Subcritical Signal Propagation at Initialization in Normalization-Free Transformers

Sergey Alekseev
Department of Physics and Astronomy
Stony Brook University
Stony Brook, NY 11794
sergey.alekseev@stonybrook.edu

Abstract

We study signal propagation at initialization in transformers through the averaged partial Jacobian norm (APJN), a measure of gradient amplification across layers. We extend APJN analysis to transformers with bidirectional attention and permutation-symmetric input token configurations by deriving recurrence relations for activation statistics and APJNs across layers. Our theory predicts how attention modifies the asymptotic behavior of the APJN at large depth and matches APJNs measured in deep vision transformers. The criticality picture known from residual networks carries over to transformers: the pre-LayerNorm architecture exhibits power-law APJN growth, whereas transformers with LayerNorm replaced by elementwise $\tanh$-like nonlinearities have stretched-exponential APJN growth, indicating that the latter are subcritical. Applied to Dynamic Tanh (DyT) and Dynamic erf (Derf) transformers, the theory explains why these architectures can be more sensitive to initialization and optimization choices and require careful tuning for stable training.

1 Introduction

Trainability of deep neural networks is closely tied to signal propagation at initialization [Glorot and Bengio, 2010, Pascanu et al., 2013, Schoenholz et al., 2017, Pennington et al., 2017, Noci et al., 2022]. Stable training requires *criticality*, meaning that signals do not vanish or explode exponentially with depth. We study criticality through the averaged partial Jacobian norm (APJN), for which criticality means that the APJN remains finite or grows according to a power law with depth [Doshi et al., 2023].

Architecture strongly affects signal propagation and training stability [Liu et al., 2023, Bachlechner et al., 2020, Wang et al., 2022, Shleifer et al., 2021]. Pre-LayerNorm residual networks are critical for any weight initialization, whereas other setups often yield exponential gradient growth or decay [Doshi et al., 2023]. In transformers, post-LayerNorm can lead to exponentially vanishing gradients [Kedia et al., 2024] and larger final-layer gradients than pre-LN [Xiong et al., 2020], making pre-LN more robust across optimization settings.

Recent work replaces $\mathrm{LayerNorm}$ in transformers with elementwise $\tanh$-like nonlinearities, such as Dynamic Tanh (DyT) and Dynamic erf (Derf), motivated by the similarity of their input-output curves [Zhu et al., 2025, Chen et al., 2025]. Following this literature, we refer to these architectures in the title as *normalization-free* transformers. This raises a natural question: how does replacing LayerNorm with such nonlinearities affect signal propagation and training stability in deep transformers?

To address this question, we extend the APJN analysis of Doshi et al. [2023] to transformers with bidirectional attention and permutation-symmetric input token configurations, following the framework of Cowsik et al. [2024]. We show that replacing LayerNorm with elementwise $\tanh$-like nonlinearities leads to stretched-exponential APJN growth in deep layers, in contrast to the power-law

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

growth of the pre-LN baseline. This regime was previously identified in residual networks and termed *subcritical* [Yang and Schoenholz, 2017, Doshi et al., 2023]. Despite its simplifying assumptions, our theory captures both the scale and depth dependence of APJNs on real inputs in deep layers. In particular, it captures how attention modifies the scale parameter and critical exponent governing asymptotic APJN growth. Since the APJN provides a scalar measure of gradient amplification, this analysis also suggests a mechanism by which replacing LayerNorm with $\tanh$-like nonlinearities can make optimization more sensitive to architectural and initialization hyperparameters.

We further examine the optimization implications of this prediction through a series of training-stability experiments on Vision Transformers (ViTs) [Dosovitskiy et al., 2021], comparing Derf and pre-LN models, as well as Derf models initialized with different values of the scale parameter α. In the tested ViT/CIFAR-100 settings, we observe that the trainability of Derf models is more sensitive than that of pre-LN models to hyperparameters such as model depth and weight initialization, especially at larger α. We also find that Derf models with smaller α train stably over a wider range of learning rates, warmup lengths, and network depths, consistent with our theory that smaller α reduces gradient amplification.

The code to reproduce our experiments is publicly available at `https://github.com/labofdoubt/transformers-apjn`.

2 Background and Setup

2.1 Transformer

Consider a transformer [Vaswani et al., 2017] with context size n and L layers, alternating between bidirectional self-attention and a position-wise MLP with ReLU nonlinearity. For notational simplicity, we restrict attention to the single-head case.

Let $\boldsymbol{h}_s^l \in \mathbb{R}^d$ denote the activation vector at position s in layer l, with components $h_{s,i}^l$. Its evolution is given by

$$\boldsymbol{h}_s^{l+1} = \boldsymbol{h}_s^l + \begin{cases} W_O^l W_V^l \sum_{t=1}^{n} A_{st}^l \, \tilde{\boldsymbol{h}}_t^l, & l \text{ even (attn)}, \\ W_2^l \, \mathrm{ReLU}\big(W_1^l \tilde{\boldsymbol{h}}_s^l\big), & l \text{ odd (MLP)}. \end{cases} \tag{1}$$

Here, $\tilde{\boldsymbol{h}}_s^l$ is the input to the residual branch after normalization or nonlinearity, defined below. The attention weights A_{st}^l are defined in the standard way by softmax-normalized scaled dot products; see Eq. (35). We call each attention or MLP component a *layer*, and each attention-MLP pair a *block*. Thus $l = 0, \ldots, L-1$ indexes layers, $b = 0, \ldots, B-1$ indexes blocks, and $L = 2B$.

All weights are initialized from zero-mean Gaussian distributions, with variances shared across transformer blocks. In the attention layers, $W_O, W_V \in \mathbb{R}^{d\times d}$ have component-wise variances σ_O^2/d and σ_V^2/d, respectively. In the MLP layers, $W_1 \in \mathbb{R}^{4d\times d}$ and $W_2 \in \mathbb{R}^{d\times 4d}$ have component-wise variances σ_1^2/d and $\sigma_2^2/(4d)$, respectively.

2.2 LayerNorm and DyT/Derf

The inputs to the residual branches in Eq. (1), denoted by $\tilde{\boldsymbol{h}}_s^l$, are obtained from $\boldsymbol{h}_s^l$ either by applying LayerNorm [Ba et al., 2016] or by an elementwise transformation:

$$\tilde{\boldsymbol{h}}_s^l = \mathrm{LayerNorm}(\boldsymbol{h}_s^l), \qquad \text{or} \qquad \tilde{\boldsymbol{h}}_s^l = \boldsymbol{\gamma} \odot \phi_\alpha(\boldsymbol{h}_s^l) + \boldsymbol{\beta}, \tag{2}$$

where $\alpha \in \mathbb{R}$, $\boldsymbol{\beta}, \boldsymbol{\gamma} \in \mathbb{R}^d$ are learnable parameters, $\odot$ denotes elementwise multiplication, and the nonlinearity ϕ_α is applied elementwise.

