Bordelon et al., 2024; Noci et al., 2022), $\langle k \rangle$ stays fixed as the depth grows (Figure 3, center) and the sawtooth no longer decays (Figure 1, center). Without downscaling, the growing residual stream needs normalization; the normalization placement determines how the branch-to-skip ratio and therefore the mean path length $\langle k \rangle$ change across depth (Section 4).

4 Normalization Placement and the Branch-to-Skip Ratio

A normalization layer can be placed at three locations in a residual block (Kim et al., 2025). Writing f now for the branch without its normalization, the branch input placement (Pre-Norm) gives $h_\ell = h_{\ell-1} + \beta f(\mathrm{Norm}(h_{\ell-1}))$, the branch output placement (which we call Output-Norm) gives $h_\ell = h_{\ell-1} + \beta\,\mathrm{Norm}(f(h_{\ell-1}))$, and the post-residual placement (Post-Norm) gives $h_\ell = \mathrm{Norm}(h_{\ell-1} + \beta f(h_{\ell-1}))$. The original Transformer used Post-Norm (Vaswani et al., 2017), but most current language models use Pre-Norm (Qwen Team, 2025; OpenAI, 2025; DeepSeek-AI, 2024). We

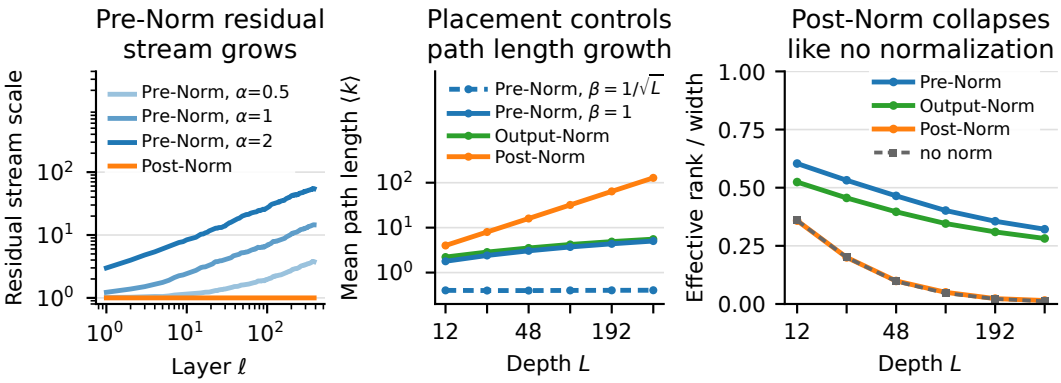

Figure 3: **The normalization placement determines how the mean path length grows with depth, and the rank follows: Post-Norm collapses like a network with no normalization, while Pre-Norm and Output-Norm decay slowly.** ReLU residual MLPs at initialization, $d = 256$, $\alpha = \beta = 1$ unless stated. *Left:* residual stream scale by layer ($L = 384$). *Center:* $\langle k \rangle$ by depth (3 seeds). *Right:* effective rank of the input–output Jacobian, normalized by width. Left and right use a branch with two $d \times d$ matrices and one sample of weights and input; center uses $4\times$ expansion.

consider normalization operations that subtract the mean, rescale the root-mean-square of the input, or do both, which includes RMSNorm (Zhang & Sennrich, 2019) and LayerNorm (Ba et al., 2016).

The Jacobian of the normalization is a natural suspect for the rank collapse: mean subtraction removes the all-ones direction from the gradient and rescaling removes the current stream direction (Eq. 17; Xu et al., 2019), and deep Post-Norm networks show collapsed Jacobian spectra at initialization (Bachlechner et al., 2021). In the Post-Norm block Jacobian $J_{\mathrm{Norm}}(I + \beta J_\ell)$, every gradient path passes through this projection, whereas in the Pre-Norm block Jacobian $I + \beta J_\ell J_{\mathrm{Norm}}$ the identity carries the gradient around it (Wang et al., 2019; Emadi, 2026). Two measurements show that the projection is not what collapses the rank in the networks we study. First, Post-Norm loses rank at the same rate as a network with no normalization at all, which removes no directions (Figure 3, right). Second, a detached rescaling (Xu et al., 2019), which keeps the forward pass of RMSNorm but stops the gradient through the stream scale, removes no direction from the gradient and still collapses the rank like ordinary RMSNorm, with the projection changing the effective rank / width by at most 0.004 (Appendix B.2).

Instead, the normalization placement determines how the branch-to-skip ratio changes across depth. In Pre-Norm, the branch acts on a normalized input, so the ratio is its value at unit stream scale divided by the stream scale $\|h_{\ell-1}\|/\sqrt{d}$. The Pre-Norm stream grows with depth, since each block adds a branch output of roughly fixed magnitude that is nearly orthogonal to the stream, so $\|h_\ell\| \propto \sqrt{\ell}$ and the ratio decays as $1/\sqrt{\ell}$ (Figure 3, left). Output-Norm fixes the scale of each branch output, so α cancels and the ratio decays with the growing stream in the same way. In Post-Norm, the rescaling resets the stream to unit scale at every block, and the ratio stays at $\sqrt{p}\,\beta\alpha^M$. With no normalization, the ratio is also constant, since a ReLU branch Jacobian does not depend on the scale of its input.

The placement therefore determines how the mean path length grows with depth: linearly for Post-Norm and logarithmically for Pre-Norm and Output-Norm (Figure 3, center). With $\langle k \rangle \approx \sum_\ell r_\ell^2/(1 + r_\ell^2)$ from Section 3, a constant ratio gives $\langle k \rangle \approx L\, r^2/(1 + r^2)$, which is $0.33L$ at $\alpha = \beta = 1$, while $r_\ell^2 \propto 1/\ell$ gives a harmonic sum that grows as $\ln L$; we measure $\langle k \rangle = 0.94 \ln L - 0.6$ for Pre-Norm and $0.97 \ln L - 0.2$ for Output-Norm. Liu et al. (2020) derive the same linear and logarithmic laws for this sum as bounds on output sensitivity to parameters. Here the sum is the mean path length of the Jacobian, and through Figure 2 it determines the rank. Post-Norm collapses from 0.36 of the width at $L = 12$ to 0.015 at $L = 384$, at the same rate as a network with no normalization, while Pre-Norm and Output-Norm decay slowly, from 0.60 to 0.32 and from 0.52 to 0.28 (Figure 3, right). Because $\langle k \rangle$ keeps growing, the Pre-Norm rank keeps falling with depth, only more slowly. Scaling the branch by $\beta = 1/\sqrt{L}$ holds the Pre-Norm $\langle k \rangle$ at 0.40 for every depth, consistent with the iso-rank contours in Appendix B.3.

Training failures on CIFAR-10 (Krizhevsky, 2009) coincide with rank collapse at initialization: the range of learning rates at which Post-Norm trains narrows with depth until, for $\alpha \geq 1$, no learning rate succeeds, whereas Pre-Norm trains at the largest learning rate we test throughout the grid (Appendix B.4). Section 6 tests the same prediction in transformer language models.

Finally, this branch-to-skip ratio perspective unifies the normalization-placement literature. Architectures that normalize only inside the branch allow this ratio to decay with depth, including Sandwich-LN, which normalizes both the input and the output of each branch (Ding et al., 2021), and related placements (Shleifer et al., 2021; Wang et al., 2023; Kim et al., 2025). Methods that make deep Post-Norm trainable reduce this ratio at every depth: DeepNorm (Wang et al., 2024; MiniMax, 2025) and Keel (Chen & Wei, 2026) up-scale the skip, and BranchNorm (Liu et al., 2024) rescales the branch during training. Other work reduces the number of post-residual normalizations the gradient passes through (Takase et al., 2023; Wang et al., 2026; Li et al., 2025), and branch scales such as $\beta = 1/\sqrt{L}$ hold $\langle k \rangle$ fixed as the depth grows (Section 3). These designs were motivated by training stability, magnitude control or hyperparameter transfer, but each one also changes how much of the gradient passes through the branches, and therefore how much rank survives with depth.

Pre-Norm implicitly downscales the branch: as the stream grows, each branch adds a fixed-size increment to a larger stream, which has a similar effect to using a small β in the deep blocks, even at $\beta = 1$. At initialization, the rank survives in Pre-Norm for the *same reason* that its deep blocks do little: each one is close to the identity. It is therefore no surprise that, in trained Pre-Norm language models, deep layers can be removed with little loss (Sun et al., 2025; Gromov et al., 2025; Men et al., 2025) and their block Jacobians approach the identity (Sun et al., 2025). We can view Pre-Norm as the short-path side of the tradeoff between rank collapse and ensemble-like behavior: it preserves more rank at the cost of deep layers that contribute little.

5 The Two-Matrix Branch and Width Expansion

Transformer feedforward blocks place two weight matrices inside every residual branch. The previous sections might suggest the opposite design: if skip connections restore the rank that branches lose (Section 3), then perhaps a skip connection around every single matrix would be preferable. In this section, we show that the second matrix protects the rank of the forward representation Z_ℓ, which collapses at initialization when each branch has a single matrix, and that expanding the width between the two matrices raises the rank of the branch Jacobian. The mean of the residual stream over inputs is a running sum, $\mu_\ell = b_1 + \cdots + b_\ell$, where b_k is the mean of the output of branch k. When this sum grows large, it creates a rank-one mean spike that dominates the spectrum of Z_ℓ, and the representations of all inputs converge to this single direction. An uncentered activation gives every branch output such a mean: Davis & Drusvyatskiy (2025) show that the post-activation matrix of an uncentered activation contains a mean spike, and Brock et al. (2021) observe the resulting mean shift in unnormalized residual networks.

With one matrix per branch, the alignment between consecutive branch means settles at the activation moment ratio $c = \mathbb{E}[\sigma]^2/\mathbb{E}[\sigma^2]$, and a second matrix removes it (Figure 4, first panel). For ReLU, the cosine between consecutive branch means falls from 0.92 in the first layer to 0.320 ± 0.027 beyond layer 100, against $1/\pi \approx 0.318$; for GELU it settles at 0.19, against $c = 0.187$. With two matrices it is 0.001 ± 0.035, and tanh branch means vanish by odd symmetry. With a single matrix, the alignment comes from a shared direction: the activation writes directly into the stream, and an uncentered activation has a positive mean in every coordinate, so every branch mean contains the same component along the all-ones direction, carrying a fraction c of its energy, plus a remainder that is independent across layers. Once the accumulated mean dominates the stream, the remainders cancel in the cosine and only the shared component survives, which gives c (Appendix C). A second matrix removes the shared component, because the W_{down} of each layer rotates the all-ones part to a different random direction.

Aligned means accumulate linearly, whereas decorrelated means accumulate like a random walk: the norm of the mean grows with fitted exponent 0.99 in depth for one matrix and 0.56 for two. As a result, one ReLU matrix per branch leaves the representation of $n = 512$ inputs with 0.009 of the width at $L = 512$, while two matrices keep 0.50 and a single tanh matrix keeps 0.58 (Figure 4, second panel); $4\times$ SwiGLU branches form no spike and keep 0.91. The second matrix slows the collapse rather than preventing it, since the two-matrix rank still falls from 0.86 to 0.50 over 512

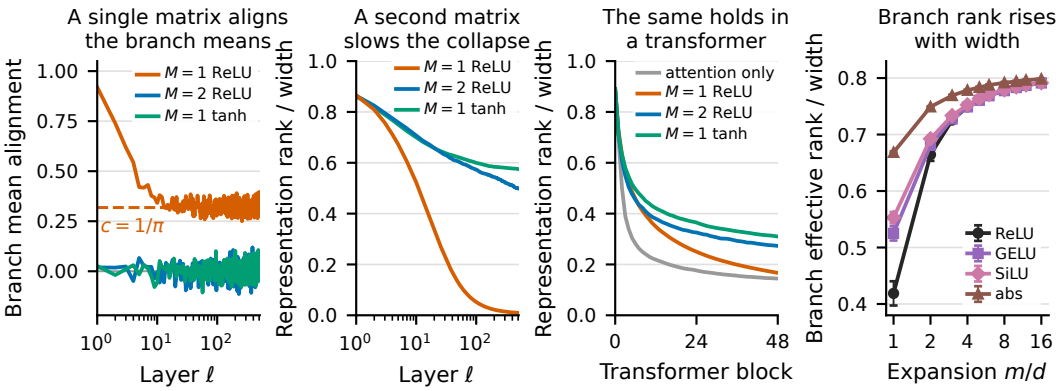

Figure 4: **A single ReLU branch matrix aligns the branch means across layers and collapses the representation rank; a second matrix or a centered activation slows the collapse, in MLPs and in a transformer, and a wider branch has a higher-rank Jacobian.** Residual networks at initialization (Pre-Norm, $\beta = 1$, $d = 256$). *First:* cosine between consecutive branch means ($n = 4096$ inputs); dashed line at $c = 1/\pi$. *Second:* effective rank of the residual representation of $n = 512$ inputs ($L = 512$, 3 seeds). *Third:* token effective rank in a Pre-LN transformer ($N = 48$, 2048 FineWeb tokens, 3 seeds) with the FFN removed (attention only) or replaced by each branch type. *Right:* effective rank of the branch Jacobian J_f against the expansion ratio (40 seeds, mean $\pm$ std). Effective ranks are normalized by width.

layers. The same ordering appears in a Pre-LN transformer at initialization, where the token rank after 48 blocks is 0.17 with one ReLU matrix, 0.27 with two and 0.31 with tanh (Figure 4, third panel). In Post-LN, every variant collapses (Appendix F), and the LayerNorm after each block removes the all-ones component that carries the one-matrix spike.

A normalization removes the spike only if it subtracts the mean after the activation, where the mean is created: mean subtraction in the Pre-Norm only slows the collapse, and RMS rescaling alone leaves it unchanged (Appendix C). Standard Transformer blocks do not need this extra normalization, because the second matrix already removes the aligned component. On CIFAR-10 with Pre-RMSNorm, the one- and two-matrix networks reach similar accuracy despite the collapsed representation, so whether this collapse matters on harder tasks remains open (Appendix C).

The two-matrix structure also allows the width to expand inside the branch: the first matrix maps the input to m hidden units before the activation is applied, and the second contracts back to the original width d. Differentiating the branch $f(x) = W_{\text{down}}\,\sigma(W_{\text{up}}x)$ gives $J_f = W_{\text{down}}\,D\,W_{\text{up}}$, where $D = \operatorname{diag}(\sigma'(W_{\text{up}}x))$ holds the activation derivatives; for ReLU it keeps only the hidden units with nonzero derivative, a fraction $p = 1/2$. Because W_{down} is full rank and independent of $D\,W_{\text{up}}$ at initialization, $\operatorname{rank} J_f = \operatorname{rank}(D\,W_{\text{up}})$, which is full only if at least d of the m hidden units survive the mask. In the large-width limit, this happens above an expansion ratio of $m/d = 2$ for ReLU, the threshold that Lei et al. (2019) and Puthawala et al. (2022) derive for the invertibility and injectivity of ReLU layers; at finite width the number of survivors is binomial, and at $m = 2d$ with $d = 512$ only about half of the draws reach d. The Gram matrix of the surviving rows follows the Marchenko–Pastur law (Marchenko & Pastur, 1967) with aspect ratio $\gamma = pm/d$, so a wider branch improves the conditioning of $D\,W_{\text{up}}$, though not of J_f as a whole (Appendices D and D.5). For GELU and SiLU (Hendrycks & Gimpel, 2016; Elfwing et al., 2018), the derivative is nonzero almost surely, so J_f is full rank already at $m = d$.

The effective rank of the branch Jacobian rises with the expansion ratio for every activation we test (Figure 4, right). From $m/d = 1$ to 16, $\operatorname{erank}(J_f)/d$ rises from 0.42 to 0.79 for ReLU, from 0.53 and 0.55 to 0.79 for GELU and SiLU, and from 0.67 to 0.80 for the absolute value. All four approach 0.805, the effective rank per width of a square Gaussian matrix, which J_f approaches as $D\,W_{\text{up}}$ becomes well conditioned. At small expansion ratios, the smooth activations keep more effective rank than ReLU, consistent with attenuating directions rather than removing them. While the skip connection preserves the rank of the propagated gradient by routing it around the branch, the width expansion preserves rank inside the branch itself, at the cost of additional parameters.

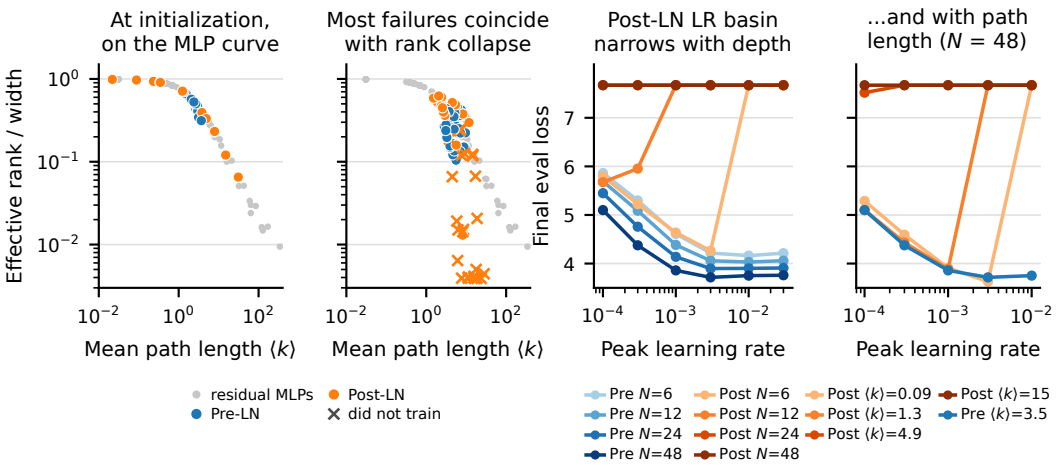

Figure 5: **Transformers (Section 6) at initialization fall along the MLP rank curve, most training failures coincide with rank collapse, and the range of trainable learning rates for Post-LN narrows as depth and the mean path length grow.** *First:* effective rank of the same-token Jacobian at initialization against $\langle k \rangle$ for all transformers, over the residual MLPs of Figure 2 (gray). *Second:* the same after training, for every run of the depth grid; $\times$ marks runs that fail. *Third:* final evaluation loss against peak learning rate with the plain initialization; darker is deeper ($N = 6$–48). *Fourth:* Post-LN at $N = 48$ with the branch scale β set to give the initial $\langle k \rangle$ in the legend; Pre-LN for reference. One seed per configuration.

6 Transformer Language Models

Transformers interleave the feedforward blocks with self-attention, which could change the rank behavior from residual MLPs in either direction. Self-attention can collapse the representations of all tokens toward a single vector (Dong et al., 2021; Noci et al., 2022), but it also mixes information across positions, which could restore rank that the feedforward blocks lose. We therefore test whether the predictions from residual MLPs hold in transformers: that the rank at initialization follows the mean path length along the same curve, and that a longer path length makes Post-LN harder to train. We train transformers of width $D = 256$ with $N \in \{6, 12, 24, 48\}$ blocks in the Pre-LN or Post-LN placement to 20 tokens per parameter (Hoffmann et al., 2022) on FineWeb (Penedo et al., 2024), at six peak learning rates on one seed (Appendix F). To compare with the MLPs, we use the same-token Jacobian $\partial h_N^t / \partial h_0^t$ of the residual stream at token t, which in a causal transformer is exactly an ordered product over the $2N$ attention and feedforward sublayers, with attention entering as one more residual branch. The path decomposition and $\langle k \rangle$ therefore carry over with $2N$ blocks.

At initialization, attention does not change the relationship between path length and rank: every transformer in the depth grid and the branch-scale sweep falls along the rank curve of the residual MLPs, with $\langle k \rangle$ from 0.02 to 31 (Figure 5, first panel).

For Post-LN, the range of trainable learning rates narrows as the mean path length at initialization grows, whether $\langle k \rangle$ grows through the depth or through the branch scale (Figure 5, third and fourth panels). With the plain initialization, Post-LN trains (final loss below 7 nats) up to a learning rate of 3×10^{-3} at $N = 6$ and 3×10^{-4} at $N = 12$, and not at all from $N = 24$, as $\langle k \rangle$ grows from 1.9 to 15. At $N = 48$, branch scales that reduce $\langle k \rangle$ to 0.09 or 1.3 restore training, as does the depth-scaled initialization of GPT-2 (Radford et al., 2019), which holds $\langle k \rangle$ at 0.23 (Appendix F), whereas with $\langle k \rangle \geq 4.9$, Post-LN does not train at any of our learning rates. Pre-LN trains at every learning rate and depth, with $\langle k \rangle$ between 2.7 and 3.5 at the plain initialization.

The mean path length at initialization does not determine trainability across placements: Pre-LN at $\langle k \rangle = 3.5$ trains at every learning rate, whereas Post-LN at $\langle k \rangle = 4.9$ trains at none. Post-LN also differs from Pre-LN in ways that $\langle k \rangle$ does not capture (Xiong et al., 2020), so our claim concerns the ordering within a placement. Training increases the mean path length, and changes in rank during training coincide with trainability (Figure 5, second panel). Runs that train stay near the initialization curve at the longer path lengths they reach, with a median of 0.88 of the rank the curve gives at the

same $\langle k \rangle$. Most runs that fail end far below the curve, 17 of 28 at less than 0.3 of its rank, and seven barely change from initialization. Runs that train still lose some absolute rank, consistent with the low-rank bias of gradient-based training (Arora et al., 2019; Huh et al., 2023; Galanti et al., 2025). Since the failed runs stall, their rank collapse may be a consequence of the failure as well as a cause.

7 Related Work

The literature on dynamical isometry shows that preserving the rank of the input–output Jacobian across depth requires narrow conditions on the initialization and the activation (Saxe et al., 2014; Pennington et al., 2017; Murray et al., 2022). Tarnowski et al. (2019) obtain dynamical isometry in residual networks for any activation function by downscaling the branch so that most of the gradient flows through the skip connections, which, under our perspective, preserves rank at the cost of shorter paths. Feng et al. (2022) prove that the exact rank of the Jacobian and of the features can only decrease with depth and measure this decay in pretrained networks, and Dong et al. (2021) and Noci et al. (2022) show that attention layers collapse the rank along the token dimension.

Deep residual networks behave like ensembles of shallow subnetworks, with most of the gradient coming from relatively short paths (Veit et al., 2016). Dherin & Munn (2025) and Liu et al. (2026) connect this ensemble behavior to the branch scale and the depth scaling of language models; we find that, at initialization, preserving rank requires it. Appendix G discusses further related work.

8 Discussion, Limitations and Future Work

We propose that the right tradeoff for deep residual networks between rank collapse and ensemble-like behavior could depend on the task. In particular, next-token prediction in language models may require mostly shallow computation and be solved well by ensemble-like networks, whereas multi-step reasoning tasks may benefit from more composition between layers. Consistent with this perspective, the depth-dependent part of language-model loss scales as expected from ensemble averaging (Liu et al., 2026), whereas deeper transformers outperform wider ones on multi-step reasoning (Ye et al., 2025).

Architectures with several residual streams replace the identity skip with learned mixing across streams (Zhu et al., 2025; Xie et al., 2025). Zhu et al. (2025) motivate this design with a seesaw between vanishing gradients in Post-Norm and overly similar deep layers in Pre-Norm, both of which follow from how $\langle k \rangle$ grows with depth. We conjecture that multiple streams partly escape the tradeoff, since the network can send long paths through the branches in some streams while other streams keep the near-identity paths that preserve rank.

Our analysis focuses on the feedforward block at initialization. The transformer experiments are small (width 256, at most 48 blocks, one seed) and measure trainability rather than whether the layers learn to compose, so testing whether the ensemble regime limits composition requires tasks where depth is necessary. Tracking how the spectra evolve during training is a natural next step.

9 Conclusion

The components of the Transformer feedforward block are often understood in other terms: skip connections as preventing vanishing gradients across depth, normalization as controlling magnitudes, and the second matrix and the width expansion as adding capacity. Seen through rank, they navigate a three-way tradeoff among rank collapse, ensemble-like behavior and parameter count: skip connections and the normalization placement determine how much of the gradient passes through the rank-losing branches, and the second matrix and the width expansion reduce the rank lost across and within branches, at the cost of additional parameters. That the feedforward block has changed so little since the original Transformer suggests that its architecture may have evolved, through practice rather than design, to navigate this spectral tradeoff across depth.

AI USE STATEMENT

In this work, we used generative AI tools to help develop the conceptual framework, propose and refine hypotheses, formulate mathematical claims and derive them (including the derivations in the appendices), design and implement the experiments and analyses, and interpret the results. Generating synthetic data sets, translation, cleaning or reformatting datasets, and qualitative or thematic data analysis are not applicable to this work. Additionally, we used generative AI tools to write and edit code, create figures, search for and summarize related literature, format references, suggest the structure of the paper, and draft and edit text. We have reviewed all AI-assisted work: we revised and approved the main text sentence by sentence and reviewed the appendices; we checked derivations against numerical experiments, reviewed the code, recomputed the numbers reported in the paper from the stored experimental data, and checked citations against the cited papers. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REFERENCES

Josh Alman and Zhao Song. Only large weights (and not skip connections) can prevent the perils of rank collapse. *arXiv preprint arXiv:2505.16284*, 2025.

Federico Arangath Joseph, Jerome Sieber, Melanie N. Zeilinger, and Carmen Amo Alonso. Lambda-skip connections: the architectural component that prevents rank collapse. In *International Conference on Learning Representations*, 2025.

Sanjeev Arora, Nadav Cohen, Wei Hu, and Yuping Luo. Implicit regularization in deep matrix factorization. *Advances in neural information processing systems*, 32, 2019.

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016.

Thomas Bachlechner, Bodhisattwa Prasad Majumder, Huanru Henry Mao, Garrison W. Cottrell, and Julian McAuley. ReZero is all you need: Fast convergence at large depth. In *Uncertainty in Artificial Intelligence*, volume 161 of *Proceedings of Machine Learning Research*, pp. 1352–1361, 2021.

Alexei Baevski and Michael Auli. Adaptive input representations for neural language modeling. In *International Conference on Learning Representations (ICLR)*, 2019.

Bradley T. Baker, Barak A. Pearlmutter, Robyn Miller, Vince D. Calhoun, and Sergey M. Plis. Low-rank learning by design: the role of network architecture and activation linearity in gradient rank collapse. *arXiv preprint arXiv:2402.06751*, 2024.

David Balduzzi, Marcus Frean, Lennox Leary, J. P. Lewis, Kurt Wan-Duo Ma, and Brian McWilliams. The shattered gradients problem: If resnets are the answer, then what is the question? In *International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 342–350, 2017.

Charlie Blake, Constantin Eichenberg, Josef Dean, Lukas Balles, Luke Y Prince, Björn Deiseroth, Andres Felipe Cruz-Salinas, Carlo Luschi, Samuel Weinbach, and Douglas Orr. u-μP: The unit-scaled maximal update parametrization. In *International Conference on Learning Representations*, 2025.

Blake Bordelon, Lorenzo Noci, Mufan Bill Li, Boris Hanin, and Cengiz Pehlevan. Depthwise hyperparameter transfer in residual networks: Dynamics and scaling limit. In *International Conference on Learning Representations*, 2024.

Andrew Brock, Soham De, and Samuel L. Smith. Characterizing signal propagation to close the performance gap in unnormalized ResNets. In *International Conference on Learning Representations*, 2021.

Bo Chang, Lili Meng, Eldad Haber, Lars Ruthotto, David Begert, and Elliot Holtham. Reversible architectures for arbitrarily deep residual neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, pp. 2811–2818, 2018.

Bo Chang, Minmin Chen, Eldad Haber, and Ed H. Chi. AntisymmetricRNN: A dynamical system view on recurrent neural networks. In *International Conference on Learning Representations*, 2019.

Chen Chen and Lai Wei. Post-LayerNorm is back: Stable, ExpressivE, and deep. *arXiv preprint arXiv:2601.19895*, 2026.

Hadi Daneshmand, Jonas Kohler, Francis Bach, Thomas Hofmann, and Aurelien Lucchi. Batch normalization provably avoids ranks collapse for randomly initialised deep networks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 18387–18398, 2020.

Damek Davis and Dmitriy Drusvyatskiy. When do spectral gradient updates help in deep learning? *arXiv preprint arXiv:2512.04299*, 2025.

Soham De and Samuel L. Smith. Batch normalization biases residual blocks towards the identity function in deep networks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 19964–19975, 2020.

DeepSeek-AI. DeepSeek-V3 technical report. *arXiv preprint arXiv:2412.19437*, 2024.

Mostafa Dehghani, Josip Djolonga, Basil Mustafa, Piotr Padlewski, Jonathan Heek, Justin Gilmer, Andreas Peter Steiner, Mathilde Caron, Robert Geirhos, Ibrahim Alabdulmohsin, Rodolphe Jenatton, Lucas Beyer, Michael Tschannen, Anurag Arnab, Xiao Wang, Carlos Riquelme Ruiz, Matthias Minderer, Joan Puigcerver, Utku Evci, Manoj Kumar, Sjoerd van Steenkiste, Gamaleldin Fathy Elsayed, Aravindh Mahendran, Fisher Yu, Avital Oliver, Fantine Huot, Jasmijn Bastings, Mark Collier, Alexey A. Gritsenko, Vighnesh Birodkar, Cristina Nader Vasconcelos, Yi Tay, Thomas Mensink, Alexander Kolesnikov, Filip Pavetić, Dustin Tran, Thomas Kipf, Mario Lučić, Xiaohua Zhai, Daniel Keysers, Jeremiah J. Harmsen, and Neil Houlsby. Scaling vision transformers to 22 billion parameters. In *International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 7480–7512, 2023.

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*, volume 38, pp. 137707–137739, 2025.

Benoit Dherin and Michael Munn. On residual network depth. *arXiv preprint arXiv:2510.03470*, 2025.

Ming Ding, Zhuoyi Yang, Wenyi Hong, Wendi Zheng, Chang Zhou, Da Yin, Junyang Lin, Xu Zou, Zhou Shao, Hongxia Yang, et al. CogView: Mastering text-to-image generation via transformers. *Advances in Neural Information Processing Systems*, 34:19822–19835, 2021.

Yihe Dong, Jean-Baptiste Cordonnier, and Andreas Loukas. Attention is not all you need: Pure attention loses rank doubly exponentially with depth. In *International conference on machine learning*, volume 139, pp. 2793–2803. PMLR, 2021.

Darshil Doshi, Tianyu He, and Andrey Gromov. Critical initialization of wide and deep neural networks using partial Jacobians: General theory and applications. In *Advances in Neural Information Processing Systems*, volume 36, pp. 40054–40095, 2023.

Gbètondji J-S Dovonon, Michael M. Bronstein, and Matt J. Kusner. Setting the record straight on transformer oversmoothing. *Transactions on Machine Learning Research*, 2025.

Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural Networks*, 107:3–11, 2018.

Seyed Morteza Emadi. Exact attention sensitivity and the geometry of transformer stability. *arXiv preprint arXiv:2602.18849*, 2026.

Md Muhtasim Munif Fahim and Md Rezaul Karim. The depth delusion: Why transformers should be wider, not deeper. *arXiv preprint arXiv:2601.20994*, 2026.

Ruili Feng, Kecheng Zheng, Yukun Huang, Deli Zhao, Michael I. Jordan, and Zheng-Jun Zha. Rank diminishing in deep neural networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

Jesseba Fernando and Grigori Guitchounts. Dynamics of the transformer residual stream: Coupling spectral geometry to network topology. *arXiv preprint arXiv:2605.14258*, 2026.

Advait Gadhikar and Rebekka Burkholz. Dynamical isometry for residual networks. *arXiv preprint arXiv:2210.02411*, 2022.

Tomer Galanti, Zachary S Siegel, Aparna Gupte, and Tomaso A Poggio. SGD with weight decay secretly minimizes the ranks of your neural networks. In *The Second Conference on Parsimony and Learning (Proceedings Track)*, volume 280 of *Proceedings of Machine Learning Research*, pp. 1388–1412, 2025.

Gemma Team. Gemma 2: Improving open language models at a practical size. *arXiv preprint arXiv:2408.00118*, 2024.

Alessio Giorlandino and Sebastian Goldt. Two failure modes of deep transformers and how to avoid them: a unified theory of signal propagation at initialisation. In *International Conference on Learning Representations*, 2026.

Andrey Gromov, Kushal Tirumala, Hassan Shapourian, Paolo Glorioso, and Daniel A. Roberts. The unreasonable ineffectiveness of the deeper layers. In *International Conference on Learning Representations*, 2025.

Soufiane Hayou, Eugenio Clerico, Bobby He, George Deligiannidis, Arnaud Doucet, and Judith Rousseau. Stable ResNet. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pp. 1324–1332, 2021.

Bobby He and Thomas Hofmann. Simplifying transformer blocks. In *International Conference on Learning Representations*, 2024.

Bobby He, James Martens, Guodong Zhang, Aleksandar Botev, Andrew Brock, Samuel L. Smith, and Yee Whye Teh. Deep transformers without shortcuts: Modifying self-attention for faithful signal propagation. In *International Conference on Learning Representations*, 2023.

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*, pp. 770–778, 2016a.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016b.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (GELUs). *arXiv preprint arXiv:1606.08415*, 2016.

Alex Henry, Prudhvi Raj Dachapally, Shubham Shantaram Pawar, and Yuxuan Chen. Query-key normalization for transformers. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pp. 4246–4253, 2020.

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Minyoung Huh, Hossein Mobahi, Richard Zhang, Brian Cheung, Pulkit Agrawal, and Phillip Isola. The low-rank simplicity bias in deep networks. *Transactions on Machine Learning Research*, 2023.

Yiping Ji, James Martens, Jianqiao Zheng, Ziqin Zhou, Peyman Moghadam, Xinyu Zhang, Hemanth Saratchandran, and Simon Lucey. Cutting the skip: Training residual-free transformers. In *International Conference on Learning Representations*, 2026.

Nikita Karagodin, Shu Ge, Yury Polyanskiy, and Philippe Rigollet. Normalization in attention dynamics. In *Advances in Neural Information Processing Systems*, 2025.

Akhil Kedia, Mohd Abbas Zaidi, Sushil Khyalia, Jungho Jung, Harshith Goka, and Haejun Lee. Transformers get stable: An end-to-end signal propagation theory for language models. In *International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 23449–23531, 2024.

Jeonghoon Kim, Byeongchan Lee, Cheonbok Park, Yeontaek Oh, Beomjun Kim, Taehwan Yoo, Seongjin Shin, Dongyoon Han, Jinwoo Shin, and Kang Min Yoo. Peri-LN: Revisiting normalization layer in the transformer architecture. In *International Conference on Machine Learning*, pp. 30400–30436. PMLR, 2025.

Kimi Team. Kimi K2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*, 2025.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Qi Lei, Ajil Jalal, Inderjit S. Dhillon, and Alexandros G. Dimakis. Inverting deep generative models, one layer at a time. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Pengxiang Li, Lu Yin, and Shiwei Liu. Mix-LN: Unleashing the power of deeper layers by combining Pre-LN and Post-LN. In *International Conference on Learning Representations*, 2025.

Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. Understanding the difficulty of training transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 5747–5763, 2020.

Yijin Liu, Xianfeng Zeng, Fandong Meng, and Jie Zhou. BranchNorm: Robustly scaling extremely deep transformers. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11675–11687, 2024.

Yizhou Liu, Sara Kangaslahti, Ziming Liu, and Jeff Gore. Inverse depth scaling from most layers being similar. In *International Conference on Machine Learning*, 2026.

Llama Team. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

V. A. Marchenko and L. A. Pastur. Distribution of eigenvalues for some sets of random matrices. *Mathematics of the USSR-Sbornik*, 1(4):457–483, 1967.

James Martens, Andy Ballard, Guillaume Desjardins, Grzegorz Swirszcz, Valentin Dalibard, Jascha Sohl-Dickstein, and Samuel S. Schoenholz. Rapid training of deep neural networks without skip connections or normalization layers using Deep Kernel Shaping. *arXiv preprint arXiv:2110.01765*, 2021.

Xin Men, Mingyu Xu, Qingyu Zhang, Qianhao Yuan, Bingning Wang, Hongyu Lin, Yaojie Lu, Xianpei Han, and Weipeng Chen. ShortGPT: Layers in large language models are more redundant than you expect. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 20192–20204, 2025.

Laurent Meunier, Blaise J. Delattre, Alexandre Araujo, and Alexandre Allauzen. A dynamical system perspective for Lipschitz neural networks. In *International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 15484–15500, 2022.

MiniMax. MiniMax-01: Scaling foundation models with lightning attention. *arXiv preprint arXiv:2501.08313*, 2025.

Michael Murray, Vinayak Abrol, and Jared Tanner. Activation function design for deep networks: linearity and effective initialisation. *Applied and Computational Harmonic Analysis*, 59:117–154, 2022.

Thiziri Nait Saada, Alireza Naderi, and Jared Tanner. Mind the gap: a spectral analysis of rank collapse and signal propagation in attention layers. In *International Conference on Machine Learning*, pp. 45561–45587, 2025.

Toan Q. Nguyen and Julian Salazar. Transformers without tears: Improving the normalization of self-attention. In *Proceedings of the 16th International Conference on Spoken Language Translation (IWSLT)*, 2019.

Lorenzo Noci, Sotiris Anagnostidis, Luca Biggio, Antonio Orvieto, Sidak Pal Singh, and Aurelien Lucchi. Signal propagation in transformers: Theoretical perspectives and the role of rank collapse. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27198–27211, 2022.