We focus on $\tanh$-like nonlinearities, namely antisymmetric, monotonically increasing functions that approach 1 as their argument tends to $+\infty$. The main examples we consider are

$$\phi_\alpha(h) = \tanh(\alpha h) \qquad \text{and} \qquad \phi_\alpha(h) = \mathrm{erf}(\alpha h), \tag{3}$$

corresponding to Dynamic Tanh (DyT) and Dynamic erf (Derf), respectively [Zhu et al., 2025, Chen et al., 2025].[1] When the dependence on α is not central, we suppress it and write simply ϕ.

[1]The term "dynamic" refers to the fact that α is a learnable parameter; however, this does not play a role in the signal propagation analysis at initialization considered below.

Since our theory concerns signal propagation at initialization, we set $\boldsymbol{\gamma} = \mathbf{1}$ and $\boldsymbol{\beta} = \mathbf{0}$, which is their standard initialization in both LayerNorm and DyT/Derf, and suppress them in the notation below. The initialization of α is one of the aspects studied below. For LayerNorm, we additionally assume that the feature-wise mean is close to zero, so that LayerNorm effectively reduces to RMSNorm [Zhang and Sennrich, 2019].

2.3 Signal Propagation at Initialization

We make three simplifying assumptions to derive closed signal-propagation recurrences at initialization.

Assumption 2.1 (Signal-propagation setting).

1. *Permutation-symmetric token configurations.* Following Cowsik et al. [2024], we restrict our analysis to permutation-symmetric input token configurations, in which all token vectors have the same squared norm and pairwise dot product. This symmetry is preserved across layers in expectation, so the layerwise geometry of the activation vectors is fully characterized by two quantities: the normalized self-dot product $q^l = \mathbb{E}_\theta\big[\boldsymbol{h}_s^l \cdot \boldsymbol{h}_s^l/d\big]$, which is independent of s, and the normalized dot product between activation vectors at different positions, $p^l = \mathbb{E}_\theta\big[\boldsymbol{h}_s^l \cdot \boldsymbol{h}_t^l/d\big]\ ,\ s \neq t$, which is independent of the choice of distinct positions s, t. Here, $\mathbb{E}_\theta$ denotes expectation over random weight initializations.

2. *Gaussian mean-field approximation.* We adopt the large-width mean-field approximation [Poole et al., 2016, Schoenholz et al., 2017, Cowsik et al., 2024], under which the components of the activation vectors at layer l are modeled as jointly Gaussian. Specifically, for any feature index i and any two positions $s \neq t$,
$$(h_{s,i}^l,\, h_{t,i}^l) \sim \mathcal{N}(0,\, \Sigma^l), \qquad \Sigma^l = \begin{pmatrix} q^l & p^l \\ p^l & q^l \end{pmatrix}, \tag{4}$$
while components with different feature indices are uncorrelated.

3. *Uniform attention at initialization.* At initialization, we use the uniform-attention approximation $A_{st}^l \approx 1/n$, following Noci et al. [2022]. Under this approximation, attention acts as uniform mixing of the activation vectors.

We emphasize that our theoretical analysis does not directly apply to transformers with causal attention. In particular, the causal mask introduces an asymmetry between token positions that is not captured by the permutation-symmetric setting and uniform-attention approximation considered above. Extending the framework to account for this positional asymmetry, and establishing analogous theoretical conclusions for autoregressive language models, are beyond the scope of this work.

2.4 Averaged Partial Jacobian Norm

The Jacobian of the mapping from layer l to layer l', with $l' > l$, is defined as
$$J_{si,tj}^{l',l} = \frac{\partial h_{s,i}^{l'}}{\partial h_{t,j}^{l}}. \tag{5}$$
It characterizes both forward perturbation propagation and backward gradient propagation between layers l and l' [Schoenholz et al., 2017]. Its squared Frobenius norm, averaged over weight initializations, defines the *averaged partial Jacobian norm (APJN)*, a scalar measure of gradient amplification or attenuation under backpropagation from layer l' to layer l [Doshi et al., 2023]:
$$\mathcal{J}^{l',l} = \frac{1}{nd}\,\mathbb{E}_\theta \|J^{l',l}\|_F^2 = \frac{1}{nd}\mathbb{E}_\theta\left[\sum_{st,ij}\left(\frac{\partial h_{s,i}^{l'}}{\partial h_{t,j}^{l}}\right)^2\right]. \tag{6}$$

Notation and terminology. We refer to $\mathcal{J}^{L,l}$, viewed as a function of l, as the *backward APJN*, and to $\mathcal{J}^{l,0}$ as the *forward APJN*. At block level, we define
$$Q^b = q^{2b}, \quad P^b = p^{2b}, \quad \mathscr{J}^{b',b} = \mathcal{J}^{2b',\,2b}, \tag{7}$$
We use the same terminology for the block-level functions $\mathscr{J}^{B,b}$ and $\mathscr{J}^{b,0}$, referring to them as the *backward APJN* and *forward APJN*, respectively; the distinction from layer-level quantities will be clear from context.

3 Theory of Signal Propagation

3.1 Covariance Recurrence Relations and APJN

We now present covariance and APJN recurrences for transformers with either LayerNorm or an elementwise nonlinearity replacing LayerNorm. Under standard initialization, the same recurrences hold for multi-head attention; see Appendices B.3 and C.2.3. We then compare the resulting APJN predictions with measurements. Derivations are provided in Appendices B and C.

Result 3.1 (Covariance recurrence relations). Under Assumption 2.1, and in the large-context limit, Eq. (1) gives the recurrence relation

$$q^{l+1} = q^l + \begin{cases} \sigma_{OV}^2 \tilde{p}^l, & \text{if } l \text{ is even (attn)}, \\ \dfrac{1}{2}\sigma_{21}^2 \tilde{q}^l, & \text{if } l \text{ is odd (MLP)}, \end{cases} \tag{8}$$

and similarly

$$p^{l+1} = p^l + \begin{cases} \sigma_{OV}^2 \tilde{p}^l, & \text{if } l \text{ is even (attn)}, \\ \dfrac{1}{2}\sigma_{21}^2 \tilde{q}^l \kappa\left(\dfrac{\tilde{p}^l}{\tilde{q}^l}\right), & \text{if } l \text{ is odd (MLP)}. \end{cases} \tag{9}$$

Here $\sigma_{OV} = \sigma_O \sigma_V$ and $\sigma_{21} = \sigma_2 \sigma_1$, while $\tilde{q}^l$ and $\tilde{p}^l$ denote the covariance components after propagation through either an elementwise nonlinearity or $\mathrm{LayerNorm}$. The function κ, which comes from covariance propagation through ReLU, is given in Eq. (28) of Appendix B.1.