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, pp. 54250–54281, 2023.

OpenAI. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Guilherme Penedo, Hynek Kydlíček, Loubna Ben allal, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, and Thomas Wolf. The FineWeb datasets: Decanting the web for the finest text data at scale. In *Advances in Neural Information Processing Systems*, volume 37, pp. 30811–30849, 2024.

Jeffrey Pennington, Samuel Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice. *Advances in neural information processing systems*, 30, 2017.

George Philipp, Dawn Song, and Jaime G. Carbonell. The exploding gradient problem demystified – definition, prevalence, impact, origin, tradeoffs, and solutions. *arXiv preprint arXiv:1712.05577*, 2017.

Michael Puthawala, Konik Kothari, Matti Lassas, Ivan Dokmanić, and Maarten de Hoop. Globally injective ReLU networks. *Journal of Machine Learning Research*, 23(105):1–55, 2022.

Qwen Team. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. *OpenAI Technical Report*, 2019.

Olivier Roy and Martin Vetterli. The effective rank: A measure of effective dimensionality. In *15th European Signal Processing Conference (EUSIPCO)*, pp. 606–610. EURASIP, 2007.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *International Conference on Learning Representations*, 2014.

Biswa Sengupta, Jinhua Wang, and Leo Brunswic. JPmHC dynamical isometry via orthogonal hyper-connections. *arXiv preprint arXiv:2602.18308*, 2026.

Noam Shazeer. GLU variants improve transformer. *arXiv preprint arXiv:2002.05202*, 2020.

Han Shi, Jiahui Gao, Hang Xu, Xiaodan Liang, Zhenguo Li, Lingpeng Kong, Stephen M. S. Lee, and James T. Kwok. Revisiting over-smoothing in BERT from the perspective of graph. In *International Conference on Learning Representations*, 2022.

Sam Shleifer, Jason Weston, and Myle Ott. NormFormer: Improved transformer pretraining with extra normalization. *arXiv preprint arXiv:2110.09456*, 2021.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. RoFormer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Mingjie Sun, Xinlei Chen, J. Zico Kolter, and Zhuang Liu. Massive activations in large language models. In *First Conference on Language Modeling*, 2024.

Wenfang Sun, Xinyuan Song, Pengxiang Li, Lu Yin, Yefeng Zheng, and Shiwei Liu. The curse of depth in large language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Sho Takase, Shun Kiyono, Sosuke Kobayashi, and Jun Suzuki. B2T connection: Serving stability and performance in deep transformers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 3078–3095, 2023.

Wojciech Tarnowski, Piotr Warchoł, Stanisław Jastrzębski, Jacek Tabor, and Maciej A. Nowak. Dynamical isometry is achieved in residual networks in a universal way for any activation function. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, volume 89 of *PMLR*, pp. 2221–2230, 2019.

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*, 2017.

Andreas Veit, Michael J Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. *Advances in neural information processing systems*, 29, 2016.

Chao Wang, Bei Li, Jiaqi Zhang, Xinyu Liu, Yuchun Fan, Linkun Lyu, Xin Chen, Jingang Wang, Tong Xiao, Peng Pei, and Xunliang Cai. SpanNorm: Reconciling training stability and performance in deep transformers. In *International Conference on Machine Learning*, 2026.

Hongyu Wang, Shuming Ma, Shaohan Huang, Li Dong, Wenhui Wang, Zhiliang Peng, Yu Wu, Payal Bajaj, Saksham Singhal, Alon Benhaim, et al. Magneto: A foundation transformer. In *International conference on machine learning*, volume 202, pp. 36077–36092. PMLR, 2023.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. DeepNet: Scaling transformers to 1,000 layers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(10):6761–6774, 2024.

Peihao Wang, Wenqing Zheng, Tianlong Chen, and Zhangyang Wang. Anti-oversmoothing in deep vision transformers via the fourier domain analysis: From theory to practice. In *International Conference on Learning Representations*, 2022.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F. Wong, and Lidia S. Chao. Learning deep transformer models for machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 1810–1822, 2019.

Xinyi Wu, Amir Ajorlou, Yifei Wang, Stefanie Jegelka, and Ali Jadbabaie. On the role of attention masks and LayerNorm in transformers. In *Advances in Neural Information Processing Systems*, 2024.

Zhenda Xie, Yixuan Wei, Huanqi Cao, Chenggang Zhao, Chengqi Deng, Jiashi Li, Damai Dai, Huazuo Gao, Jiang Chang, Kuai Yu, Liang Zhao, Shangyan Zhou, Zhean Xu, Zhengyan Zhang, Wangding Zeng, Shengding Hu, Yuqing Wang, Jingyang Yuan, Lean Wang, and Wenfeng Liang. mHC: Manifold-constrained hyper-connections. *arXiv preprint arXiv:2512.24880*, 2025.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International conference on machine learning*, volume 119, pp. 10524–10533. PMLR, 2020.

Jingjing Xu, Xu Sun, Zhiyuan Zhang, Guangxiang Zhao, and Junyang Lin. Understanding and improving layer normalization. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Greg Yang and Edward J Hu. Tensor programs IV: Feature learning in infinite-width neural networks. In *International Conference on Machine Learning*, pp. 11727–11737, 2021.

Greg Yang and Etai Littwin. Tensor programs IVb: Adaptive optimization in the infinite-width limit. *arXiv preprint arXiv:2308.01814*, 2023.

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.

Tian Ye, Zicheng Xu, Yuanzhi Li, and Zeyuan Allen-Zhu. Physics of language models: Part 2.1, grade-school math and the hidden reasoning process. In *International Conference on Learning Representations*, 2025.

Biao Zhang and Rico Sennrich. Root mean square layer normalization. *Advances in neural information processing systems*, 32, 2019.

Guodong Zhang, Aleksandar Botev, and James Martens. Deep learning without shortcuts: Shaping the kernel with tailored rectifiers. In *International Conference on Learning Representations*, 2022.

Hongyi Zhang, Yann N Dauphin, and Tengyu Ma. Fixup initialization: Residual learning without normalization. In *International Conference on Learning Representations*, 2019.

Leo Zhang and James Martens. Orthogonal self-attention. *arXiv preprint arXiv:2602.05996*, 2026.

Zhongwang Zhang, Pengxiao Lin, Zhiwei Wang, Yaoyu Zhang, and Zhi-Qin John Xu. Initialization is critical to whether transformers fit composite functions by reasoning or memorizing. In *Advances in Neural Information Processing Systems*, volume 37, pp. 14093–14126, 2024.

Defa Zhu, Hongzhi Huang, Zihao Huang, Yutao Zeng, Yunyao Mao, Banggu Wu, Qiyang Min, and Xun Zhou. Hyper-connections. In *International Conference on Learning Representations*, 2025.

A Skip Connections and the Path Decomposition: Derivations

Table 1: **At a small branch-to-skip ratio r, a residual network with an orthogonal linear branch, which keeps full rank without skip connections, loses rank as fast as one with a tanh or ReLU branch.** Effective rank of the input–output Jacobian after L layers at initialization, normalized by width ($d = 256$; mean over 5 seeds). Plain rows compose the branches directly; residual rows compose blocks $I + rB$, with each branch Jacobian normalized so that r is the per-block ratio of branch norm to skip norm. Each layer draws fresh weights, and each branch sees its input rescaled to unit root-mean-square (with no gradient through the scale); the near-linear tanh branch uses pre-activation standard deviation 0.1. At $r = 1$ the tanh and ReLU branches lose somewhat more rank than the linear and near-linear ones. A linear residual network with skew-symmetric branch matrices is the exception: each block $I + rB$ is then orthogonal to first order in r, and at $r = 0.3$ the network keeps most of its rank.

	Branch Weights	Linear orthogonal	tanh (near-linear) orthogonal	tanh orthogonal	ReLU, $4\times$ orthogonal	ReLU, $4\times$ Gaussian	Linear skew-symmetric
Plain	$L = 12$	1.000	0.999	0.332	0.243	0.177	0.243
Plain	$L = 48$	1.000	0.995	0.114	0.071	0.051	0.071
Residual	$r = 0.3,\ L = 48$	0.396	0.395	0.394	0.392	0.392	0.961
Residual	$r = 0.3,\ L = 192$	0.121	0.121	0.119	0.117	0.117	0.852
Residual	$r = 1,\ L = 12$	0.366	0.366	0.325	0.301	0.284	0.706
Residual	$r = 1,\ L = 48$	0.132	0.131	0.104	0.092	0.084	0.301

This appendix collects the material behind Section 3: the skew-symmetric exception in Table 1, the path decomposition of the input–output Jacobian (Eq. 4) with the recurrence and numerical check behind it, the argument that the skip must be the identity, the growth of the stream magnitude, and the branch-scale symmetry (Appendix A.4).

A linear network with skew-symmetric branches is the one exception in Table 1. When $B^\top = -B$, we have $(I + rB)^\top (I + rB) = I + r^2 B^\top B$, so the singular values of the block move away from one only at second order in r, and the network keeps 0.96 of the width after 48 blocks at $r = 0.3$. In general, only the symmetric part of B spreads the spectrum at first order. Meunier et al. (2022) make the same split for continuous residual flows and show that a map whose Jacobian is skew-symmetric

everywhere is linear, so a nonlinear branch can take this route only approximately, as architectures built on skew-symmetric or orthogonal maps do (Chang et al., 2019; 2018; Zhang & Martens, 2026).

A.1 THE PATH DECOMPOSITION

The chain rule turns the residual recurrence $h_\ell = h_{\ell-1} + \beta f(h_{\ell-1})$ into an ordered product over the L blocks,

$$\frac{\partial h_L}{\partial h_0} = (I + \beta J_L)(I + \beta J_{L-1}) \cdots (I + \beta J_1), \qquad J_\ell := \frac{\partial f}{\partial h}\Big|_{h_{\ell-1}}. \tag{6}$$

Each J_ℓ is the branch Jacobian at the forward activation $h_{\ell-1}$, so the product is evaluated along one forward pass and holds for any differentiable branch.

Distributing the product gives one term per subset $A \subseteq \{1, \dots, L\}$: each factor contributes either I or βJ_ℓ, and A records which blocks contribute the branch. This expands the Jacobian into 2^L paths.

A subset of size $|A| = k$ carries a factor β^k and a product of k branch Jacobians. Grouping the 2^L paths by k defines the order-k sum

$$S_k := \sum_{1 \le \ell_1 < \cdots < \ell_k \le L} J_{\ell_k} \cdots J_{\ell_1}, \qquad \frac{\partial h_L}{\partial h_0} = \sum_{k=0}^{L} \beta^k S_k, \tag{7}$$

which is Eq. 4. The order k counts the branches a path passes through, from $S_0 = I$ (the pure skip) to $S_L = J_L \cdots J_1$ (every branch).

The order-k component $\beta^k S_k$ is the part of the Jacobian built from paths through exactly k branches. We measure its size by the Frobenius norm $m_k := \|\beta^k S_k\|_F$, as in Eq. 5. The components sum to the Jacobian, $\sum_k \beta^k S_k = \partial h_L / \partial h_0$, but their norms do not, so each m_k reports one order on its own.

To efficiently compute the order-k sum with $\binom{L}{k}$ terms, we use a dynamic programming algorithm over the blocks. Let $S_k^{(\ell)}$ be the sum of order-k paths through the first ℓ blocks, so $S_k = S_k^{(L)}$. An order-k path through the first ℓ blocks either takes the skip at block ℓ, leaving an order-k path through the first $\ell - 1$ blocks, or takes the branch, extending an order-$(k-1)$ path through the first $\ell - 1$ blocks by the factor J_ℓ. Block ℓ has the highest index of the first ℓ blocks, so the descending-order product puts J_ℓ on the left:

$$S_k^{(\ell)} = S_k^{(\ell-1)} + J_\ell \, S_{k-1}^{(\ell-1)}, \qquad S_0^{(\ell)} = I, \quad S_k^{(0)} = 0 \;\; (k \ge 1). \tag{8}$$

Filling the table for $\ell = 1, \dots, L$ and $k = 0, \dots, L$ computes every S_k in $O(L^2)$ matrix multiplications, one per (ℓ, k) cell.

For numerical verification of the recurrence, we draw a random Gaussian input and random weights, compute the Jacobian both by the reconstruction $\sum_k \beta^k S_k$ and by automatic differentiation of the forward pass, and confirm the two agree to a relative Frobenius error of order 10^{-15}. Because the masses m_k span many orders of magnitude, we store each S_k as a unit-Frobenius matrix and a log-norm scalar and accumulate in log-sum-exp form. Appendix E gives the same recursion for Post-Norm, whose skip term is the normalization Jacobian, and checks it on every run in Figure 2 (right).

A.2 WHY THE SKIP MUST BE THE IDENTITY UNDER PRE-NORM

We reproduce the argument of He et al. (2016b) that the skip connection should be the identity, through the path decomposition above. Consider a residual network where the skip is scaled by a constant λ, so that each block is $h_\ell = \lambda h_{\ell-1} + \beta f(h_{\ell-1})$ and the Jacobian factor becomes $\lambda I + \beta J_\ell$:

$$\frac{\partial h_L}{\partial h_0} = (\lambda I + \beta J_L)(\lambda I + \beta J_{L-1}) \cdots (\lambda I + \beta J_1). \tag{9}$$

This argument needs the skip path to apply nothing but the scaling λ, which holds for Pre-Norm but not for Post-Norm, whose normalization sits on the skip path. Distributing as before, each factor contributes either λI or βJ_ℓ, so a path of order k carries $\lambda^{L-k}\beta^k$, and the order-0 path that stays on the skip at every block is $\lambda^L I$. This skip path passes through no weights, so its scale λ^L is

independent of the network's weights, at initialization or anywhere in training. The same factor governs both passes: from block ℓ, the skip reaches the output through $L-\ell$ scaled steps, contributing $\lambda^{L-\ell}h_\ell$ to the forward signal h_L and $\lambda^{L-\ell}I$ to the backward Jacobian $\partial h_L/\partial h_\ell$. Because $\lambda^{L-\ell}$ is exponential in depth, any $\lambda \neq 1$ makes the skip path explode ($\lambda > 1$) or vanish ($\lambda < 1$), and β cannot correct it because the skip path carries no factor of β. Only $\lambda = 1$ holds the skip contribution at $O(1)$ across depth, which is why the standard residual block fixes the skip to the identity and leaves β as the only free scale.

A.3 Magnitude growth

Squaring the block $h_\ell = h_{\ell-1} + \beta f(h_{\ell-1})$ gives the magnitude recurrence

$$\|h_\ell\|^2 = \|h_{\ell-1}\|^2 + 2\beta \langle h_{\ell-1}, f(h_{\ell-1})\rangle + \beta^2 \|f(h_{\ell-1})\|^2. \tag{10}$$

Let $g_\ell = \|f(h_{\ell-1})\|/\|h_{\ell-1}\|$ be the branch gain and $\rho_\ell = \langle h_{\ell-1}, f(h_{\ell-1})\rangle/(\|h_{\ell-1}\|\,\|f(h_{\ell-1})\|)$ the alignment between the branch output and the stream. Then

$$\|h_\ell\|^2 = \|h_{\ell-1}\|^2 \,(1 + 2\beta\rho_\ell g_\ell + \beta^2 g_\ell^2). \tag{11}$$

At initialization the two-matrix branch output is uncorrelated with the stream: because W_{down} is zero-mean and independent of $h_{\ell-1}$, the cross term $\rho_\ell g_\ell$ has mean zero and the factor has conditional expectation $1 + \beta^2\,\mathbb{E}[g_\ell^2] \geq 1$. The magnitude therefore grows in expectation at every block, faster at larger β and not at all as $\beta \to 0$, although a single block can shrink it when $\rho_\ell < -\beta g_\ell/2$. Whether the growth is additive or multiplicative depends on the gain. Without normalization the branch output scales with its input, $g_\ell = O(1)$, and the factor compounds to exponential growth in depth. A normalization on the branch input fixes the branch output scale, so $g_\ell = \|f\|/\|h_{\ell-1}\|$ shrinks as the stream grows; the increment $\beta^2\|f\|^2$ is then roughly constant and the stream grows additively in expectation, $\|h_L\|^2 \approx \|h_0\|^2 + \beta^2 \sum_\ell \|f\|^2 = O(L)$.

A.4 The branch-scale symmetry

We show that for residual networks with positively homogeneous activations and biasless branches, networks with the same $\beta\alpha^M$ compute the same function at initialization. Write the branch weights as $W_i = \alpha\bar{W}_i$, for arbitrary matrices $\bar{W}_i$. Positive homogeneity gives $\sigma(\alpha z) = \alpha\,\sigma(z)$ for $\alpha > 0$; applying this identity at each activation, together with the linearity of the matrices, factors the weight scale out of the branch,

$$\beta f(x;\, \alpha\bar{W}_1, \ldots, \alpha\bar{W}_M) = \beta\alpha^M f(x;\, \bar{W}_1, \ldots, \bar{W}_M). \tag{12}$$

Two configurations of (α, β) with the same product $\beta\alpha^M$ therefore compute the same network function, with the same input–output Jacobian and effective rank (Figure 6). In Pre-Norm the normalization acts on the branch input, before the first branch matrix, and in Post-Norm it acts on the stream after the residual addition; in both cases the branch itself is a product of matrices and activations, so the argument applies to either placement. Output-Norm is an exception: its normalization follows the last branch matrix and cancels the factor α^M, so when the normalization ϵ is negligible the network depends on β alone (Appendix B.3).

The equivalence persists throughout training if the optimizer hyperparameters are rescaled accordingly for each matrix. To see why, rescale $\alpha \leftarrow \lambda\alpha$ and $\beta \leftarrow \beta/\lambda^M$ for some $\lambda > 0$, which preserves $\beta\alpha^M$. The forward pass is unchanged, but each branch matrix is λ times larger and its gradient is λ times smaller: the gradient with respect to one matrix carries the branch scale β/λ^M and the other $M-1$ matrices, which contribute λ^{M-1}. For the rescaled network to follow the same trajectory, its matrices must remain λ times larger after every update, so the learning rate must be rescaled in accordance with how the gradient changed: $\eta \leftarrow \lambda^2\eta$ for SGD, whose update is proportional to the gradient, and $\eta \leftarrow \lambda\eta$ for Adam, whose update is invariant to the gradient scale. Similar symmetries were observed for MLP layers in Yang & Hu (2021); Yang & Littwin (2023) and for the Pre-Norm residual stream in Blake et al. (2025). The rescaling must extend to other optimizer hyperparameters, including ϵ in Adam, weight decay, and gradient clipping (Yang & Littwin, 2023). This equivalence holds exactly only in infinite precision and infinite dynamic range.

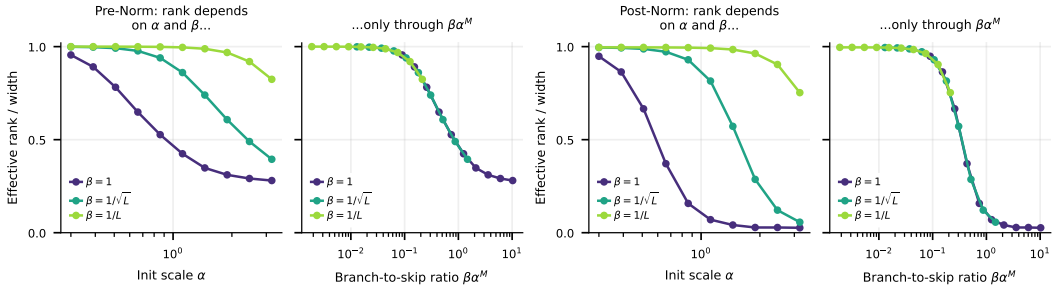

Figure 6: **The effective rank at initialization depends on the branch scale β and the initialization scale α only through the nominal branch-to-skip ratio $\beta\alpha^M$ (M the number of branch matrices), for both normalization placements.** Effective rank of the input–output Jacobian, normalized by the width, for a residual MLP at initialization ($d = 256$, $L = 48$, two square $d \times d$ branch matrices, single seed). *(a)* Pre-Norm: the rank depends on both the initialization scale α and the branch scale β. *(b)* Plotted against $\beta\alpha^M$, the β-sweeps collapse onto a single curve. *(c, d)* The same holds for Post-Norm, but the rank collapses further at a given ratio.

B Normalization Placement: Additional Results and Derivations

B.1 Block Jacobians and the normalization projection

We consider a residual block that may contain a normalization operation in each of three locations (Kim et al., 2025): the branch input (P), the branch output (B), and the residual stream after the addition (R). With branch f, the block maps

$$h_\ell = R\big(h_{\ell-1} + \beta\, B(f(P(h_{\ell-1})))\big). \tag{13}$$

By the chain rule its Jacobian factorizes as

$$\frac{\partial h_\ell}{\partial h_{\ell-1}} = J_R\left(I + \beta\, J_B J_f J_P\right), \tag{14}$$

where each J is the Jacobian of its operation at the relevant point. The placement determines where a normalization Jacobian lands: a branch operation (P or B) sits inside the β-scaled branch term, whereas the post-residual operation R multiplies the entire sum, the skip included. Only R applies its Jacobian to the identity path. Appendix B.2 shows that its projection is not what collapses the rank in the networks we study, and Appendix B.3 shows how the placement changes the branch-to-skip ratio across depth.

The RMSNorm Jacobian is a scaled projection. With its gain at the standard initialization of one, RMSNorm is $N(x) = \sqrt{d}\, x/\|x\|$, and

$$J_N = \frac{\partial N}{\partial x} = \frac{\sqrt{d}}{\|x\|}\big(I - \hat{x}\hat{x}^\top\big), \qquad \hat{x} = \frac{x}{\|x\|}. \tag{15}$$

The factor $\sqrt{d}/\|x\|$ is the inverse stream scale (the root-mean-square of x is $\|x\|/\sqrt{d}$); the projection $I - \hat{x}\hat{x}^\top$ removes the current stream direction $\hat{x}$.

The detached rescaling separates the scale from the projection. To separate the effect of the projection term from the effect of the scalar rescaling, we define a *detached rescaling* (Xu et al., 2019) that divides by the stream scale exactly as RMSNorm does but stops the gradient through that scale: it computes $x/\,\mathrm{sg}(\mathrm{rms}(x))$, where sg is the stop-gradient. The forward pass is identical to RMSNorm, but the scale is now a constant, so the Jacobian is

$$J_{\text{detach}} = \frac{\sqrt{d}}{\|x\|}\, I, \tag{16}$$

the same inverse scale with no projection. It rescales the stream but removes no direction, which is what makes it the control that isolates the rescaling from the projection (Appendix B.2).

The four post-residual operations. In Table 2 we summarize the four operations we use in the post-residual placement. They let us vary the two factors of a normalization Jacobian independently: the scalar rescaling $\sqrt{d}/\|x\|$, which normalizes the stream to unit scale, and the projection $I - (\cdot)(\cdot)^\top$, which removes a direction. RMSNorm does both; the detached rescaling keeps only the rescaling; mean subtraction keeps only the projection, of a fixed direction and without rescaling; LayerNorm adds mean removal to RMSNorm. Comparing them isolates the contribution of each factor (Appendix B.2).

Table 2: Jacobians of the post-residual operations at gain one. $\hat{x} = x/\|x\|$; $\hat{x}_c$ is the unit centered vector; $\mathbf{1}$ is the all-ones direction.

operation	Jacobian	rescales?	removes
RMSNorm	$\frac{\sqrt{d}}{\|x\|}\big(I - \hat{x}\hat{x}^\top\big)$	yes	radial $\hat{x}$
detached rescale	$\frac{\sqrt{d}}{\|x\|}\, I$	yes	nothing
mean subtraction	$I - \frac{1}{d}\mathbf{1}\mathbf{1}^\top$	no	fixed $\mathbf{1}$
LayerNorm	$\frac{\sqrt{d}}{\|x-\mu\|}\big(I - \hat{x}_c\hat{x}_c^\top\big)\big(I - \frac{1}{d}\mathbf{1}\mathbf{1}^\top\big)$	yes	radial $+\,\mathbf{1}$

B.2 The projection is not the primary cause of rank loss

In Section 4 we observed that the Jacobians of the normalization operations contain projection terms that remove a direction from the gradient: mean subtraction removes the all-ones direction $\mathbf{1}$, and rescaling removes the current stream direction $\hat{x} = h/\|h\|$,

$$\frac{\partial}{\partial h}\big(h - \mu(h)\mathbf{1}\big) = I - \tfrac{1}{d}\mathbf{1}\mathbf{1}^\top, \qquad \frac{\partial}{\partial h}\,\frac{h}{\operatorname{rms}(h)} = \frac{\sqrt{d}}{\|h\|}\big(I - \hat{x}\hat{x}^\top\big). \tag{17}$$

Composed over L layers, these projections could in principle compound into a low-rank Jacobian, which makes the projection terms a natural candidate for explaining the rank collapse (Bachlechner et al., 2021). Here we give the evidence behind the main-text statement that the projection is not needed for the collapse in the networks we study: removing it changes the effective rank by at most 0.4% of the width, and the branch-to-skip ratio governs the rate of rank loss (Appendix B.3).

The projection is not necessary for rank collapse. We use the detached rescaling operation to ablate the projection term (Eq. 16), which lets us test whether the projection is necessary for the rank collapse to occur. Even with the projection removed, the detached rescaling collapses the rank just like ordinary RMSNorm. Across $\alpha \in \{0.5, 1, 2\}$, $L \in \{48, \ldots, 384\}$, and $\beta \in \{1,\ 1/\sqrt{L}\}$, with and without the Pre-RMSNorm, the detached and ordinary ranks differ by at most 0.4% of the full rank (largest 0.0039, at α=0.5 and β=$1/\sqrt{L}$, where the rank is near full). Figure 7b overlays the detached and ordinary curves across depth. The projection is therefore not necessary for the collapse in these networks.

Isolating the post-residual operation. Figure 7b adds a Pre-RMSNorm to every configuration so that the post-residual operations (none, mean subtraction, RMSNorm and the detached rescaling) are compared on a normalized branch input, varying only the operation. Without a Pre-RMSNorm to control the magnitude, the configurations with no post-residual operation or with mean subtraction have no normalization on the branch at all: their branch-to-skip ratio stays constant, so they collapse at nearly the rate of the rescaling operations, and none, mean subtraction, RMSNorm and LayerNorm become indistinguishable (Figure 8a). To show that this Pre-RMSNorm does not confound our results, Figure 8b shows that with or without it the post-RMSNorm and post-LayerNorm curves coincide.

B.3 The depth law: the ratio across depth, contour extraction, and width independence

The branch-to-skip ratio. The branch-to-skip ratio (Eq. 3) is the Frobenius norm of the branch term in the block Jacobian (Eq. 14) relative to that of the identity skip. At initialization the branch term $\beta\, J_B J_f J_P$ collects three factors: the explicit branch scale β; a factor α^M from the M branch

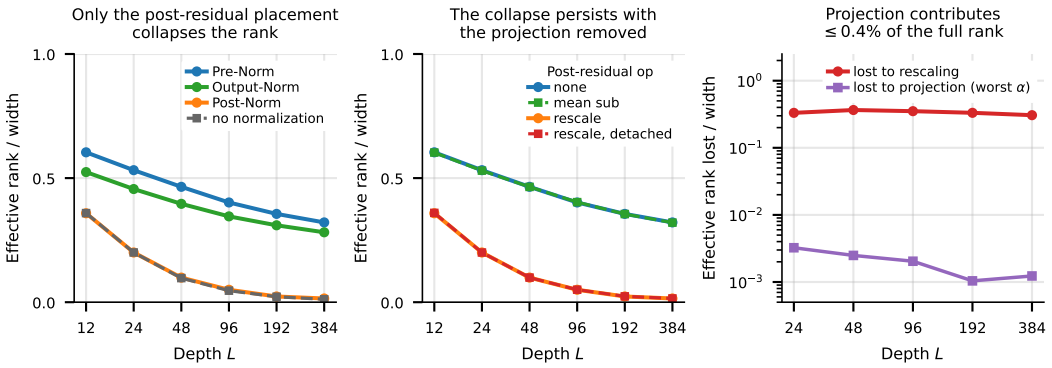

Figure 7: **The rank collapse does not need the projection terms in the normalization Jacobian.** All panels: residual networks at initialization with a two-matrix ReLU branch (both matrices $d \times d$, d=256), β=1, one sample of weights and input; effective rank of the input–output Jacobian, normalized by width. Panels (a) and (b) fix α=1. **(a)** Of the three placements, only the post-residual one collapses rank. A normalization at either branch placement, the input (Pre-Norm) or the output (Output-Norm), loses rank slowly; the post-residual placement (Post-Norm) collapses at the same rate as no normalization at all. **(b)** The collapse persists with the projection removed. A detached rescaling (fixes the scale, removes no direction) collapses identically to RMSNorm (legend: rescale); mean subtraction (removes a direction, does not rescale) matches the network with no post-residual operation. Every configuration uses a Pre-RMSNorm; we vary only the post-residual operation, and the rescaling collapse does not depend on it (Figure 8). **(c)** Removing the projection changes the rank little. The plot compares networks that differ only in the post-residual operation: relative to no post-residual operation, the detached rescaling loses almost all the rank that RMSNorm loses (shown at α=1), and the two differ by at most 0.4% of the full rank at the worst $\alpha \in \{0.5, 1, 2\}$.

weight matrices, each initialized at scale α, so the branch Jacobian J_f scales as α^M; and the inverse stream scale $\sqrt{d}/\|h_{\ell-1}\|$ contributed by the branch-input normalization in Pre-Norm (Eq. 15). Their product is

$$\tilde{r}_\ell = \frac{\beta\alpha^M\sqrt{d}}{\|h_{\ell-1}\|} = \frac{\beta\alpha^M}{\|h_{\ell-1}\|/\sqrt{d}}, \tag{18}$$

the nominal ratio $\beta\alpha^M$ divided by the stream scale $\mathrm{rms}(h_{\ell-1}) = \|h_{\ell-1}\|/\sqrt{d}$. The ratio $\tilde{r}_\ell$ omits the activation-derivative factor $\sqrt{p}$ of the Frobenius ratio in Eq. 3: at initialization $r_\ell^2 = p\,\tilde{r}_\ell^2$ in expectation (Section 3), and the factor is shared by every block and both placements. The nominal ratio $\beta\alpha^M$ is $\tilde{r}_\ell$ at unit stream scale ($\|h_{\ell-1}\| = \sqrt{d}$); the per-layer ratio scales inversely with how far the stream has grown. All of the depth dependence enters through $\|h_{\ell-1}\|$: a post-residual rescaling resets it to $\sqrt{d}$ at every block, whereas the branch normalizations leave the stream to grow with depth.

Output-Norm cancels the initialization scale. For a positively homogeneous branch, $f_\alpha(h) = \alpha^M f_1(h)$, and an RMS normalization of the branch output removes this factor: $\mathrm{Norm}(f_\alpha(h)) = \mathrm{Norm}(f_1(h))$ for every h, so the normalized branch and its Jacobian are the same for every α when the normalization ϵ is negligible. The $\beta\alpha^M$ dependence derived here therefore applies to Pre-Norm and Post-Norm only. The dependence on the stream scale remains: each branch increment has squared norm $\beta^2 d$, so $\mathbb{E}\|h_\ell\|^2 = \mathbb{E}\|h_0\|^2 + \beta^2 d\,\ell$, and the ratio decays as $1/\sqrt{\ell}$, as for Pre-Norm (Eq. 19 below).

Residual stream growth. When the skip connection is an identity (which occurs whenever there is no post-residual normalization), the residual stream is a running sum $h_\ell = h_{\ell-1} + \beta u_\ell$ of branch outputs u_ℓ. The branch input is normalized, so each u_ℓ has roughly fixed magnitude and, at initialization, is nearly orthogonal to the accumulated stream (two generic vectors in d dimensions have normalized inner product $O(1/\sqrt{d})$). Expanding,

$$\|h_\ell\|^2 = \|h_{\ell-1}\|^2 + 2\beta\langle h_{\ell-1}, u_\ell\rangle + \beta^2\|u_\ell\|^2,$$

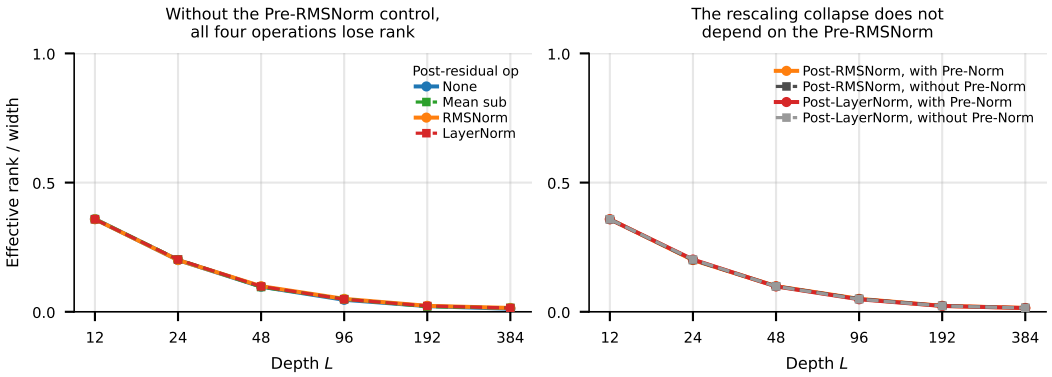

Figure 8: **Without the Pre-RMSNorm control, the unnormalized configurations collapse and mask the operation comparison.** Effective rank of the input–output Jacobian across depth at α=β=1 (one sample of weights and input), as in Figure 7b but with no Pre-RMSNorm and with LayerNorm in place of the detached rescaling. **(a)** With no operation or mean subtraction alone, there is no normalization on the branch: the branch-to-skip ratio stays constant and the rank collapses at nearly the rate of the rescaling operations, so the four configurations are indistinguishable. **(b)** The rescaling collapse does not depend on the control: the post-RMSNorm and post-LayerNorm curves with and without the Pre-RMSNorm coincide.

the cross term vanishes exactly in expectation, since W_{down} is zero-mean and independent of the rest of the branch, and the increment $\beta^2\|u_\ell\|^2$ is constant in expectation. The squared length therefore grows linearly in expectation (the magnitude recurrence in Appendix A.3): $\mathbb{E}\|h_\ell\|^2 = d\,(1+\kappa(\beta\alpha^M)^2\ell)$ for an input with $\|h_0\|^2 = d$. The coefficient is $\kappa = \mathbb{E}[\sigma(z)^2]$, the second moment of the activation values on a unit-variance input, equal to $\frac{1}{2}$ exactly for ReLU. Block ℓ's branch sees the stream $h_{\ell-1}$; replacing $\|h_{\ell-1}\|^2$ by its expectation, which is accurate when the norm concentrates at large d, the branch-to-skip ratio is

$$\tilde{r}_\ell \approx \frac{\beta\alpha^M}{\sqrt{1+\kappa(\beta\alpha^M)^2(\ell-1)}}, \tag{19}$$

which starts near $\beta\alpha^M$ and decays as $1/\sqrt{\ell}$ as the stream grows, whereas a post-residual rescaling resets $\|h_\ell\| = \sqrt{d}$ every block and holds $\tilde{r}_\ell$ constant at $\beta\alpha^M$. The measured value, $\kappa \approx 0.51$ for a ReLU branch, matches the derivation: $\mathrm{rms}(h_\ell)^2 - 1$ collapses onto a single line in $(\beta\alpha^M)^2\ell$ (Figure 9). κ is a moment of the activation values $\sigma(z)$, which determine the branch outputs accumulating on the stream. The branch Jacobians are controlled by the derivative moment $p = \mathbb{E}[\sigma'(z)^2]$ (Section 3). For ReLU both moments equal $\frac{1}{2}$, by the Gaussian identity $\mathbb{E}[z^2\mathbf{1}(z>0)] = P(z>0)$. At L=384, α=β=1, the stream reaches $\mathrm{rms}(h_L) \approx 14.5$ for Pre-Norm and ≈ 19.7 for Output-Norm, versus 1.00 for Post-Norm.

Networks with post-residual rescaling behave like networks with no normalization. For a degree-1 positively-homogeneous activation such as ReLU, $f(ch) = cf(h)$ for $c > 0$, so the branch Jacobian is scale-invariant: $J_f(ch) = J_f(h)$. This has two consequences. First, a network with no normalization keeps the same stream *direction* as Post-Norm: by induction, if $h_{\ell-1}^{\text{none}} = s_{\ell-1}\,\hat{h}_{\ell-1}^{\text{post}}$ for a positive scalar $s_{\ell-1}$, homogeneity makes the branch outputs match up to $s_{\ell-1}$, so h_ℓ^{none} is again a scalar multiple of the Post-Norm sum. The two placements feed their branches the same direction, hence the same J_f. Second, the detached rescaling is the no-norm Jacobian up to a per-layer scalar: its block Jacobian is $\frac{\sqrt{d}}{\|\cdot\|}(I + \beta J_f)$ (Eq. 16), a scalar times the no-norm block Jacobian $I + \beta J_f$. The scalars factor out of the product and do not affect the rank, so the detached rescaling and no normalization have identical Jacobian effective rank.