For an elementwise nonlinearity ϕ, the propagated covariance components are defined by

$$\tilde{\Sigma}_{st}^l = \mathbb{E}_{(h_1,h_2)\sim\mathcal{N}(0,\,\Sigma^l)}\big[\phi(h_s)\phi(h_t)\big], \qquad s,t \in \{1,2\}, \tag{10}$$

where $\tilde{\Sigma}_{11}^l = \tilde{\Sigma}_{22}^l = \tilde{q}^l$, $\tilde{\Sigma}_{12}^l = \tilde{\Sigma}_{21}^l = \tilde{p}^l$, and $\Sigma_{11}^l = \Sigma_{22}^l = q^l$, $\Sigma_{12}^l = \Sigma_{21}^l = p^l$. In the LayerNorm case, $\tilde{q}^l = 1$ and $\tilde{p}^l = p^l/q^l$, since LayerNorm rescales each activation vector to unit normalized squared norm while preserving the cosine similarity between activation vectors.

The recurrence relations in Eqs. (8) and (9) are supplemented by the initial conditions (q^0, p^0). Although permutation-symmetric input token configurations may admit negative values of p^0, we restrict the analysis to the simplified setting $p^0 \geq 0$. Related recurrence relations were derived, for example, by Cowsik et al. [2024]; our derivation uses a simplified uniform-attention approximation and extends the analysis beyond LayerNorm.

Result 3.2 (Simplified APJN recurrence). Under Assumption 2.1 and in the large-context limit, after neglecting $\mathcal{O}(n^{-1})$ attention contributions and the cross-positional Jacobian correlation term defined in Appendix C, the forward APJN satisfies

$$\mathcal{J}^{l+1,0} = \chi_{\mathcal{J}}^l \mathcal{J}^{l,0}, \qquad l \geq 0, \tag{11}$$

where

$$\chi_{\mathcal{J}}^l = \begin{cases} 1, & \text{if } l \text{ is even (attn)}, \\ 1 + \dfrac{1}{2}\sigma_{21}^2 \hat{q}^{\,l}, & \text{if } l \text{ is odd (MLP)}. \end{cases} \tag{12}$$

Here, $\hat{q}^l$ denotes the variance obtained by propagating q^l through the Jacobian of the nonlinearity ϕ or of $\mathrm{LayerNorm}$.

For elementwise functions,

$$\hat{q}^l = \mathbb{E}_{h\sim\mathcal{N}(0,\,q^l)}\left[\phi'(h)^2\right], \tag{13}$$

where the prime denotes differentiation. For $\mathrm{LayerNorm}$, $\hat{q}^l = 1/q^l$, as follows from its Jacobian. For the specific choice $\phi_\alpha(h) = \mathrm{erf}(\alpha h)$, explicit formulas for $\tilde{q}^l$, $\tilde{p}^l$, and $\hat{q}^l$ are given in Appendix E.

The APJN recurrence relation in Eq. (11) is supplemented by the initial condition $\mathcal{J}^{0,0} = 1$. The sequence $\mathcal{J}^{l,0}$ is obtained by first solving the covariance propagation equations, Eqs. (8) and (9), and then evaluating Eq. (12) using the resulting q^l. Under the approximation described above, the backward APJN is given by $\mathcal{J}^{L,l} = \mathcal{J}^{L,0}/\mathcal{J}^{l,0}$.

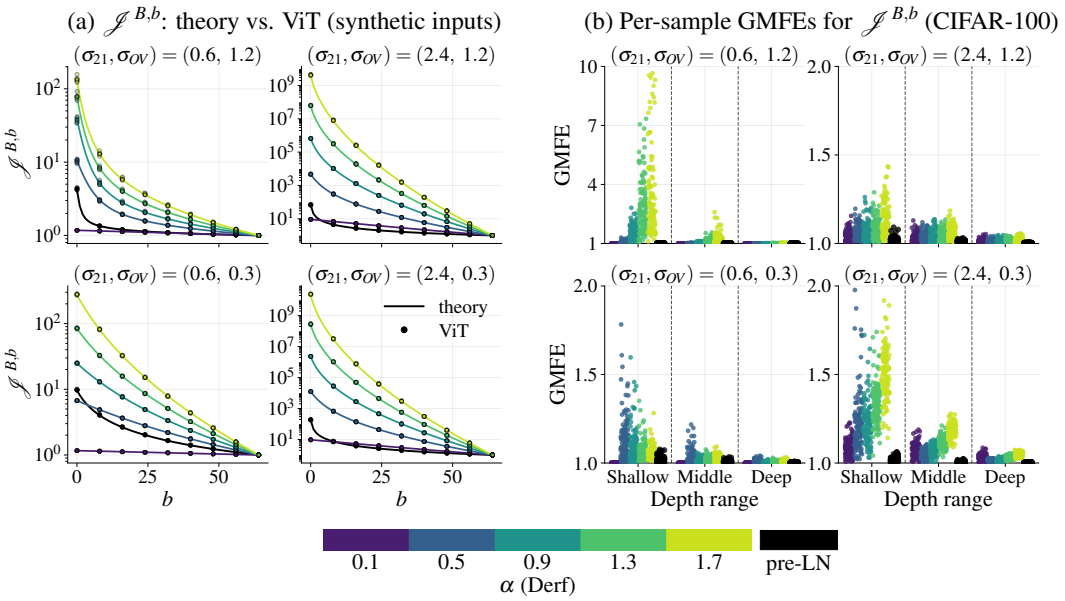

Figure 1: **(a)** Backward APJN $\mathscr{J}^{B,b}$ on synthetic permutation-symmetric inputs: ViT measurements versus theory for pre-LN and Derf variants. **(b)** Per-sample GMFE values, interpreted as typical multiplicative error factors, between theoretical and measured $\mathscr{J}^{B,b}$ values in a 128-block ViT on 120 CIFAR-100 inputs, evaluated in early, middle, and deep blocks.

In this simplified recurrence, attention affects the APJN only indirectly through its effect on covariance propagation in Eqs. (8) and (9). The recurrence is analogous to that derived in Doshi et al. [2023] for feedforward and residual networks without attention in the large-width limit. In Appendix C, we present and derive the extended recurrence, which involves, in addition to the APJN, a cross-positional Jacobian correlation term that may become non-negligible for larger values of σ_{OV}^2.[2]

Comparison with experiments. We compare the backward APJN predicted by the theoretical recurrence relations with those measured in ViT across several setups. Our main findings are summarized below; experimental details, including the ViT configuration and a formal algorithm box describing the APJN estimation procedure, are provided in Appendix A, and additional figures and results are given in Appendix G.

(1) On synthetic permutation-symmetric input token configurations, the theory accurately predicts backward APJNs in both pre-LN and Derf ViTs. We compare the backward APJNs $\mathscr{J}^{B,b}$ measured in a ViT model on synthetic permutation-symmetric inputs with $(q^0, p^0) = (1.0,\ 0.2)$ to the predictions of Eqs. (11) and (12). We consider both the pre-LN model and Derf variants with multiple values of α. As shown in Fig. 1(a), the agreement is strong, consistent with the fact that these inputs satisfy the simplifying assumptions of the theory. The corresponding values of Q^b, P^b, and P^b/Q^b are shown in Fig. 6.