Ordinary Post-Norm adds the projection $I - \hat{x}\hat{x}^\top$ to each block (Eq. 15), the only difference from the detached and no-norm cases. Removing that projection (the detached rescaling, Appendix B.2) changes the effective rank by at most 0.4% of the width, so the no-norm, detached, and Post-Norm ranks all coincide to within that margin, and $\tilde{r}_\ell$ is constant at $\beta\alpha^M$ for all three. The degree-1

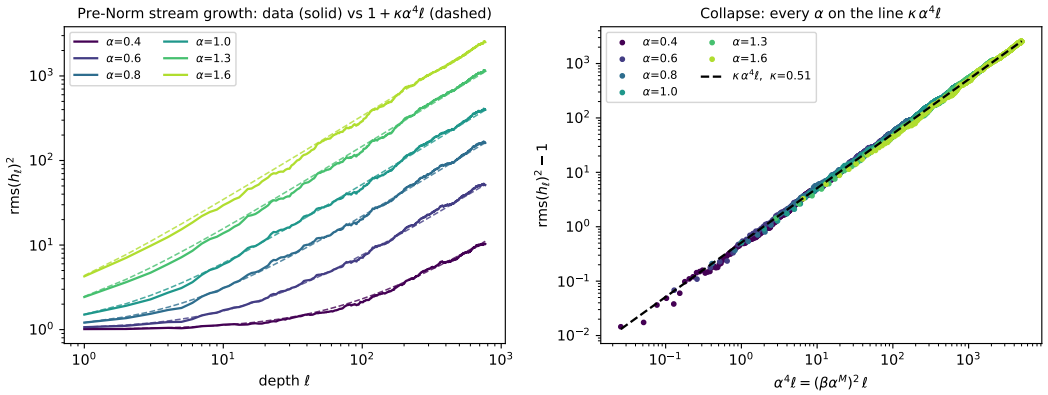

Figure 9: **At initialization,** $\mathrm{rms}(h_\ell)^2$ **grows like** $1+\kappa(\beta\alpha^M)^2\ell$ **with** $\kappa = \mathbb{E}[\sigma^2] = \frac{1}{2}$ **for ReLU; one fitted** $\kappa \approx 0.51$ **describes every initialization scale. (a)** $\mathrm{rms}(h_\ell)^2$ versus depth for a Pre-Norm two-matrix ReLU branch (both matrices $d \times d$, d=256; β=1; L=768; one sample of weights and input), shown as the measured stream (solid) against the fit (dashed); the stream grows faster at larger α. **(b)** Plotted against $(\beta\alpha^M)^2\ell$, $\mathrm{rms}(h_\ell)^2 - 1$ collapses onto the line $\kappa(\beta\alpha^M)^2\ell$ with $\kappa \approx 0.51$ for all α, consistent with $\mathbb{E}\|h_\ell\|^2 = d(1+\kappa(\beta\alpha^M)^2\ell)$.

homogeneity is essential: a degree-2 activation would make J_f scale as $\|h\|$, so the no-norm ratio would grow with depth instead of staying constant.

The $\sqrt{L}$ depth factor from path counting. The path decomposition in Section 3 (Eq. 4) writes the input–output Jacobian as a sum over path orders, where the order-k paths pass through the branch at k blocks and through the skip everywhere else. For Pre-Norm the decomposition applies directly; for Post-Norm we first drop the post-residual norm Jacobians, whose scalar factors do not change the effective rank and whose projections empirically contribute at most 0.4% of the width (Appendix B.2). Relative to the Frobenius norm $\sqrt{d}$ of the identity skip, a path that visits the branch at blocks $\ell_1 < \cdots < \ell_k$ has Frobenius norm $\tilde{r}_{\ell_1} \cdots \tilde{r}_{\ell_k}$, the product of the branch-to-skip ratios at the visited blocks, up to an order-one constant per factor that is the same for every α, β, and L.

At initialization, the branch Jacobians at different blocks have independent weights, so distinct paths point in nearly uncorrelated directions and their squared norms approximately add. The order-k path sum then has relative Frobenius norm

$$\Big(\sum_{\ell_1<\cdots<\ell_k} \tilde{r}_{\ell_1}^2 \cdots \tilde{r}_{\ell_k}^2\Big)^{1/2} \;\approx\; \frac{\big(\sum_\ell \tilde{r}_\ell^2\big)^{k/2}}{\sqrt{k!}}, \tag{20}$$

where the approximation holds when no few blocks dominate the sum, $k^2 \max_\ell \tilde{r}_\ell^2 \ll \sum_\ell \tilde{r}_\ell^2$ (for equal ratios, $k^2 \ll L$). For Post-Norm, $\tilde{r}_\ell = \beta\alpha^M$ at every block, so $\sum_\ell \tilde{r}_\ell^2 = (\beta\alpha^M)^2 L$ and the order-k sum has norm $(\beta\alpha^M\sqrt{L})^k/\sqrt{k!}$. Each branch visit multiplies a path's norm by $\beta\alpha^M$ and multiplies the number of available paths by roughly L; a sum of N randomly oriented terms of equal norm grows as $\sqrt{N}$ times that norm, so each visit contributes one factor of $\beta\alpha^M\sqrt{L}$. For Pre-Norm, the ratios decay following Eq. 19, so

$$\sum_{\ell=1}^{L} \tilde{r}_\ell^2 \;\approx\; \int_0^L \frac{(\beta\alpha^M)^2}{1+\kappa(\beta\alpha^M)^2\ell}\, d\ell \;=\; \frac{1}{\kappa}\log\big(1+\kappa(\beta\alpha^M)^2 L\big), \tag{21}$$

again a function of $\beta\alpha^M\sqrt{L}$ alone, since $(\beta\alpha^M)^2 L = (\beta\alpha^M\sqrt{L})^2$. This holds when $\kappa(\beta\alpha^M)^2 \lesssim 1$; at larger $\beta\alpha^M$ the first blocks dominate the sum ($\tilde{r}_1 = \beta\alpha^M$), and the Pre-Norm rank separates slowly with depth (Figure 10d). In the Frobenius ratio $r_\ell = \sqrt{p}\,\tilde{r}_\ell$, each order carries an additional factor $(\sqrt{p})^k$ and Eq. 21 a prefactor p/κ. These rescale $\beta\alpha^M\sqrt{L}$ by the constant $\sqrt{p}$ for Post-Norm and change only the prefactor for Pre-Norm; neither changes the $\sqrt{L}$ scaling.

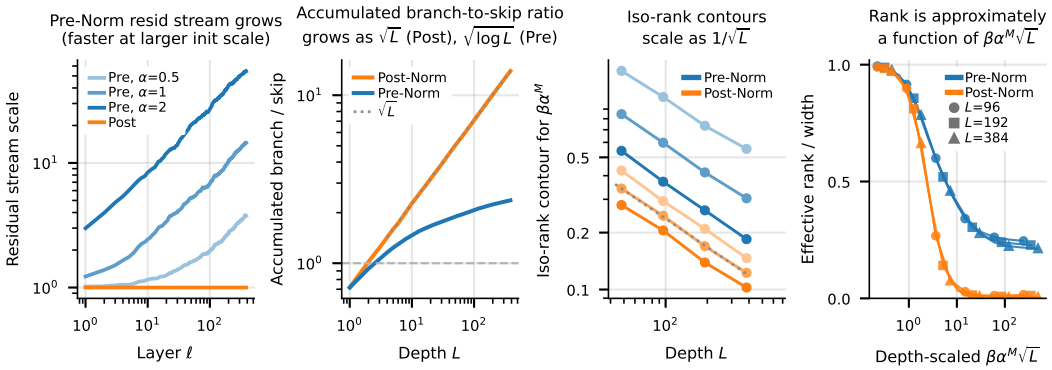

Figure 10: **Over the tested depths, the effective rank is approximately a function of the product of the nominal branch-to-skip ratio $\beta\alpha^M$ (an exact symmetry) and a $\sqrt{L}$ depth factor (empirical).** Post-Norm collapses, while Pre-Norm decays slowly. All panels at initialization with a two-matrix ReLU branch (both matrices $d \times d$, d=256), β=1, one sample of weights and input; $\beta\alpha^M$ is the nominal branch-to-skip ratio (M branch matrices). **(a)** Residual stream scale by layer ($L = 384$), which sets the ratio: the Pre-Norm residual stream grows with depth, faster at larger initialization scale α, so the ratio decays; Post-Norm rescales the residual stream to unit scale, which holds the ratio constant. **(b)** The accumulated branch norm (the order-one path sum $\| \sum_\ell \beta J_\ell \|_F$) relative to the skip norm $\sqrt{d}$ grows as $\sqrt{L}$ for Post-Norm (a random walk of constant-magnitude increments) and only as $\sqrt{\log L}$ for Pre-Norm (a random walk of shrinking increments, Eq. 21). At L=384 the accumulated branch norm reaches $14\times$ the skip for Post-Norm and $2.4\times$ for Pre-Norm. **(c)** The rank is an approximate function of $\beta\alpha^M\sqrt{L}$: for both placements, the iso-rank contours (rank levels 0.4, 0.5, 0.6, lighter to darker) scale as $1/\sqrt{L}$ (fit over $L \in \{48, 96, 192, 384\}$). **(d)** Plotted against $\beta\alpha^M\sqrt{L}$, the curves for $L = 96$, 192 and 384 fall approximately onto one curve per placement.

Under three assumptions, which we state below along with measurements that test them, every order of the expansion depends on (α, β, L) only through $\beta\alpha^M\sqrt{L}$; since the effective rank is scale-invariant, it is then a function of this single variable for each placement, and the iso-rank contours scale as $\beta\alpha^M \propto 1/\sqrt{L}$. First, the count $\binom{L}{k} \approx L^k/k!$ requires $k^2 \ll L$; near the transition at $\beta\alpha^M\sqrt{L} \sim 1$ the contributing orders are small, so this condition is mild. Second, distinct paths are treated as uncorrelated. Each branch Jacobian ends in a zero-mean output matrix, which suppresses correlations between paths, but the activation masks depend on the shared stream, so weak correlations remain; the measured exponents test the net effect, since coherently adding paths would grow the order-k sum as $(\beta\alpha^M L)^k/k!$ and give iso-rank contours scaling as $1/L$, whereas the measured contour slopes lie between -0.46 and -0.57 (Table 3) and the Post-Norm order-one sum grows with log–log slope 0.50 (Figure 10b), matching the uncorrelated prediction. Third, the Frobenius norms of the path sums do not by themselves determine the rank: we additionally assume that the path sums, normalized by their Frobenius norms, have distributions that do not otherwise depend on α, β, or L (their spectra and their alignments with one another and with the identity). For homogeneous activations like ReLU, α and β enter each path as the overall factor $(\beta\alpha^M)^k$ and, through the stream directions that set the activation masks, only via $\beta\alpha^M$ (Appendix A.4), so this assumption concerns how the normalized path sums depend on $\beta\alpha^M$ and L separately; the collapse of the measured rank curves onto a single function of $\beta\alpha^M\sqrt{L}$ (Figure 10d) shows that any further dependence on α, β, or L is small over the tested range. We do not attempt a closed form for the shape of the rank curve.

We measure the depth law in Figure 10 as follows. The branch-scale symmetry (Figure 6) lets us vary a single quantity: we fix $\beta = 1$ and sweep the initialization scale α, so that $\beta\alpha^M = \alpha^M$. At each depth L and each rank level $r^\star$, we extract the *iso-rank contour* — the value of $\beta\alpha^M$ at which the effective rank equals $r^\star$ — by interpolating $\log \beta\alpha^M$ against the measured rank. Fitting $\log \beta\alpha^M$ against $\log L$ then gives the contour's power-law exponent.

The exponent is close to $-\frac{1}{2}$ across rank levels and both placements (Table 3), consistent with the main-text observation that $\beta = 1/\sqrt{L}$ holds $\langle k \rangle$ fixed across depth. The Post-Norm contours are

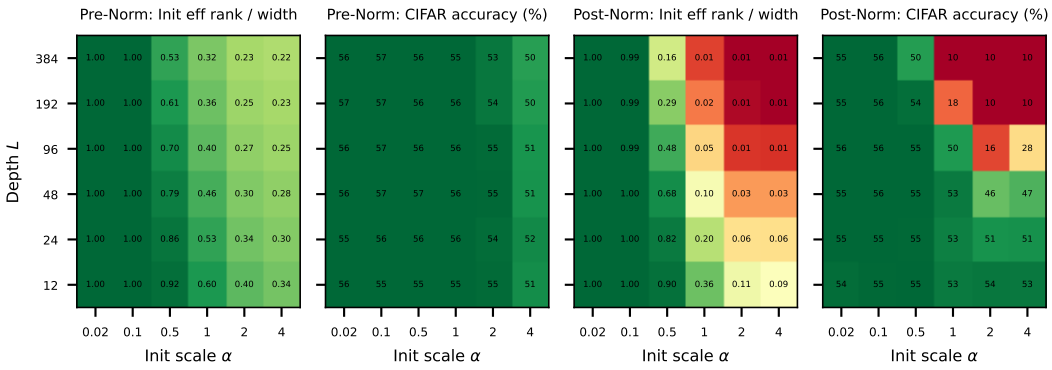

Figure 11: **Training failures on CIFAR-10 coincide with rank collapse at initialization.** Heatmaps over initialization scale α and depth L, for residual networks with a two-matrix ReLU branch (both matrices $d \times d$, d=256) and β=1. **(a,c)** Effective rank of the input–output Jacobian at initialization (one sample of weights and input). **(b,d)** Best CIFAR-10 test accuracy over a learning-rate sweep with Adam. For Pre-Norm **(a,b)**, the rank stays above 0.2 and training succeeds across the entire grid. For Post-Norm **(c,d)**, the rank collapses at large α and depth, and accuracy falls below 40%, to chance (10%) at the largest depths, in the same region.

the cleanest, with slopes between -0.49 and -0.52; the Pre-Norm contours are noisier because the Pre-Norm rank changes slowly at large α, which flattens the rank–α curve and makes the crossing harder to locate, but they remain near -0.5.

A wider branch does not change the exponent. Repeating the fit with a $4\times$ width expansion (a hidden dimension four times the model dimension, both branches having $M = 2$ matrices) leaves the slopes near -0.5 (Table 3, $4\times$). The width raises the rank at large $\beta\alpha^M$ at every depth, through the higher effective rank of the wider branch Jacobian (Section 5), but the depth scaling is unchanged, so the $\sqrt{L}$ factor is a property of depth rather than of the per-block branch.

Table 3: Fitted exponent s of the iso-rank contour $\beta\alpha^M \propto L^s$, for both placements, three rank levels $r^\star$, and two width expansions ($1\times$, two $d \times d$ matrices as in the other figures of this appendix; $4\times$, a four-times-wider hidden dimension), with one sample of weights and input per network. All are near $s = -\frac{1}{2}$.

placement	width	$r^\star$=0.4	$r^\star$=0.5	$r^\star$=0.6
Pre-Norm	$1\times$	-0.46	-0.50	-0.52
Pre-Norm	$4\times$	-0.57	-0.47	-0.48
Post-Norm	$1\times$	-0.51	-0.50	-0.49
Post-Norm	$4\times$	-0.52	-0.50	-0.50

B.4 Trainability: the range of stable learning rates

The trainability grid in Figure 11 reports the best CIFAR-10 test accuracy over an Adam (Kingma & Ba, 2015) learning-rate sweep (η from 3×10^{-5} to 10^{-2}) at each initialization scale α and depth L. We train with Adam for 5000 steps on minibatches of 256 flattened, channel-standardized images, with a 500-step linear warmup to a constant learning rate and no weight decay, and evaluate test accuracy on the full test set every 500 steps; a run's best test accuracy is the maximum over these ten evaluations, and each configuration is a single run. Because the evaluation step and the learning rate are both selected on the test set, these accuracies measure trainability rather than generalization. We say that a run trains if its test accuracy reaches 0.40 at some evaluation, that is, if its best test accuracy is at least 0.40 (chance is 0.10), even if the accuracy falls afterwards, and we define the maximum stable learning rate as the largest η at which the network trains. Table 4 reports the maximum stable learning rate across depth.

Pre-Norm is able to train with all learning rate values in our sweep (up to $\eta = 10^{-2}$) at every depth and every α, consistent with its slowly decaying rank. The stable learning rate range for Post-Norm instead narrows with depth, and faster at larger α (Table 4). Changing the placement or the depth changes more than the rank, so this comparison is correlational.

The range of stable learning rates narrows before every learning rate fails. At $\alpha = 0.5$ the maximum stable learning rate falls by more than two orders of magnitude across depth (from 10^{-2} to 3×10^{-5}), yet the best accuracy over the sweep stays near or above 0.50 at every depth (0.4985 at $L = 384$): the network still trains at a small enough learning rate. At $\alpha = 1$ the maximum stable learning rate falls thirtyfold by depth 96, and the best accuracy first falls below 0.40 at depth 192 (0.18), reaching chance at depth 384. The range of stable learning rates therefore narrows at smaller depths than those at which every learning rate fails.

Table 4: $\log_{10}$ of the maximum stable Adam learning rate (the largest η whose best CIFAR-10 test accuracy reaches 0.40; chance is 0.10) versus depth, at $\beta = 1$, from the sweep behind Figure 11. Pre-Norm stays at the top of the sweep at every depth; Post-Norm's stable range narrows with depth, faster at larger α. A dash marks a configuration that does not reach 0.4 at any learning rate in the sweep (3×10^{-5} to 10^{-2}).

placement	α	L=12	24	48	96	192	384
Pre-Norm	all	-2	-2	-2	-2	-2	-2
Post-Norm	0.02	-2	-2	-2	-2	-2	-2.5
Post-Norm	0.1	-2	-2	-2	-2	-2.5	-2.5
Post-Norm	0.5	-2	-2	-2.5	-3	-3.5	-4.5
Post-Norm	1	-2.5	-3	-3.5	-4	—	—
Post-Norm	2	-2.5	-3.5	-4	—	—	—
Post-Norm	4	-2	-3	-4	—	—	—

C Branch Coherence: Derivations and Measurements

The cross-layer correlation approaches c in the spiked regime. We derive the limit $\cos(b_\ell, b_{\ell'}) \to c = \mathbb{E}[\sigma]^2/\mathbb{E}[\sigma^2]$, the activation moment ratio in Section 5, for a deep network at initialization, in the regime where the mean spike dominates the residual stream. The branch mean at layer ℓ is $b_\ell = \mathbb{E}_x[\sigma(W_\ell z)]$, the average of the branch output over inputs, with z the Pre-Normed branch input and W_ℓ isotropic. Once the spike dominates, the representations of all inputs have collapsed onto its direction, which is the all-ones direction $\mathbf{1}$: an uncentered activation gives every coordinate of the branch output the same positive mean, so each branch mean carries a $\mathbf{1}$-component, and these accumulate in the stream mean $\mu_\ell = b_1 + \cdots + b_\ell$ until $\cos(\mu_\ell, \mathbf{1}) \to 1$ (derived below). After the Pre-RMSNorm every input then has $z \approx \mathbf{1}$, independent of x, so the pre-activation of coordinate i is

$$(W_\ell z)_i \;\approx\; (W_\ell \mathbf{1})_i \;=:\; g_i^{(\ell)}.$$

The rows of W_ℓ are independent and the Pre-Norm fixes the scale, so the $g_i^{(\ell)}$ are i.i.d. standard Gaussian and independent across layers. The pre-activation no longer depends on x, so each branch mean is a deterministic function of that layer's weights, $b_\ell[i] \approx \sigma(g_i^{(\ell)})$. The dot product and the norms then concentrate over the d coordinates, and because $g^{(\ell)}$ and $g^{(\ell')}$ are independent for $\ell \neq \ell'$,

$$b_\ell \cdot b_{\ell'} = \sum_i \sigma(g_i^{(\ell)})\,\sigma(g_i^{(\ell')}) \;\approx\; d\,\mathbb{E}[\sigma]^2, \qquad \|b_\ell\|^2 = \sum_i \sigma(g_i^{(\ell)})^2 \;\approx\; d\,\mathbb{E}[\sigma^2],$$

with $\mathbb{E}[\sigma]$, $\mathbb{E}[\sigma^2]$ the moments of σ under a standard Gaussian. Hence

$$\cos(b_\ell, b_{\ell'}) \;\to\; \frac{\mathbb{E}[\sigma]^2}{\mathbb{E}[\sigma^2]} \;=\; c.$$

The cross term factors into $\mathbb{E}[\sigma]^2$ because the two layers' pre-activations are independent: only the part common to every weight draw — the all-ones component, with per-coordinate value $\mathbb{E}[\sigma]$ — survives the product, and it carries a fraction c of the energy $\mathbb{E}[\sigma^2]$. This is the same c that bounds the

Table 5: **The moment ratio c determines the per-layer mean-spike rank, and the bound is nearly tight at initialization.** For each pointwise activation σ, the moment ratio $c = \mathbb{E}[\sigma]^2/\mathbb{E}[\sigma^2]$ (standard Gaussian pre-activation) determines the stable-rank bound $s_\sigma = 1/c$ of Davis & Drusvyatskiy (2025) for the post-activation matrix, which holds at unit pre-activation scale up to a factor $1 + o(1)$ with high probability; they compute the ReLU and absolute-value cases ($s_\sigma = \pi$ and $\pi/2$). Uncentered activations ($c > 0$) carry a low-rank mean spike; the measured post-activation stable rank $\mathrm{st}(\sigma(Wh))$ ($\mathrm{st}(A) = \|A\|_F^2/\|A\|_2^2$, the ratio of total to top-direction energy) for isotropic Gaussian inputs at initialization (d=256, n=512, mean over 3 seeds) is within 6% of s_σ. Centered activations have c=0 and no mean-induced spike (s_σ=∞), and the stable rank stays high.

Activation σ	Centered?	Moment ratio $c = \mathbb{E}[\sigma]^2/\mathbb{E}[\sigma^2]$	Spike bound $s_\sigma = 1/c$	Measured stable rank
abs	✗	$2/\pi \approx 0.637$	$\pi/2 \approx 1.57$	1.56
ReLU	✗	$1/\pi \approx 0.318$	$\pi \approx 3.14$	3.09
GELU	✗	0.187	5.34	5.16
SiLU	✗	0.120	8.33	7.85
tanh	✓	0	∞	50.66

per-layer stable rank, $s_\sigma = 1/c$ (Table 5): with a single matrix, the per-layer spike and the cross-layer alignment are set by the same ratio. The collapse $z \approx \mathbf{1}$ and the concentration of the sums are leading-order statements, exact as the spike grows and $d \to \infty$. In the first layers, before the spike dominates, the inputs are still spread out, and averaging over them makes each branch mean nearly proportional to $\mathbf{1}$, so consecutive means align more strongly than c: for ReLU the cosine is 0.92, 0.74 and 0.62 at the first three layers and settles near $1/\pi$ beyond layer 100 (Figure 12a); for GELU, not shown in the figure, it settles at 0.188 ± 0.031 (mean and standard deviation over the layers beyond 100), against $c = 0.187$.

Anchor checks. The decomposition $b_\ell = \mathbb{E}[\sigma]\,\mathbf{1} + \varepsilon_\ell$, with ε_ℓ a zero-mean remainder independent across layers, makes two predictions. At initialization (d=256, L=512, n=4096, 3 seeds), each branch mean aligns with the all-ones direction at $\cos(b_\ell, \mathbf{1}) \to \sqrt{c}$ as depth grows, measured 0.565 for ReLU and 0.433 for GELU against $\sqrt{1/\pi} = 0.564$ and $\sqrt{0.187} = 0.432$; the $\mathbf{1}$-component therefore carries a $\cos^2 = c$ fraction of the energy. The accumulated mean then aligns fully, $\cos(\mu_\ell, \mathbf{1}) \to 1$, as its $\mathbf{1}$-part grows $\propto \ell$ while the remainder grows only $\propto \sqrt{\ell}$. Figure 13 shows the cross-layer cosine near c directly.

Pairwise alignment across all layer pairs (Figure 13). Section 5 reports the alignment of consecutive branch means (Figure 4, first panel, and Figure 12a). Figure 13 plots the full matrix $\cos(b_\ell, b_{\ell'})$ for every eighth layer from layer 8, which is nearly flat: for a single uncentered matrix the pairs correlate near c (off-diagonal mean 0.320 for ReLU against $1/\pi = 0.318$, standard deviation 0.028), and for two matrices or tanh they sit near zero (off-diagonal mean 0.002 and -0.000 respectively). The correlation does not fall off with the layer separation $|\ell - \ell'|$ because it is carried by one shared direction common to every layer rather than by a chain of layer-to-layer dependences: the all-ones component sits in each branch mean regardless of that layer's weights, so the alignment is the same at every separation. Once the spike dominates, a single constant c therefore describes the correlation between any two layers.

Mean-growth exponents and the probe-size control (Figure 12b). The colored dashed guides are power laws $\|\mu_\ell\| \propto \ell^q$ anchored at the ℓ=1 value of the two ReLU curves, with q=1 for one matrix and q=$\frac{1}{2}$ for two. A finite probe of n inputs estimates a zero mean only up to sampling noise: for a zero-mean process the root-mean-square measured value is $\sqrt{\mathbb{E}\|h_\ell\|^2/n}$ (the gray noise floor in the figure), which random-walks across layers at any n and shrinks only as $1/\sqrt{n}$; we therefore vary n to separate real means from noise. Fitted over $\ell \in [50, 512]$, M=1 ReLU (q=0.99, $\propto \ell$) and M=2 ReLU (q=0.56, $\propto \sqrt{\ell}$) are n-independent, hence real. A 4× SwiGLU branch likewise has a small but n-independent mean (q≈0.5): the gated branch *decorrelates* the mean into a random walk rather than removing it. Its representation keeps 0.908 of the width at $L = 512$ ($n = 512$), with a

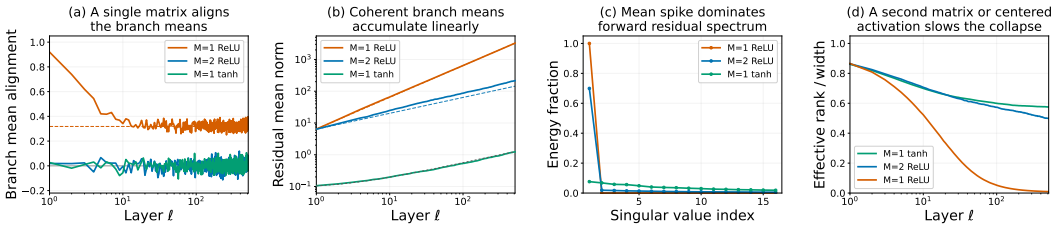

Figure 12: **A single uncentered branch matrix grows a mean spike that collapses the forward residual stream rank; a second matrix or a centered activation slows the collapse.** We track the residual stream representation Z_ℓ, the matrix whose rows are the representations of n probe inputs at layer ℓ, in a depth-L network at initialization (Pre-RMSNorm, β=1, d=256, L=512, 3 seeds). (a) The mean of Z_ℓ over inputs is the sum $b_1 + \cdots + b_\ell$ of the per-layer branch means; we plot the alignment $\cos(b_\ell, b_{\ell+1})$ of consecutive terms. With a single uncentered matrix their alignment approaches the moment ratio c ($1/\pi$ for ReLU, dashed) after the first layers; a second matrix decorrelates them; tanh's branch means vanish by odd symmetry, so there is nothing to align. (b) The norm of that sum grows linearly when the terms align ($\propto \ell$ guide) and as a random walk when they do not ($\propto \sqrt{\ell}$). tanh's true mean is zero by odd symmetry; its curve sits on the gray n-sample noise floor. (c) The spectrum of Z_L at the final layer (top 16 of d=256 singular values): the top singular value carries the mean, with energy fraction 1.00 for M=1 ReLU (a rank-1 spike) and 0.70 for M=2 ReLU (a full tail remains); tanh has no spike, and its top direction holds only 0.08, barely above the rest of its spectrum. (d) The effective rank of Z_ℓ collapses for M=1 ReLU and falls more slowly for M=2 ReLU and tanh, to 0.50 and 0.58 of the width at $L = 512$.

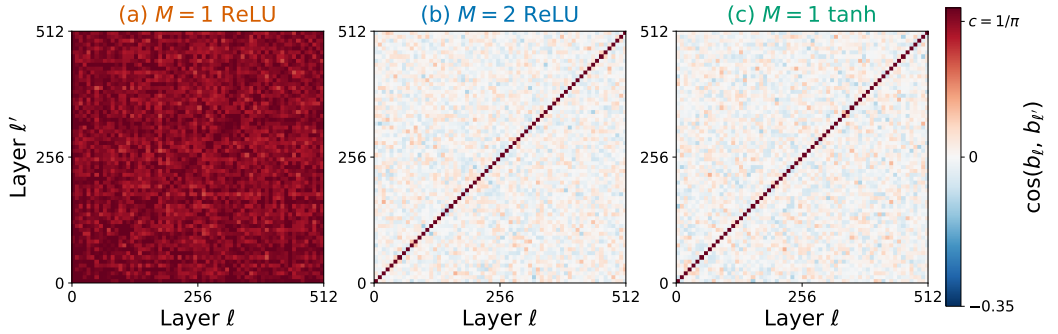

Figure 13: **Beyond the first layers, the branch-mean correlation is about the same for every pair of layers, not only neighbors.** Pairwise cosine $\cos(b_\ell, b_{\ell'})$ of the branch means at initialization (Pre-RMSNorm, β=1, d=256, L=512, n=4096, 3 seeds; the matrix samples the depth-512 network every 8 layers from layer 8). **(a)** A single uncentered matrix aligns the pairs near the moment ratio $c = 1/\pi$ (off-diagonal mean 0.320, standard deviation 0.028); the field is nearly uniform, so the alignment does not decay with layer separation. **(b)** A second matrix decorrelates the means, leaving the off-diagonal pairs near zero (mean 0.002). **(c)** tanh, which is odd, has no mean to align (mean -0.000). The diagonal is trivially 1.

hidden width of $4d$, unlike the $d \times d$ branches in Figure 12. tanh's apparent $\sqrt{\ell}$ instead scales as $1/\sqrt{n}$ and vanishes as $n \to \infty$, confirming that the tanh branch mean is zero, as the odd symmetry of the bias-free tanh network requires; in the figure its curve coincides with the floor. Probe sizes in Figure 12: the mean-magnitude panel uses n=32768 so the small real means sit well above the floor; the alignment panel uses n=4096; the rank and spectrum panels use n=512. Taking $n > d$ removes the hard rank ceiling of n, but the measured effective rank still depends on n, so we state n with each rank measurement.

Width independence (Table 6). We confirm the collapse is not specific to narrow networks. Table 6 reports the effective rank $\mathrm{erank}(Z_\ell)/d$ for M=1 ReLU at four widths from 128 to 1024: the trajectory

Table 6: **The rank collapse is not a narrow-network artifact.** Effective rank as a fraction of width, $\mathrm{erank}(Z_\ell)/d$, for M=1 ReLU at initialization (Pre-RMSNorm, β=1, mean over 3 seeds, $n = \max(1024, 2d)$ probe inputs). The collapse occurs at every width.

	Layer ℓ				
Width d	32	64	128	256	512
128	0.216	0.097	0.044	0.023	0.015
256	0.229	0.100	0.040	0.018	0.010
512	0.250	0.109	0.040	0.015	0.007
1024	0.296	0.136	0.047	0.015	0.006

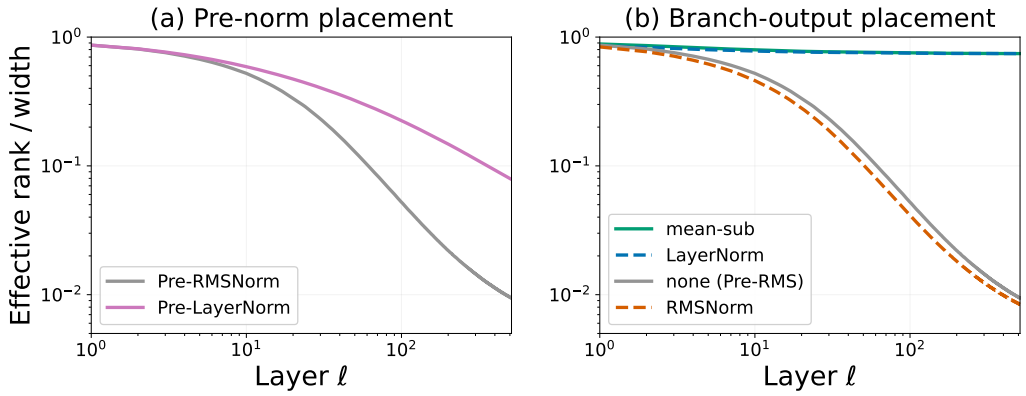

Figure 14: **Pre-Norm cannot remove the mean spike; a branch-output normalization can, but only by subtracting the mean.** Effective rank $\mathrm{erank}(Z_\ell)/d$ for M=1 ReLU at initialization (β=1, d=256, L=512, n=512, 3 seeds). **(a)** In the Pre-Norm placement both normalizers collapse: Pre-LayerNorm, which centers the branch input, reaches 0.078 by depth 512 against 0.009 for Pre-RMSNorm — about eight times higher, but still collapsed. **(b)** At the branch output (Pre-RMSNorm fixed), mean subtraction and LayerNorm keep most of the rank (0.74), while RMSNorm and no branch normalization collapse (0.008 and 0.009): the centering matters, the rescaling does not.

changes little with width, falling to a few percent of the width by depth 128 at every width, and the mean holds 1.00 of the energy of Z_ℓ ($\|\mu_\ell\|^2$ over the mean squared row norm) by the same depth.

Normalization placement (Figure 14). To understand how normalization placement interacts with this mean spike, we measure the effective rank of the representation under a Pre-Norm and under a branch-output normalization, and find that only a branch-output normalization that subtracts the mean removes the spike (Figure 14). In the Pre-Norm placement the rank collapses regardless of the normalizer (panel a): centering the branch input with LayerNorm slows the collapse relative to RMSNorm, because it breaks the feedback loop by which the stream's accumulated mean re-enters the branch, but the rank still collapses because the activation re-creates the mean at its output. At the branch output, after the activation, the rank stays high only if the normalization subtracts the mean (panel b): mean subtraction and LayerNorm hold it high, while RMSNorm, which only rescales, collapses like no normalization.

Peri-LN (Kim et al., 2025), a placement already used in several open-source models, adds a normalization at the branch output in addition to the Pre-Norm. Their MLPs use SwiGLU, with a down-projection after the activation, so they do not produce this single-matrix spike.

Training scope: the magnitude and rank channels (Figure 15). The mean spike could impair training in two ways: through the magnitude it adds to the residual stream, or through the rank it removes from the representation. We train single-matrix (M=1) and two-matrix (M=2) ReLU networks on CIFAR-10 with Adam for 5000 steps (width 256, batch 256, β=1), sweeping the learning

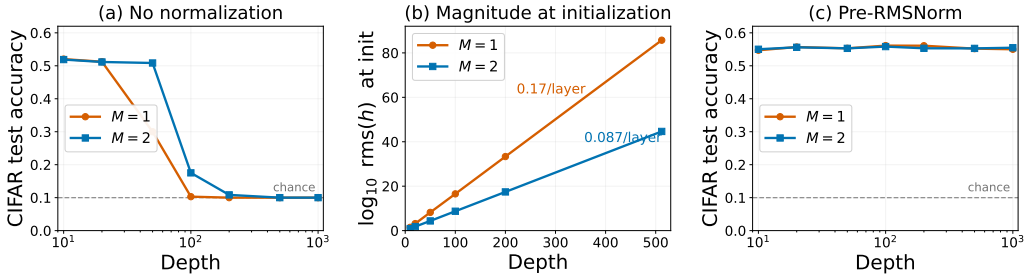

Figure 15: **Without normalization, the single-matrix network stops training at about half the depth of the two-matrix network; with Pre-RMSNorm, both reach similar CIFAR-10 accuracy despite the collapsed single-matrix representation.** All ReLU, β=1, width 256. **(a)** Without normalization, the best test accuracy over the learning-rate sweep drops to 0.30 at depth 50 and to chance by depth 100 for the single-matrix architecture (M=1), and to 0.18 at depth 100 and near chance by depth 200 for the two-matrix architecture (M=2); CIFAR chance accuracy is 0.1. **(b)** The residual magnitude at initialization grows exponentially with depth, twice as fast for the single-matrix architecture (0.17 versus 0.087 decades per layer), so it reaches a given magnitude at half the depth. **(c)** With Pre-RMSNorm the magnitudes are controlled and the two architectures reach similar best accuracy over the learning rate sweep (within a point at every depth up to 1000), even though the single-matrix representation has collapsed to an effective rank of about 2 at initialization (Figure 12d).

rate over six values from 3×10^{-5} to 10^{-2} (the two largest omitted at depths 500 and 1000, where the optimal rate is smaller) and reporting the best test accuracy over the sweep. Without normalization, the single-matrix network falls from 0.51 at depth 20 to 0.30 at depth 50 and to chance at depth 100, and the two-matrix network falls to 0.18 at depth 100 and to near chance at depth 200. The two architectures differ in their number of weight matrices as well as in the spike, so these comparisons are correlational. With Pre-RMSNorm, both reach similar best accuracy, within a point at every depth up to 1000 (Figure 15c). Across the individual learning rates, the largest systematic difference is at the smallest rate tested (3×10^{-5}, well below the optimum of 10^{-3}), where the single-matrix architecture trains 4–5 points higher at depths $\geq$100; this is consistent with an artifact of the fixed step budget, since the single-matrix network has half as many weight matrices to train at a learning rate this small. The initialization magnitude in panel (b) is the accumulated $\log_{10}$ of the residual RMS; the per-layer renormalization that keeps it finite at large depth is exact for ReLU, so the two slopes are measured rather than fit.