(2) On CIFAR-100, the theoretical backward-APJN curves approximate measured APJNs in middle and deep layers within modest multiplicative error. We compare the theory with backward APJNs measured on CIFAR-100 samples using a 128-block ViT, for both pre-LN and Derf variants across several values of α. Since CIFAR-100 inputs do not satisfy the simplifying assumptions of the theory, we set the per-sample initial condition by taking q^0 to be the empirical average normalized self-dot product and p^0 to be the empirical average normalized cross-positional dot product at the input to the first transformer block. As shown in Fig. 1(b), the resulting theoretical curves approximate the measured backward APJNs in middle and deep layers, with per-sample geometric mean fold

[2]In all figures showing or using theoretical APJNs, we use the extended theory for $(\sigma_{21},\ \sigma_{OV}) = (0.6,\ 1.2)$ without stating this explicitly each time, since it provides a better match to the empirical APJN measurements. Fig. 11 in Appendix G shows that, in this case, the contribution of the neglected terms is no longer non-negligible.

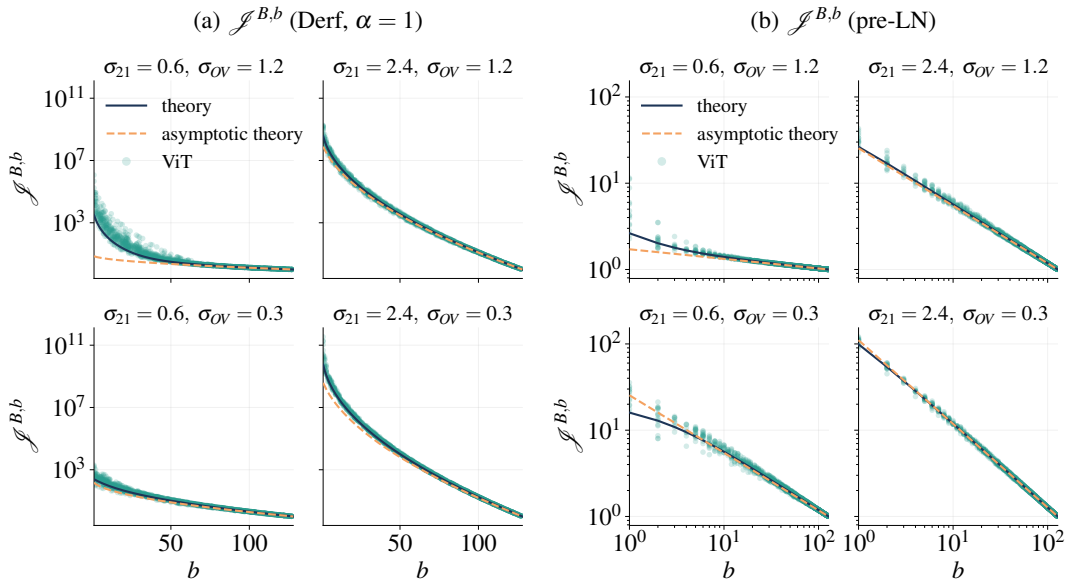

Figure 2: Backward APJNs for Derf and pre-LN at several values of $(\sigma_{21}, \sigma_{OV})$: ViT vs. theory. **(a)** Derf with $\alpha = 1$. **(b)** pre-LN. Teal circles show ViT measurements of $\mathscr{J}^{B,b}$ on 100 random CIFAR-100 samples. Black solid lines show full-theory predictions from Eqs. (8)-(12) with $(q^0, p^0) = (0.5,\ 0.25)$, and orange dashed lines show the large-b asymptotics: Eq. (18) in (a) and Eq. (16) in (b).

errors (GMFEs) typically below 1.25 and often substantially lower. GMFE is interpreted as a typical multiplicative error factor and is defined in Appendix A.

The reasonable agreement between the theory and measured backward APJNs on CIFAR-100 inputs is consistent with the observation that, as depth increases, the relative fluctuations of the self- and cross-positional dot-product statistics across positions decrease, making the empirical activations closer to the permutation-symmetric setting assumed by the theory; see Figs. 13 and 15. The most challenging case for the theory is $(\sigma_{21}, \sigma_{OV}) = (0.6,\ 1.2)$, where larger attention weights may make the APJNs more sensitive to violations of the theoretical assumptions. Representative comparisons and analogous forward-APJN results are shown in Figs. 7 and 9.

Fig. 2 further illustrates the depth dependence and sample-to-sample variability of backward APJNs on 100 CIFAR-100 training samples, comparing ViT measurements with theory for Derf with $\alpha = 1$ and for pre-LN. The theoretical curves are computed using the fixed initial condition $(q^0, p^0) = (0.5,\ 0.25)$, chosen to be close to the average of the empirical per-sample (q^0, p^0) values. Agreement improves in deeper blocks, where $\mathscr{J}^{B,b}$ has smaller sample-to-sample variance; see Figs. 14 and 16 Fig. 10 shows analogous plots for Derf models with different values of α.

3.2 LayerNorm vs. tanh-Like Nonlinearity: Asymptotic Behavior

We now summarize large-depth APJN behavior for LayerNorm transformers and for transformers where LayerNorm is replaced by a tanh-like nonlinearity. Since attention and MLP layers affect covariance dynamics and APJNs differently, we work at the block level, using Q^b, P^b, $\mathscr{J}^{b,0}$, and $\mathscr{J}^{B,b}$ from Eq. (7).

In both architectures, the activation variance Q^b grows linearly with depth. The difference in APJN behavior comes from the dependence of $\hat{q}$ on q: for LayerNorm, $\hat{q} = 1/q$, leading to power-law APJN growth, whereas for tanh-like nonlinearities, $\hat{q} \sim 1/\sqrt{q}$ at large depth, leading to stretched-exponential APJN growth. We refer the reader to Appendix D for details.

Result 3.3 (Asymptotic growth of Q^b). Using the covariance recurrence in Result 3.1, assume that $\tilde{q}^l$ and $\tilde{p}^l$ approach limiting values $\tilde{q}_\star$ and $\tilde{p}_\star$ as $l \to \infty$. Then, for both LayerNorm and tanh-like

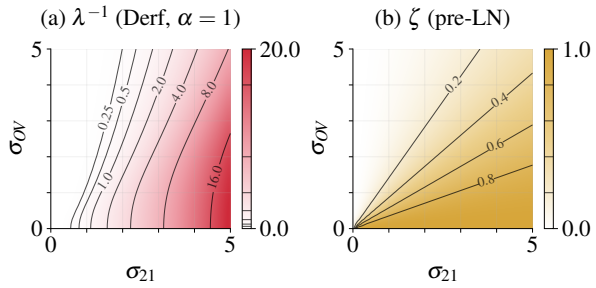

Figure 3: Theoretical values of the scale parameter λ^{-1} and the critical exponent ζ that determine the asymptotic behavior of the forward and backward APJNs. **(a)** The scale parameter λ^{-1} as a function of σ_{21} and σ_{OV} for Derf with $\alpha = 1$. **(b)** The critical exponent ζ as a function of σ_{21} and σ_{OV} for pre-LN.