D Width Expansion: Full Marchenko–Pastur Derivation

This appendix derives the results used in Section 5. Sections D.1–D.3 treat ReLU and absolute value: we factorize the branch Jacobian, reduce the Gram matrix of the surviving submatrix to a Wishart matrix, and read off the large-width rank threshold $m/d = 1/p(\sigma)$ and the conditioning of the submatrix. The later subsections cover smooth activations, the branch Jacobian's own conditioning, and a Gaussian-product reference.

D.1 Jacobian factorization and the survival rate

The feedforward branch $f(x) = W_{\text{down}}\,\sigma(W_{\text{up}}x)$, with $W_{\text{up}} \in \mathbb{R}^{m\times d}$ and $W_{\text{down}} \in \mathbb{R}^{d\times m}$, has Jacobian

$$J_f(x) = W_{\text{down}}\,D(x)\,W_{\text{up}}, \qquad D(x) = \operatorname{diag}\big(\sigma'(W_{\text{up}}x)\big). \tag{22}$$

The mask $D(x)$ is diagonal, so it can only lower rank: $\operatorname{rank} J_f \leq \#\{i : \sigma'((W_{\text{up}}x)_i) \neq 0\}$. We call the expected fraction of units with nonzero derivative the survival rate, $p(\sigma) = \Pr_{z\sim\mathcal{N}(0,1)}[\sigma'(z) \neq 0]$. ReLU has $\sigma' = \mathbf{1}[z > 0]$ and $p = \frac{1}{2}$; absolute value has $\sigma' = \operatorname{sign}(z)$ and $p = 1$. At initialization the pre-activations are i.i.d. and mean-zero, so D keeps pm rows of W_{up} in expectation.

D.2 THE SURVIVING SUBMATRIX

For ReLU and absolute value the derivative takes only the values 0 and ± 1, so D either drops a row of W_{up} or keeps it at unit magnitude; it never rescales the surviving rows, which is why the analysis below is exact for these two activations.

For absolute value, $D = \operatorname{diag}(\operatorname{sign}(W_{\text{up}}x))$ flips the signs of some rows, and the signs cancel in the Gram, $(D\,W_{\text{up}})^\top (D\,W_{\text{up}}) = W_{\text{up}}^\top W_{\text{up}}$. The surviving submatrix has exactly the spectrum of the full Gaussian W_{up}, with aspect ratio $\gamma = m/d$.

For ReLU the surviving rows $\{w_i : w_i^\top x > 0\}$ are Gaussian rows restricted to a half-space. The conditioning gives each a mean $\sqrt{2/\pi}\,\hat{x}$ along $\hat{x} = x/\|x\|$ and shrinks its variance there from 1 to $1 - \frac{2}{\pi}$, while the other $d-1$ directions stay standard Gaussian. Their second moment $\mathbb{E}[w_i w_i^\top \mid w_i^\top x > 0]$ is nevertheless the identity, because the reduced variance and the squared mean cancel along $\hat{x}$:

$$\mathbb{E}\big[w_i w_i^\top \mid w_i^\top x > 0\big] = \underbrace{\big(I - \tfrac{2}{\pi}\hat{x}\hat{x}^\top\big)}_{\text{covariance}} + \underbrace{\tfrac{2}{\pi}\hat{x}\hat{x}^\top}_{(\text{mean})(\text{mean})^\top} = I. \tag{23}$$

The surviving rows are not centered Gaussian, so the Marchenko–Pastur law needs a further argument, which sign symmetry provides. A row w and its negative have the same outer product, so $ww^\top$ does not depend on the sign of $w^\top x$. Conditional on the number K of surviving rows, the Gram matrix $(D\,W_{\text{up}})^\top (D\,W_{\text{up}})$ therefore has exactly the Wishart distribution of K unconditioned Gaussian rows. For ReLU $K \sim \text{Binomial}(m, \frac{1}{2})$, and the aspect ratio $\gamma = K/d$ approaches $pm/d = \frac{1}{2}\,(m/d)$ at large width.

D.3 MARCHENKO–PASTUR LAW AND THE FULL-RANK THRESHOLD

In the proportional limit $d, pm \to \infty$ with $\gamma = pm/d$ fixed, the eigenvalues of $\frac{1}{pm}(D\,W_{\text{up}})^\top (D\,W_{\text{up}})$ converge to the Marchenko–Pastur law (Marchenko & Pastur, 1967) on $[\lambda_-, \lambda_+]$ with $\lambda_\pm = (1 \pm 1/\sqrt{\gamma})^2$. The lower edge λ_- is zero at $\gamma = 1$ and positive for $\gamma > 1$. At finite width the submatrix is, almost surely, full rank exactly when $K \geq d$. For absolute value $K = m$, so this holds for every $m \geq d$. For ReLU, as d grows with m/d fixed, the probability that $K \geq d$ tends to one above $m/d = 2$ and to zero below it, so the threshold $m/d = 1/p(\sigma)$ is a large-width transition. At $m = 2d$ the ReLU submatrix is full rank with probability tending to one half (0.51 at $d = 512$), although the fraction of missing directions tends to zero. Near the threshold the smallest eigenvalue approaches zero, so the submatrix is badly conditioned; for $\gamma > 1$ the spectrum separates from zero and the condition number $\sigma_{\max}/\sigma_{\min} = (1 + 1/\sqrt{\gamma})/(1 - 1/\sqrt{\gamma})$ is finite, falling from ∞ at $\gamma = 1$ to 5.8 at $\gamma = 2$ and 3.0 at $\gamma = 4$ (the ReLU row of Figure 16, at $m/d = 2, 4, 8$).

The branch Jacobian inherits this rank. Because $W_{\text{down}} \in \mathbb{R}^{d \times m}$ with $m \geq d$ is full rank and independent of $D\,W_{\text{up}}$ at initialization, the column space of $D\,W_{\text{up}}$ (of dimension $\operatorname{rank}(D\,W_{\text{up}}) \leq d$) meets the $(m-d)$-dimensional null space of W_{down} only at the origin almost surely, so $\operatorname{rank} J_f = \operatorname{rank}(D\,W_{\text{up}})$. The width threshold for the branch Jacobian is the same as for the surviving submatrix.

D.4 SMOOTH ACTIVATIONS

For GELU and SiLU the derivative is nonzero almost surely under a Gaussian pre-activation, so $D = \operatorname{diag}(\sigma'(W_{\text{up}}x))$ rescales every row without dropping any, and J_f has rank d almost surely for every $m \geq d$. This exact rank says nothing about conditioning or effective rank, since small derivatives can strongly attenuate some directions. The Gram becomes $\frac{1}{n_{\text{eff}}} W_{\text{up}}^\top \Lambda\, W_{\text{up}}$ with $\Lambda = \operatorname{diag}(\sigma'(W_{\text{up}}x)^2)$ and $n_{\text{eff}} = \sum_i \sigma'((W_{\text{up}}x)_i)^2$: a Gaussian matrix with a per-row variance profile, not the masked submatrix in Section D.2.

For ReLU and absolute value n_{eff} is the number of surviving rows. For GELU and SiLU it is the total squared derivative, with $n_{\text{eff}}/m \to \mathbb{E}[\sigma'^2]$, which is 0.46 for GELU and 0.38 for SiLU. These moments measure derivative magnitude and define no full-rank threshold.

The spectrum is then no longer Marchenko–Pastur: it depends on the whole distribution of the diagonal weights σ'^2 and on their dependence on the Gaussian rows, not only on $\mathbb{E}[\sigma'^2]$. For diagonal

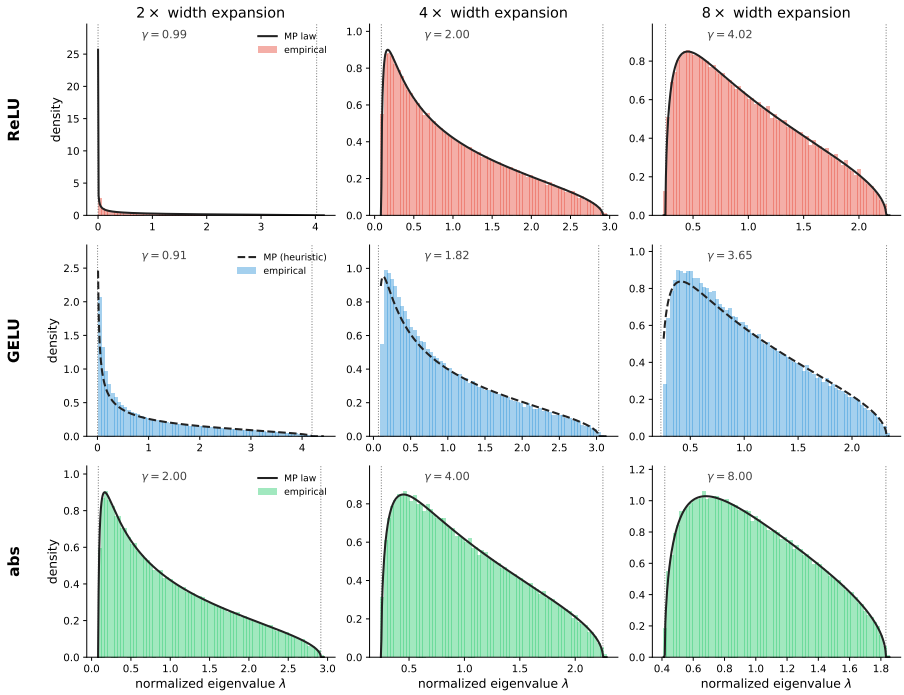

Figure 16: **The masked up-projection follows the Marchenko–Pastur law for ReLU and abs, and its conditioning improves as m/d rises above $1/p$.** Eigenvalue histogram (filled) of the Gram matrix $G = \frac{1}{n_{\text{eff}}}(D\,W_{\text{up}})^\top (D\,W_{\text{up}})$ of the masked up-projection $D\,W_{\text{up}}$, with $D = \operatorname{diag}(\sigma'(W_{\text{up}}x))$ and $n_{\text{eff}} = \sum_i \sigma'((W_{\text{up}}x)_i)^2$, for ReLU, GELU, and abs (rows) at width expansions $m/d \in \{2, 4, 8\}$ (columns), against the Marchenko–Pastur density with aspect ratio $\gamma = n_{\text{eff}}/d$ (line) and its support edges $(1 \pm 1/\sqrt{\gamma})^2$ (dotted). For ReLU, D keeps or drops rows; for abs, it flips signs, which cancel in G. In both cases n_{eff} counts the surviving rows. At $\gamma = 1$ ($m/d = 2$ for ReLU) the lower edge of the spectrum is zero and $D\,W_{\text{up}}$ is badly conditioned; for $\gamma > 1$ the spectrum separates from zero. Since W_{down} is full rank, this submatrix sets $\operatorname{rank} J_f$. GELU reweights every row without dropping any, so its $D\,W_{\text{up}}$ is full rank for $m \geq d$, and its line (dashed) is a heuristic reference. Random Gaussian W_{up} with unit-variance entries, $d = 512$, 40 seeds.

weights independent of the rows, this is the generalized Marchenko–Pastur law of a diagonally weighted Gram matrix (Marchenko & Pastur, 1967). The standard Marchenko–Pastur density at $\gamma = \mathbb{E}[\sigma'^2]\,(m/d)$, which we overlay for GELU, matches the scale of the measured spectrum but not its shape, and we show it only as a heuristic reference.

D.5 Conditioning of the branch Jacobian itself

Well above the threshold the submatrix is well-conditioned, but the branch Jacobian $J_f = W_{\text{down}}(D\,W_{\text{up}})$ is not. Restricted to the d-dimensional column space of $D\,W_{\text{up}}$, W_{down} acts as a square $d \times d$ Gaussian, which is poorly conditioned; widening the branch enlarges m but leaves this $d \times d$ map unchanged.

The submatrix conditioning improves with width (Figure 16) but the conditioning of the Jacobian J_f does not: for ReLU at $d = 512$, as γ grows from 2 to 16 the smallest eigenvalue of the normalized Gram G of Figure 16 rises from 0.09 to 0.57, while the smallest eigenvalue of $J_f^\top J_f$ (weights at the default $1/\text{fan-in}$ variance) stays near 10^{-6}, although J_f is full rank. The effective rank of J_f, however, does improve with width, and at initialization the residual skip connection keeps the propagated gradient full rank almost surely.

D.6 THE GAUSSIAN-PRODUCT REFERENCE

As a reference, we compute the effective rank per dimension, $\mathrm{erank}(J_f)/d$, of a product of two Gaussian matrices (W_{up} and W_{down} both Gaussian, with no activation, so $\gamma = m/d$): at each γ we draw the two factors, form the product, and average $\mathrm{erank}(J_f)/d$ over 20 seeds ($d = 256$). It rises with γ and reaches 0.79 at $\gamma = 10$.

It sits below the surviving submatrix's effective rank at every γ because the product spreads the spectrum (the same structure behind J_f's poor conditioning in Section D.5), so the branch Jacobian stays below the effective rank of the surviving submatrix, even where it is full rank, and approaches 0.805 of the width, the effective rank of the square Gaussian map that W_{down} applies to the column space of $D\,W_{\mathrm{up}}$.

E RESIDUAL MLP GRID

This appendix describes the residual MLPs in Figures 2 (right), 3 (center) and 5 (gray), and how we compute their mean path length. Each network has width $d = 256$, depth $L \in \{12, 24, 48, 96, 192, 384\}$, RMSNorm without a gain ($\epsilon = 10^{-6}$) in the Pre-Norm, Output-Norm or Post-Norm placement, and a branch without biases whose weights are drawn from $\mathcal{N}(0, \alpha^2/\text{fan-in})$. The branch with $4\times$ expansion and ReLU covers the full grid of three placements, $\alpha \in \{0.5, 1, 2\}$, $\beta \in \{1, 1/\sqrt{L}\}$ and six depths (108 configurations), and at $\alpha = \beta = 1$, a branch with two $d \times d$ ReLU matrices and a $4\times$ GELU branch (tanh approximation) add 24 configurations in Pre-Norm and Post-Norm, for 132 in total. Since α cancels in Output-Norm (Section 4), the 24 Output-Norm configurations at $\alpha = 0.5$ and 2 reuse the random draws at $\alpha = 1$ and match them to a relative difference below 10^{-4}, so the grid holds 108 distinct networks. We average over three seeds, each drawing the weights and one input $h_0 \sim \mathcal{N}(0, I_d)$ rescaled to unit root-mean-square, and compute in float64 after the draws, rescaling the input–output Jacobian to unit Frobenius norm after each block.

Exact path masses. Each block Jacobian is the sum of a skip term and a branch term, $\partial h_\ell/\partial h_{\ell-1} = A_\ell + B_\ell$. In Pre-Norm and Output-Norm, $A_\ell = I$ and $B_\ell = \beta J_\ell$, with the normalization included in J_ℓ. In Post-Norm, the block Jacobian is $J_{\mathrm{Norm}}(I + \beta J_\ell)$ with J_{Norm} evaluated at the residual sum, so $A_\ell = J_{\mathrm{Norm}}$ and $B_\ell = \beta J_{\mathrm{Norm}} J_\ell$. Every path through the first ℓ blocks takes either the skip or the branch at block ℓ, so the sum $P_k^{(\ell)}$ of the paths that take exactly k branches satisfies

$$P_k^{(\ell)} = A_\ell\, P_k^{(\ell-1)} + B_\ell\, P_{k-1}^{(\ell-1)}, \qquad P_0^{(0)} = I, \quad P_k^{(0)} = 0 \;\; (k \geq 1), \quad P_{-1}^{(\ell)} = 0. \tag{24}$$

The masses are $m_k = \|P_k^{(L)}\|_F$, which give $\langle k \rangle$ in Eq. 5; with identity skips, $P_k^{(L)} = \beta^k S_k$ in Eq. 4. The recursion takes $O(L^2)$ matrix products, and we store each $P_k^{(\ell)}$ as a unit-Frobenius matrix and a log-norm because the masses span many orders of magnitude. In every run, $\sum_k P_k^{(L)}$ reproduces the directly composed Jacobian to a relative Frobenius error below 4×10^{-14}, and in three-block networks of every placement and branch, the analytic block Jacobians match automatic differentiation to below 10^{-14}.

Branch-to-skip ratio. The ratio of block ℓ is $r_\ell = \|B_\ell\|_F / \|A_\ell\|_F$, which reduces to Eq. 3 when $A_\ell = I$. At initialization and unit stream scale, averaging over the weights gives $\mathbb{E}\|W_{\mathrm{down}} D\, W_{\mathrm{up}}\|_F^2 \approx d\, p\, \alpha^4$ (exactly for ReLU) with $D = \mathrm{diag}(\sigma'(W_{\mathrm{up}} x))$, so, in Pre-Norm and Post-Norm, $r_\ell^2 \approx p\, \beta^2 \alpha^{2M}$ with $M = 2$, and $M = 1$ for one matrix; in Pre-Norm with $\beta = 1$, the first blocks have $r_1^2 = 0.506$, 8.10 and 0.0316 at $\alpha = 1$, 2 and 0.5. If each A_ℓ acts as the identity (after rescaling the block so that $\|A_\ell\|_F = \sqrt{d}$) and the B_ℓ are independent across blocks, have zero mean and scale squared Frobenius norms by r_ℓ^2 on average, then the cross terms between paths vanish in expectation and $\mathbb{E}\, m_k^2 = d\, e_k(r_1^2, \ldots, r_L^2)$, the elementary symmetric polynomial of degree k. The logarithmic derivative of $\prod_\ell (1 + r_\ell^2 x) = \sum_k e_k x^k$ at $x = 1$ then gives $\langle k \rangle \approx \sum_\ell r_\ell^2/(1 + r_\ell^2)$. Evaluated with the measured ratio of every block, this approximation differs from the exact $\langle k \rangle$ by 0.3% at the median over the 396 runs and by 1.5% in the worst run.

Mean path length across depth. At $\alpha = \beta = 1$ with the $4\times$ ReLU branch, least-squares fits over the six depths give $\langle k \rangle = 0.334L - 0.02$ for Post-Norm, $0.94 \ln L - 0.57$ for Pre-Norm and

$0.97 \ln L - 0.20$ for Output-Norm, each with $R^2 \geq 0.9998$. With $\beta = 1/\sqrt{L}$ and $\alpha = 1$, $\langle k \rangle$ stays between 0.398 and 0.404 at every depth in Pre-Norm (Figure 3, center), and between 0.48 and 0.50 in Post-Norm.

One curve. The configurations run from $\langle k \rangle = 0.030$, with an effective rank of 0.992 of the width, to $\langle k \rangle = 341$, with 0.0094. A cubic regression spline with five interior knots, fit to $\ln(\mathrm{erank}/d)$ against $\ln\langle k \rangle$ over the 132 configurations, has $R^2 = 0.998$ and a root-mean-square residual of 0.047, about 5% of the effective rank; the largest residual, -0.24, is Post-Norm with two $d \times d$ matrices at $L = 192$. The same fit gives $R^2 = 0.94$ against $\sum_\ell r_\ell^2$ and $R^2 = 0.49$ against $\beta\alpha^M\sqrt{L}$.