nonlinearities, at large block index b,

$$Q^b \sim b\left(\frac{1}{2}\sigma_{21}^2\tilde{q}_\star + \sigma_{OV}^2\tilde{p}_\star\right). \tag{14}$$

In both cases, $\tilde{q}_\star = 1$. The value of $\tilde{p}_\star$ is determined by the asymptotic cosine similarity between activation vectors at different positions, denoted by $c_\star$, via $\tilde{p}_\star = \tilde{p}(c_\star)$, where $\tilde{p}(c) = \frac{2}{\pi}\arcsin c$ for $\tanh$-like nonlinearities and $\tilde{p}(c) = c$ for LayerNorm. The value $c_\star$ satisfies

$$c_\star = \frac{\frac{1}{2}\sigma_{21}^2\kappa\left(\tilde{p}(c_\star)\right) + \sigma_{OV}^2\tilde{p}(c_\star)}{\frac{1}{2}\sigma_{21}^2 + \sigma_{OV}^2\tilde{p}(c_\star)}. \tag{15}$$

The linear growth in Eq. (14) follows directly from the covariance recurrence in Eq. (8): once $\tilde{q}^l$ and $\tilde{p}^l$ approach limiting values, each transformer block adds an asymptotically constant contribution to Q^b. In both architectures considered here, $\tilde{q}_\star = 1$; this is immediate for LayerNorm and follows from saturation of ϕ at large pre-activations for $\tanh$-like nonlinearities.

For LayerNorm, the unique solution of Eq. (15) is $c_\star = 1$, so $\tilde{p}_\star = 1$. For $\tanh$-like nonlinearities with $\sigma_{21}^2 \neq 0$, Eq. (15) has the solution $c_\star = 1$ and another solution with $c_\star < 1$; the former is unstable, so the system generically converges to the stable solution with $c_\star < 1$. The derivation of Eq. (15) and the stability analysis are given in Appendix D.2.1.

Result 3.4 (Pre-LN: power-law APJN growth). Combining the asymptotic growth of Q^b in Result 3.3 with the simplified APJN recurrence of Result 3.2, the pre-LN APJNs satisfy, at large b,

$$\mathscr{J}^{b,0} \sim b^\zeta, \qquad \mathscr{J}^{B,b} \sim \left(\frac{B}{b}\right)^\zeta, \tag{16}$$

where

$$\zeta = \frac{\frac{1}{2}\sigma_{21}^2}{\frac{1}{2}\sigma_{21}^2 + \sigma_{OV}^2}. \tag{17}$$

Thus, pre-LN lies in the *critical* regime of Doshi et al. [2023], characterized by power-law growth of the APJN with critical exponent ζ. Specifically, the forward APJN increases with the block index, whereas the backward APJN increases toward earlier blocks. The derivation is provided in Appendix D.

Result 3.5 ($\tanh$-like nonlinearities: stretched-exponential APJN growth). For $\tanh$-like nonlinearities ϕ_α, the APJNs satisfy, at large b,

$$\mathscr{J}^{b,0} \sim e^{\sqrt{b/\lambda}}, \qquad \mathscr{J}^{B,b} \sim e^{(\sqrt{B}-\sqrt{b})/\sqrt{\lambda}}, \tag{18}$$

with scale parameter

$$\lambda^{-1} = \frac{C_\alpha^2\sigma_{21}^4}{\frac{1}{2}\sigma_{21}^2 + \sigma_{OV}^2\tilde{p}_\star}, \qquad C_\alpha = \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}}\phi_\alpha'(h)^2\,dh. \tag{19}$$

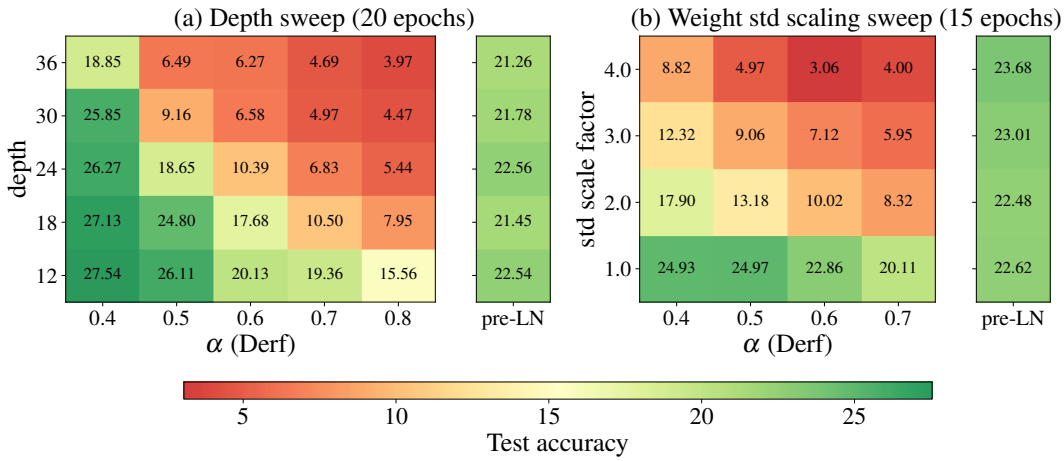

Figure 4: Training stability comparison between Derf and pre-LN ViTs. **(a)** Test accuracy on CIFAR-100 after 20 epochs of training versus model depth. **(b)** Test accuracy on CIFAR-100 after 15 epochs of training versus the MLP and attention weight standard deviation scale (depth $B = 12$).

Following Doshi et al. [2023], the network lies in the *subcritical* regime at initialization. In this regime, the forward APJN grows asymptotically faster than any power law, and therefore faster than in pre-LN. Likewise, for sufficiently large network depth, the backward APJN increases more rapidly than in pre-LN from the final block toward earlier blocks. Moreover, unlike the bounded pre-LN exponent ζ in Eq. (17), the scale parameter λ^{-1} is unbounded, so the stretched-exponential APJN growth can be made arbitrarily steep by varying the weight initialization. The derivation, including the refined asymptotic expression with the power-law prefactor in Eq. (77), is provided in Appendix D.1.2.

Role of α in DyT/Derf. Decreasing α in the nonlinearity ϕ_α reduces the effective residual-branch contribution and therefore slows the growth of both Q^b and the APJN. This has two consequences. First, the asymptotic expression in Eq. (18) becomes valid only at greater depths for smaller α. Second, at fixed depth, sufficiently small α can make Derf APJNs comparable to pre-LN, as in Fig. 1(a), but matching pre-LN at increasing depth requires progressively smaller α.

Comparison with experiment. Fig. 2 compares backward APJN values measured in ViT for pre-LN and Derf with the corresponding asymptotic predictions: Eq. (16) for pre-LN, and for Derf the refined asymptotic expression in Eq. (77), including the power-law prefactor, derived in Appendix D.1.2. In both cases, the asymptotic expressions become increasingly accurate as the depth increases. Fig. 3 shows the theoretical values of the scale parameter λ^{-1} and the critical exponent ζ as functions of σ_{21} and σ_{OV}. Consistent with Eqs. (16) and (18), these quantities determine the slope of the asymptotic APJN curves in the two regimes.