F TRANSFORMER EXPERIMENTS

This appendix describes the transformers in Section 6 and the third panel of Figure 4, and reports the measurements that Section 6 summarizes.

Architecture and training. The models are decoder-only transformers of width $D = 256$ with N blocks, each a causal self-attention sublayer with four heads of dimension 64 followed by a feedforward sublayer $W_2\,\mathrm{GELU}(W_1 x)$ with $4\times$ expansion. Attention applies LayerNorm to the queries and keys of each head (Henry et al., 2020; Dehghani et al., 2023) and then rotary position embeddings (Su et al., 2024). The untied embedding and readout cover the GPT-2 vocabulary (Radford et al., 2019), padded to 50,304 tokens, the context has 2048 tokens, and no layer has a bias. Pre-LN and Post-LN are the Pre-Norm and Post-Norm placements in Section 4 with LayerNorm (gain, no bias): a sublayer f updates the stream as $h \leftarrow h + \beta f(\mathrm{LN}(h))$ in Pre-LN, which adds a final LayerNorm before the readout, and as $h \leftarrow \mathrm{LN}(h + \beta f(h))$ in Post-LN. The branch scale β multiplies every branch throughout training and equals 1 outside the branch-scale sweep. The plain initialization draws the embedding from $\mathcal{N}(0, 1/D)$ and every other weight matrix from $\mathcal{N}(0, 1/\text{fan-in})$; the scaled initialization of GPT-2 (Radford et al., 2019) also multiplies the output projection of every branch by $(2N)^{-1/2}$. We train on the 350B-token sample of FineWeb (Penedo et al., 2024) with the GPT-2 tokenizer, reading 2048-token windows without repetition, in a seeded order, from a 1.46B-token shard. We report the loss on 983,040 tokens from a fixed set of 1,470 FineWeb documents that also occur in the training shard, so it is not a held-out estimate. Runs of the same depth read the same training windows, so the overlap is the same across placements, initializations, branch scales and learning rates at each depth. Each run trains to 20 tokens per parameter (Hoffmann et al., 2022), counting $12D^2N + DV$ parameters, with batches of 64 sequences, which gives 2,685, 3,405, 4,845 and 7,725 steps for $N = 6$, 12, 24 and 48. The optimizer is AdamW (Loshchilov & Hutter, 2017) with decay rates 0.9 and 0.95, $\epsilon = 10^{-8}$ and weight decay 0, after clipping the gradient to global norm 1; the learning rate rises linearly from $0.01\times$ its peak over 2% of the steps (at least 100) and decays with a cosine to $0.1\times$ the peak. Every run uses seed 0.

Sweeps. The depth grid crosses $N \in \{6, 12, 24, 48\}$, both placements, both initializations and six peak learning rates, $\{1, 3\} \times \{10^{-4}, 10^{-3}, 10^{-2}\}$, for 96 runs at $\beta = 1$. The branch-scale sweep uses the plain initialization with $\beta \in \{1/16, 1/8, 1/4, 1/2, 1, 2\}$, $N \in \{12, 48\}$, both placements and the five learning rates up to 10^{-2}, for 120 runs. A run trains if its final evaluation loss, the mean cross-entropy over these 983,040 tokens after the last step, is below 7 nats. No run diverges; every run that trains ends at or below 5.96 nats, and every run that fails ends between 7.5 and 7.7 nats, so the overlap with the training shard does not change which runs train.

Same-token Jacobian. We measure the Jacobian from the embedding h_0^t to the stream h_{2N}^t after the last of the $2N$ sublayers (before the final LayerNorm of Pre-LN), where h_ℓ^s is the stream at position s after sublayer ℓ (so h_{2N}^t is the stream h_N^t after block N in Section 6). With a causal mask, h_ℓ^s depends on $h_{\ell-1}^{s'}$ only for $s' \leq s$, so the Jacobian of each sublayer over all positions is block lower triangular, and the diagonal blocks of a product of such matrices are the products of their diagonal blocks. This gives exactly

$$\frac{\partial h_{2N}^t}{\partial h_0^t} = (A_{2N} + B_{2N}) \cdots (A_1 + B_1), \qquad A_\ell + B_\ell = \frac{\partial h_\ell^t}{\partial h_{\ell-1}^t}, \tag{25}$$

where factor ℓ is the Jacobian of sublayer ℓ with respect to its input at position t, with the other positions held at their forward values. In Pre-LN, $A_\ell = I$ and B_ℓ is the Jacobian of $\beta f(\mathrm{LN}(h))$; in

Table 7: **Learning-rate outcomes of the depth grid.** ✓ marks a run that trains (final evaluation loss below 7 nats) and × a run that fails. Columns give the peak learning rate in units of 10^{-4}; $\langle k\rangle_0$ is the mean path length of the same-token Jacobian at initialization. One seed per run.

		Plain initialization							Scaled initialization						
	N	$\langle k\rangle_0$	1	3	10	30	100	300	$\langle k\rangle_0$	1	3	10	30	100	300
Pre-LN	6	2.69	✓	✓	✓	✓	✓	✓	2.14	✓	✓	✓	✓	✓	✓
	12	3.00	✓	✓	✓	✓	✓	✓	2.32	✓	✓	✓	✓	✓	✓
	24	3.29	✓	✓	✓	✓	✓	✓	2.46	✓	✓	✓	✓	✓	✓
	48	3.54	✓	✓	✓	✓	✓	✓	2.57	✓	✓	✓	✓	✓	✓
Post-LN	6	1.91	✓	✓	✓	✓	×	×	0.22	✓	✓	✓	✓	×	×
	12	3.82	✓	✓	×	×	×	×	0.23	✓	✓	✓	✓	×	×
	24	7.55	×	×	×	×	×	×	0.23	✓	✓	✓	×	×	×
	48	15.02	×	×	×	×	×	×	0.23	✓	✓	✓	×	×	×

Post-LN, A_ℓ is the Jacobian of the LayerNorm at the residual sum and B_ℓ is A_ℓ times the Jacobian of βf, and Eq. 24 over the $2N$ sublayers gives m_k and $\langle k\rangle$. We compute the factors by automatic differentiation in float32 at the last position of 256-token prefixes of two evaluation sequences outside the loss evaluation, multiply them in float64, and average $\mathrm{erank}\,/D$ and $\langle k\rangle$ over the two sequences, at initialization and after training. At $D = 64$, $N = 3$ and 16 tokens, the product matches automatic differentiation of the whole network to a relative Frobenius error below 10^{-6} for both placements and initializations.

Learning-rate outcomes. Table 7 marks which runs of the depth grid train. In the branch-scale sweep, Post-LN at $N = 48$ with $\beta = 1/16$, $1/8$ and $1/4$ has $\langle k\rangle = 0.09$, 0.35 and 1.3 and trains up to 3×10^{-3}, 3×10^{-3} and 10^{-3}, whereas $\beta \geq 1/2$ ($\langle k\rangle \geq 4.9$) trains at no learning rate; at $N = 12$, where $\langle k\rangle$ runs from 0.022 to 7.9, the range shrinks in the same order, from all five learning rates at $\beta \leq 1/8$ to none at $\beta = 2$. The scaled initialization holds the Post-LN $\langle k\rangle$ at 0.22 to 0.23 at every depth, and at $N = 48$ it trains up to 10^{-3} with a best final loss of 3.84, against 3.72 for Pre-LN with the plain initialization and 3.69 with the scaled one. Pre-LN trains at every learning rate in both sweeps, with $\langle k\rangle$ between 1.5 and 3.7. Across both sweeps and both initializations, every Post-LN network with an initial $\langle k\rangle$ of at most 3.82 trains at some learning rate, and every one with $\langle k\rangle$ of at least 4.90 trains at none. The initial $\langle k\rangle$ does not set the trainable range by itself: at a matched $\langle k\rangle$ of about 0.09, Post-LN trains up to 10^{-2} at $N = 12$ ($\beta = 1/8$) but only up to 3×10^{-3} at $N = 48$ ($\beta = 1/16$), and at $N = 48$ the scaled initialization ($\langle k\rangle = 0.23$) trains only up to 10^{-3}, whereas $\beta = 1/8$ ($\langle k\rangle = 0.35$) trains up to 3×10^{-3}.

Rank relative to the MLP curve. We evaluate the MLP curve at a transformer's $\langle k\rangle$ by linear interpolation of $\ln(\mathrm{erank}\,/d)$ in $\ln\langle k\rangle$ between the 132 configurations in Appendix E, sorted by $\langle k\rangle$, and divide the transformer's $\mathrm{erank}\,/D$ by the result; below the smallest MLP value, $\langle k\rangle = 0.030$, the curve keeps its end value. At initialization, the 36 distinct networks in Figure 5 (first panel), with $\langle k\rangle$ from 0.022 to 31 and effective rank from 0.99 to 0.065 of the width, lie at 0.75 to 1.12 of the curve. After training, the depth grid has 68 runs that train and 28 that fail, all with final $\langle k\rangle$ inside the range of the MLP configurations. The runs that train end at a median of 0.88 of the curve (interquartile range 0.74 to 1.18). Of the runs that fail, 17 end below 0.3 of the curve, and the median over all 28 is 0.07. Seven failed runs barely change from initialization: Post-LN with the plain initialization at $N = 12$ and learning rate 10^{-3}, and at $N = 24$ and 48 with learning rates from 10^{-4} to 10^{-3}. Their $\langle k\rangle$ falls by 4 to 12% and their rank changes by at most 7%, which leaves them at 0.92 to 1.00 of the curve; at $N = 48$, $\langle k\rangle$ goes from 15.0 to 14.4–14.5 and the rank stays at 0.12. Training roughly doubles $\langle k\rangle$ (median final-to-initial ratio 2.0 over the 96 runs) and lengthens the paths in 67 of the 68 runs that train. The only run that trains and ends below 0.3 of the curve, at 0.06, is Post-LN with the plain initialization at $N = 12$ and 3×10^{-4}, which has the highest final loss among the runs that train (5.95). The spline in Appendix E gives the same medians to within 0.01.

Token representation. For the third panel of Figure 4, the rows of $H_\ell \in \mathbb{R}^{2048\times D}$ are the stream after block ℓ at every position of eight 256-token evaluation sequences, and we average $\mathrm{erank}(H_\ell)/D$ at initialization over three seeds of the plain initialization. The feedforward sublayer is removed

(attention only), reduced to one $D \times D$ matrix followed by the activation, or kept as the $4\times$ two-matrix block, with ReLU, GELU or tanh. After 48 blocks, attention only, one ReLU matrix, the two-matrix ReLU block and one tanh matrix keep 0.14, 0.17, 0.27 and 0.31 of the width in Pre-LN, and 0.024, 0.028, 0.013 and 0.025 in Post-LN, where all seven variants keep 0.013 to 0.028. The mean-update alignment is the cosine between the changes $\mu_\ell - \mu_{\ell-1}$ of the token mean in consecutive blocks, which include the attention sublayer, unlike the branch means in Section 5. Averaged over the last 11 of the 47 pairs in Pre-LN at $N = 48$, it is 0.45 with one ReLU matrix and 0.02 with the two-matrix ReLU block. With one ReLU matrix, the token mean holds 0.89 of the energy of H_{48} ($\|\mu_{48}\|^2$ over the mean squared row norm) in Pre-LN and 0.19 in Post-LN, whose LayerNorm leaves each token with zero mean over its coordinates; with the two-matrix ReLU block, the Post-LN token mean holds 0.95, along a direction orthogonal to the all-ones vector.

Compute. Each run uses one TPU v4-8 host, data parallel over its four chips; the median wall-clock time per run, including the measurements, is 6, 10, 22 and 60 minutes for $N = 6$, 12, 24 and 48, and the 96 depth-grid runs took 39 host-hours.

G Extended Related Work

Dynamical isometry and signal propagation. In random networks, dynamical isometry at large depth requires every factor of the Jacobian product to be close to an isometry up to a global scale, which calls for orthogonal weights (Saxe et al., 2014) and an activation whose derivative has nearly constant magnitude over the bulk of the preactivations, a condition that ReLU cannot meet (Pennington et al., 2017). Activations with a large linear region around the origin attain approximate isometry, although an overly large linear region reduces expressivity (Murray et al., 2022). In deep residual networks whose branch variance scales as $1/L$, the singular-value density of the Jacobian depends only on the summed branch variance, for any activation and for Gaussian or orthogonal weights (Tarnowski et al., 2019); this sum is $\sum_\ell r_\ell^2$, the small-ratio limit of $\langle k \rangle$ (Eq. 5). Gadhikar & Burkholz (2022) instead make each ReLU residual block an orthogonal map at initialization with looks-linear weights, in line with our observation that the block, not the branch, must be near-isometric (Section 3). When the skip is a learned mixing matrix across residual streams (Zhu et al., 2025), the mixing matrix must itself be close to an isometry: with the doubly stochastic mixers of manifold-constrained hyper-connections (mHC) (Xie et al., 2025), the Jacobian spectrum develops mass near zero as the depth grows, whereas orthogonal mixers maintain dynamical isometry (Sengupta et al., 2026). In transformers, shaping the attention or the initialization controls signal propagation at large depth, in some cases without skip connections (Noci et al., 2023; He et al., 2023; He & Hofmann, 2024; Ji et al., 2026). These approaches keep each factor or block close to an isometry. We study finite branch-to-skip ratios, at which every block spreads the singular values, and find that the effective rank still follows one function of $\langle k \rangle$ (Figure 2).

Rank collapse in Jacobians and representations. In exact rank, a residual block with a small branch loses nothing, since its Jacobian is full rank (Feng et al., 2022), whereas the effective rank we track decays through every block by an amount that depends on its branch-to-skip ratio (Section 3). The rank of the representations of a batch collapses quickly with depth in random networks without normalization, which batch normalization prevents (Daneshmand et al., 2020). The mean spike in Section 5 forms at initialization, unlike the massive activations that trained language models develop in a few fixed feature dimensions (Sun et al., 2024). Baker et al. (2024) use gradient rank collapse for a different object, the rank of a weight gradient accumulated over a batch. Fernando & Guitchounts (2026) find that during the training of OLMo, the effective rank of same-token Jacobian products falls far below its value at initialization; we study the rank at initialization, which in our transformers follows the curve of the residual MLPs (Section 6).

Rank collapse along the token dimension. A separate line of work studies collapse along the token dimension, where self-attention drives the representations of all tokens toward a common vector (Dong et al., 2021; Noci et al., 2022), which appears as over-smoothing in vision transformers and BERT (Wang et al., 2022; Shi et al., 2022). Nait Saada et al. (2025) identify a second form, rank collapse in width as the context length grows, caused by a spectral gap in the softmax attention matrix that also exacerbates the collapse in depth. With LayerNorm and suitable value matrices, the attention dynamics admit equilibria of every rank between one and full, although collapse still occurs for other

value matrices (Wu et al., 2024). With skip connections, the outcome depends on the spectrum of the value and projection weights (Dovonon et al., 2025), a strong enough skip prevents the collapse (Arangath Joseph et al., 2025), and the strength required depends on the depth (Giorlandino & Goldt, 2026). Even with skip connections, small weights leave the network well approximated by a single attention layer (Alman & Song, 2025). These works concern the token dimension rather than the feature dimension. Several of them treat a strong skip as a remedy, whereas we find that it preserves rank by shortening the gradient paths.

Residual scaling and initialization. Many analyses scale the branch as $\beta = 1/\sqrt{L}$, which holds $\langle k \rangle$ fixed across depth (Section 3), but Dey et al. (2025) (CompleteP) argue for $1/L$ in Pre-LN transformers, the unique exponent that ensures complete feature learning, and find that it also transfers hyperparameters across depth. A $1/L$ multiplier makes $\langle k \rangle$ shrink as $1/L$ ($\beta = 1/L$ gives $\langle k \rangle = 0.01$ at $L = 48$ in our networks), and whether the layers then compose depends on the feature learning that CompleteP analyzes and our analysis at initialization does not. The technical reports of recent open-weight language models describe no explicit branch multiplier, that is, $\beta = 1$ (Llama Team, 2024; DeepSeek-AI, 2024; Gemma Team, 2024; Qwen Team, 2025; OpenAI, 2025; Kimi Team, 2025). Some set the branch scale in other ways: DeepSeek-V3 initializes every weight with standard deviation 0.006, about half of $1/\sqrt{\text{fan-in}}$ at its width of 7168 (DeepSeek-AI, 2024), and Gemma 2 normalizes both the input and the output of every sublayer (Gemma Team, 2024), as in Sandwich-LN. These choices set the branch-to-skip ratio at initialization; we do not test how they affect the trained model.

Normalization placement and depth laws. The linear and logarithmic growth of $\langle k \rangle$ for Post-Norm and Pre-Norm (Section 4) has counterparts in quantities other than the output sensitivity of Liu et al. (2020). Philipp et al. (2017) show that a skip connection multiplies the excess growth rate of the gradient scale through a block by $r_\ell^2/(1 + r_\ell^2)$ to first order, with r_ℓ the ratio of the branch output to the skip output, and note that normalizing between blocks rather than inside them would keep this ratio constant with depth. With batch normalization inside the branch, the growing stream variance makes the correlation between the gradients of different inputs decay only as $1/\sqrt{L}$ rather than exponentially (Balduzzi et al., 2017). Jacobian and gradient norms change only as a power of depth under Pre-LN but exponentially under Post-LN (Doshi et al., 2023; Kedia et al., 2024), and token directions cluster exponentially fast in depth under Post-LN but only polynomially under Pre-LN (Karagodin et al., 2025). Blake et al. (2025) prove that any Pre-Norm network is equivalent to one with a unit-scale stream whose per-layer multipliers preserve the branch-to-skip ratio, which decays as the Pre-Norm stream grows. These analyses track norms, correlations and token clustering; we show that the same depth laws, through $\langle k \rangle$, also determine the effective rank of the Jacobian.

Residual networks as ensembles and composition. The ensemble view of residual networks also rests on lesion studies: deleting individual layers of a trained residual network barely changes its test error, and reordering layers degrades it only smoothly (Veit et al., 2016). Earlier work noted that small branches limit what the layers compute: a block whose branch is small relative to its skip is close to a linear function (Philipp et al., 2017), the branch scaling that keeps gradient correlations from decaying makes the contribution of each layer increasingly trivial (Balduzzi et al., 2017), and zero-initialized branches reach dynamical isometry only with very limited feature diversity (Gadhikar & Burkholz, 2022). We make this tradeoff quantitative at initialization, as one curve of effective rank against $\langle k \rangle$, but do not test whether short paths at initialization limit composition after training (Section 8). Other properties of the initialization also bear on composition: Zhang et al. (2024) find that transformers initialized at a smaller scale than usual converge to compositional rather than memorized solutions, an effect they hypothesize stems from the low-complexity bias of small initialization.