4 Training Stability Experiments

The theory above suggests several empirical predictions. Since the backward APJN provides a scalar measure of gradient amplification during backpropagation, Derf is expected to be less robust to hyperparameter choices than pre-LN. Specifically, increasing model depth or the weight-initialization scale is expected to impair trainability more strongly in Derf than in pre-LN, whereas decreasing α is expected to improve trainability in Derf by reducing gradient amplification, both across transformer blocks and at the final Derf layer; the latter effect is discussed in Appendix F.

We test these predictions in ViT models trained on CIFAR-100, comparing Derf and pre-LN during early training. Although fully training a ViT model on CIFAR-100 may require hundreds of epochs, we typically truncate training after 10–20 epochs, which is sufficient to identify configurations with pathologically slow convergence. Model and training details are provided in Appendix A. Supplementary experiments and example training/test curves are provided in Appendix H.

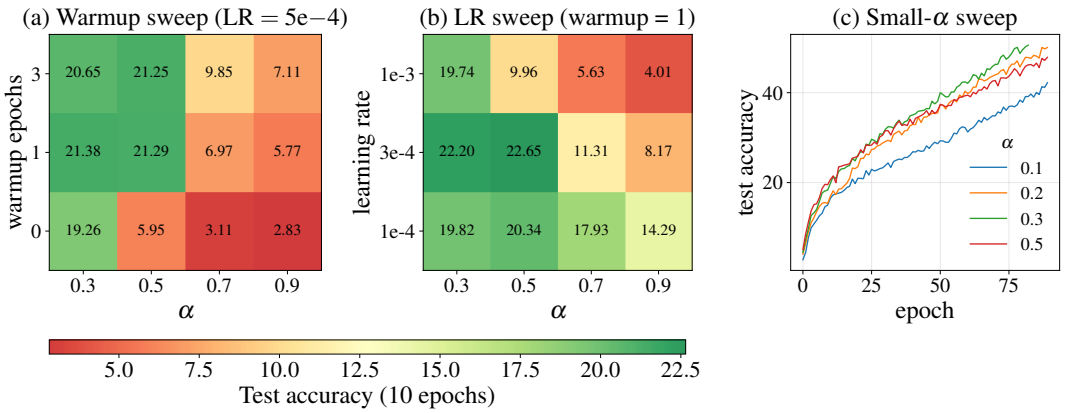

Figure 5: Effects of warmup, learning rate, and small initial α in Derf ViTs on CIFAR-100. **(a)** Test accuracy after 10 epochs versus number of warmup epochs. **(b)** Test accuracy after 10 epochs versus learning rate. **(c)** Test accuracy over 90 epochs for Derf models initialized with small α.

The main results are as follows.

(1) Derf trainability is more sensitive to model depth than pre-LN trainability. Fig. 4(a) shows that Derf models train more slowly as depth increases and eventually fail to converge beyond a critical depth, with smaller initialization values of α pushing this instability to greater depth. We choose the range of α values so that this instability boundary is visible. Although we expect the destabilization to occur even for smaller values of α, it becomes less relevant at depths typically used in practice. By comparison, the pre-LN baseline, with the same weight initialization, maintains an approximately constant convergence rate across the full tested depth range without hyperparameter tuning. Repeating the depth sweep with DyT yields qualitatively similar results to Derf (Appendix H, Fig. 24).

(2) Derf trainability is more sensitive to weight initialization than pre-LN trainability. We vary the initialization scale of the MLP and attention weights by a common factor up to 4.0, excluding query and key weights to avoid entropy-collapse effects [Zhai et al., 2023, Giorlandino and Goldt, 2026], while keeping the model depth fixed at $B = 12$. Fig. 4(b) shows that Derf variants with larger α tend to become unstable at smaller weight initialization scales, whereas the pre-LN baseline remains stable over the entire range considered.

(3) Derf models with larger α benefit from a smaller learning rate or a longer warmup. At fixed depth $B = 12$, we separately sweep the number of warmup epochs and the maximum learning rate. Figs. 5(a) and (b) show that both changes improve the trainability of Derf models at larger α, with reducing the learning rate having a particularly strong effect.

(4) Choosing an overly small initial value of α in Derf models can slow convergence. Improving stability by decreasing α comes with a trade-off. Fig. 5(c) shows that choosing α to be too small, such as $\alpha = 0.1$, can substantially slow convergence. Fig. 26 in Appendix H shows that this effect is robust across different settings.

Overall, these results agree qualitatively with the predicted role of gradient amplification, which we measure directly in Appendix H.1. Although α is trainable in Derf models, its initialization remains important; freezing α can further worsen convergence, as shown in Fig. 23.

5 Conclusion

We extended the APJN framework to transformers with uniform bidirectional attention and permutation-symmetric input token configurations. Despite these simplifying assumptions, the theory reasonably approximates APJNs measured in deeper ViT layers on real inputs. We derived APJN asymptotics for both LayerNorm transformers and transformers in which LayerNorm is re-

placed by $\tanh$-like nonlinearities, showing that the latter exhibit stretched-exponential APJN growth, as in residual networks without attention.

The subcritical behavior of DyT/Derf, together with the overall gradient rescaling induced by the final normalization layer, can lead to larger gradients and thus requires more careful tuning of initialization parameters, such as the weight standard deviation and the parameter α. Across several experiments comparing the training stability of Derf models initialized with different values of α with that of the pre-LN baseline, we observed the trends predicted by the theory.

Limitations. Our experiments use ViT models on CIFAR-100, limiting conclusions about other architectures, modalities, or tasks. Preliminary next-token-prediction experiments on TinyStories did not identify a comparable instability boundary (Appendix H.3), leaving open whether the same instability pattern occurs in causal language models. Although truncated runs reveal early instability boundaries, they do not assess long-term training dynamics or final performance. Theoretically, extending the framework to more complex token geometries remains important. Finally, since the theory describes signal propagation at initialization, it explains trends in trainability and gradient amplification, but not full training dynamics or final performance.

6 Acknowledgements and Disclosures

This research received no external funding and was supported by the author's personal resources. The author declares no competing interests.

References

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

Thomas Bachlechner, Bodhisattwa Prasad Majumder, Huanru Henry Mao, Garrison W. Cottrell, and Julian McAuley. Rezero is all you need: Fast convergence at large depth, 2020. URL https://arxiv.org/abs/2003.04887.

Mingzhi Chen, Taiming Lu, Jiachen Zhu, Mingjie Sun, and Zhuang Liu. Stronger normalization-free transformers, 2025. URL https://arxiv.org/abs/2512.10938.

Aditya Cowsik, Tamra Nebabu, Xiao-Liang Qi, and Surya Ganguli. Geometric dynamics of signal propagation predict trainability of transformers, 2024. URL https://arxiv.org/abs/2403.02579.

Darshil Doshi, Tianyu He, and Andrey Gromov. Critical initialization of wide and deep neural networks through partial jacobians: General theory and applications, 2023. URL https://arxiv.org/abs/2111.12143.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale, 2021. URL https://arxiv.org/abs/2010.11929.

Alessio Giorlandino and Sebastian Goldt. Two failure modes of deep transformers and how to avoid them: a unified theory of signal propagation at initialisation, 2026. URL https://arxiv.org/abs/2505.24333.

Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In Yee Whye Teh and Mike Titterington, editors, *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, volume 9 of *Proceedings of Machine Learning Research*, pages 249–256, Chia Laguna Resort, Sardinia, Italy, 13–15 May 2010. PMLR. URL https://proceedings.mlr.press/v9/glorot10a.html.

Judy Hoffman, Daniel A. Roberts, and Sho Yaida. Robust learning with jacobian regularization, 2019. URL https://arxiv.org/abs/1908.02729.

Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. Deep networks with stochastic depth, 2016. URL https://arxiv.org/abs/1603.09382.

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, 2024. URL https://arxiv.org/abs/2403.09635.

Liyuan Liu, Xiaodong Liu, Jianfeng Gao, Weizhu Chen, and Jiawei Han. Understanding the difficulty of training transformers, 2023. URL https://arxiv.org/abs/2004.08249.

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, 2022. URL https://arxiv.org/abs/2206.03126.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In Sanjoy Dasgupta and David McAllester, editors, *Proceedings of the 30th International Conference on Machine Learning*, volume 28 of *Proceedings of Machine Learning Research*, pages 1310–1318, Atlanta, Georgia, USA, 17–19 Jun 2013. PMLR. URL https://proceedings.mlr.press/v28/pascanu13.html.

Jeffrey Pennington, Samuel S. Schoenholz, and Surya Ganguli. Resurrecting the sigmoid in deep learning through dynamical isometry: theory and practice, 2017. URL https://arxiv.org/abs/1711.04735.

Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos, 2016. URL https://arxiv.org/abs/1606.05340.

Samuel S. Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation, 2017. URL https://arxiv.org/abs/1611.01232.

Sam Shleifer, Jason Weston, and Myle Ott. Normformer: Improved transformer pretraining with extra normalization, 2021. URL https://arxiv.org/abs/2110.09456.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Hongyu Wang, Shuming Ma, Li Dong, Shaohan Huang, Dongdong Zhang, and Furu Wei. Deepnet: Scaling transformers to 1,000 layers, 2022. URL https://arxiv.org/abs/2203.00555.

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. On layer normalization in the transformer architecture, 2020. URL https://arxiv.org/abs/2002.04745.

Greg Yang and Samuel S. Schoenholz. Mean field residual networks: On the edge of chaos, 2017. URL https://arxiv.org/abs/1712.08969.

Shuangfei Zhai, Tatiana Likhomanenko, Etai Littwin, Dan Busbridge, Jason Ramapuram, Yizhe Zhang, Jiatao Gu, and Josh Susskind. Stabilizing transformer training by preventing attention entropy collapse, 2023. URL https://arxiv.org/abs/2303.06296.

Biao Zhang and Rico Sennrich. Root mean square layer normalization, 2019. URL https://arxiv.org/abs/1910.07467.

Jiachen Zhu, Xinlei Chen, Kaiming He, Yann LeCun, and Zhuang Liu. Transformers without normalization, 2025. URL https://arxiv.org/abs/2503.10622.

Appendix

The appendix is organized as follows. Sec. A provides experimental details for the results presented in the main text. Sec. B contains derivations of the covariance propagation equations. Sec. C derives the APJN recurrence relations, and Sec. D presents derivations of the asymptotic APJN behavior. Sec. E provides explicit expressions for the covariance components propagated through the erf nonlinearity that appear in the recurrence relations in the main text. Sec. F discusses the role of the final normalization layer following the transformer blocks. Sec. G provides supplementary material on signal propagation experiments. Finally, Sec. H presents supplementary training stability experiments.

A Experimental Details

Resources. All experiments were conducted in Google Colab. For the APJN computation experiments, we used a single T4 or L4 GPU, taking approximately 48 hours in total. The most computationally intensive experiments were the training-stability runs, for which we used a single A100 GPU to train models with up to 24 transformer blocks and a single RTX PRO 6000 GPU (G4 VM) to train the deeper models. Training a ViT model with 12 blocks for 30 epochs takes around 1 hour on an A100 GPU. The experiments included in the paper took approximately 200 GPU-hours in total.

ViT model. As the base model, we use `vit_base_patch16_224` from `timm`. By default, this model has 12 transformer blocks, hidden dimension $d = 768$, GELU activations in its MLP layers, context size $n = 196$ on CIFAR-100, and weights initialized with standard deviation 0.02. LayerNorm is applied to the inputs of the residual branches and to the final output. In the Derf models, we replace all of these LayerNorm layers, including the final one, with the Derf nonlinearity parameterized by α, following Zhu et al. [2025], Chen et al. [2025].

For the experiments in Section 3, we replace the GELU activations in the MLP layers with ReLU. To match the weight initialization of the ViT model in our theory, we set $\sigma_O = \sigma_V = \sigma_Q = \sigma_K = \sigma_1 = 0.02 \times \sqrt{d}$ and $\sigma_2 = 0.02 \times \sqrt{4d}$ in the theoretical model. Consequently, $\sigma_{OV} \approx 0.31$ and $\sigma_{21} \approx 0.61$.

In experiments where we vary σ_{OV}, we scale σ_O and σ_V by the same factor while keeping σ_Q and σ_K fixed. Analogously, when varying σ_{21}, we scale σ_1 and σ_2 by the same factor.

Training stability experiments. Unless stated otherwise, we train the models using AdamW ($\beta_1 = 0.9$, $\beta_2 = 0.999$) with weight decay 0.05. We use a linear warmup schedule, after which the learning rate is held constant. We do not use stochastic depth [Huang et al., 2016], in order to properly capture depth-scaling effects.

The training-stability sweeps are single-run experiments for each configuration. We therefore interpret these sweeps as qualitative evidence for large instability trends and approximate instability boundaries, rather than as statistically precise estimates of final accuracy or small performance differences.

The peak learning rate, number of warmup epochs, and batch size used in each training stability experiment in the main text are listed below:

- Fig. 4(a): $\mathrm{LR} = 3 \times 10^{-4},\ \mathrm{warmup} = 3,\ \mathrm{batch} = 128$;
- Fig. 4(b): $\mathrm{LR} = 3 \times 10^{-4},\ \mathrm{warmup} = 3,\ \mathrm{batch} = 256$;
- Fig. 5(a): $\mathrm{LR} = 5 \times 10^{-4},\ \mathrm{batch} = 256$;
- Fig. 5(b): $\mathrm{warmup} = 1,\ \mathrm{batch} = 256$;
- Fig. 5(c): $\mathrm{LR} = 3 \times 10^{-4},\ \mathrm{warmup} = 10,\ \mathrm{batch} = 256$.

Images from CIFAR-100 were resized to 224×224. During training, we used the default augmentations from the `timm` repository: color jitter with magnitude 0.4, RandAugment (`rand-m9-mstd0.5-inc1`), bicubic interpolation, random erasing with probability 0.25 in pixel mode and count 1, and mixup/cutmix with mixup 0.8, cutmix 1.0, application probability 1.0, switch probability 0.5, and batch mode. We trained using `SoftTargetCrossEntropy`, with label smoothing set to 0.1.

Fig. 1(a). Backward APJN: ViT (synthetic permutation-symmetric inputs) vs. theory. The APJN values are computed using 8 generated permutation-symmetric inputs with $(q^0, p^0) = (1.0,\ 0.2)$. The components of each input token are drawn from a Gaussian distribution. For each input, we compute the APJNs across the full set of models: pre-LN and Derf variants with several values of α. Note that the variance across inputs is visible only for $(\sigma_{21}, \sigma_{OV}) = (0.6, 1.2)$. The Jacobian Frobenius norms are estimated using Hutchinson's method [Hoffman et al., 2019, Doshi et al., 2023] with 10 random draws, following the procedure described in Algorithm 1. These Frobenius norms are then averaged over 5 random weight initializations to estimate the APJNs.

Fig. 1(b). Backward APJN: ViT with CIFAR-100 inputs vs. theory. For a randomly selected CIFAR-100 training example, we measure APJNs at every fourth transformer block of a 128-block ViT model, excluding the endpoints. The geometric mean fold error (GMFE), used to compare the

Algorithm 1 Hutchinson estimation of the backward APJN at initialization

Input: A B-block ViT; a fixed input (image x or synthetic tokens X^0); indices $\mathcal{B} \subseteq \{0, \dots, B\}$ at which to estimate the APJN; M weight initializations; K probes.

Output: Estimates $\widehat{\mathscr{J}}^{B,b}$ of $\mathscr{J}^{B,b} = \mathcal{J}^{2B,2b}$ for $b \in \mathcal{B}$.

Let $X^0 \in \mathbb{R}^{n\times d}$ be the input to the first block and X^b the activations after b blocks. Thus X^B precedes the final norm and head.

$\widehat{\mathscr{J}}^{B,b} \leftarrow 0$ for all $b \in \mathcal{B}$
If $B \in \mathcal{B}$, set $\widehat{\mathscr{J}}^{B,B} \leftarrow 1$
for $m = 1, \dots, M$ **do**
    Independently initialize the ViT weights θ_m.
    Obtain X^0 from x (or the supplied tokens) and enable its gradients.
    Compute $X^{j+1} \leftarrow \mathrm{Block}_j(X^j; \theta_m)$ for $j = 0, \dots, B-1$;
    retain all X^j and the computation graph for repeated differentiation.
    for $b \in \mathcal{B} \setminus \{B\}$ **do**
        for $k = 1, \dots, K$ **do**
            Draw $R \in \mathbb{R}^{n\times d}$ with independent $R_{si} \sim \mathcal{N}(0, 1)$.
            $G \leftarrow \dfrac{\partial \langle X^B, R\rangle_F}{\partial X^b}$ ▷ one vector–Jacobian product
            $\widehat{\mathscr{J}}^{B,b} \leftarrow \widehat{\mathscr{J}}^{B,b} + \|G\|_F^2/(MKnd)$
        end for
    end for
end for
return $\{\widehat{\mathscr{J}}^{B,b}\}_{b\in\mathcal{B}}$

For $J^{2B,2b} = \partial \operatorname{vec}(X^B)/\partial \operatorname{vec}(X^b)$, the Gaussian probes satisfy

$$\mathbb{E}_R\|G\|_F^2 = \|J^{2B,2b}\|_F^2, \qquad \mathscr{J}^{B,b} = \frac{1}{nd}\,\mathbb{E}_\theta\|J^{2B,2b}\|_F^2.$$

Thus the returned estimates are unbiased.

Paper settings: $K = 10$ and $M \in \{4, 5, 8\}$, depending on the experiment.

measured APJN values with theory, is defined as

$$\mathrm{GMFE}(q^0, p^0) = \exp\left(\frac{1}{|S|}\sum_{b\in S}\left|\log\frac{\mathscr{J}^{B,b}_{\mathrm{theory}}(q^0, p^0)}{\mathscr{J}^{B,b}_{\mathrm{ViT}}}\right|\right), \tag{20}$$

where S is a set of blocks in the first third of the network (early), the second third (middle), or the last third (deep). To compute the theoretical values, we estimate q^0 and p^0 for the given input as

$$q^0 = \frac{1}{n}\sum_{s=1}^{n}\frac{\boldsymbol{h}_s^0\cdot\boldsymbol{h}_s^0}{d}, \quad p^0 = \frac{1}{n(n-1)}\sum_{s\neq t}\frac{\boldsymbol{h}_s^0\cdot\boldsymbol{h}_t^0}{d}. \tag{21}$$

We repeat this measurement and comparison for multiple CIFAR-100 samples and multiple values of $(\sigma_{21}, \sigma_{OV})$, as shown in Fig. 1(b). The APJNs are obtained by averaging Jacobian Frobenius norms over 8 independent model-weight initializations. Each Jacobian Frobenius norm is estimated using Hutchinson's method with 10 random matrix draws.

The input statistics used to evaluate the initial condition (q^0, p^0) are extracted immediately before the first transformer block, after the CIFAR-100 image has passed through the initial patch embedding and positional embedding.

Fig. 2. APJNs: ViT (CIFAR-100 inputs) vs. theory vs. asymptotic theory. For each CIFAR-100 input, we randomly choose 16 blocks at which the APJNs are evaluated. The APJNs are obtained by averaging Jacobian Frobenius norms over 4 model weight initializations. Each Jacobian Frobenius norm is estimated using Hutchinson's method with 10 random matrix draws.

B Covariance Propagation

This section derives Result 3.1.

One of the main ingredients in the signal propagation calculation is covariance propagation through elementwise nonlinearities followed by linear layers. We illustrate this for a transformation of the form $W\phi(\boldsymbol{h})$, where the entries of W are i.i.d. $\mathcal{N}(0, \sigma_W^2/d)$. Consider two activation vectors at different positions, $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$, with component-wise covariance matrix Σ, such that $\Sigma_{11} = \Sigma_{22} = q$ and $\Sigma_{12} = \Sigma_{21} = p$. Equivalently, for any component index i, $(h_{1,i}, h_{2,i}) \sim \mathcal{N}(0, \Sigma)$, and different components are uncorrelated. The covariance after applying the transformation is then $\sigma_W^2 \Sigma^\phi$, where

$$\Sigma^\phi_{ab} = \mathbb{E}_{(h_1,h_2)\sim\mathcal{N}(0,\,\Sigma)}\big[\phi(h_a)\,\phi(h_b)\big], \qquad a, b \in \{1, 2\}. \tag{22}$$

In this expression, each h_a is a scalar random variable; the component index i is suppressed.

In what follows, we derive the recurrence relations given in Eqs. (8) and (9).

B.1 Covariance Propagation Through MLP

Given the MLP-layer activation vector update in Eq. (1),

$$\boldsymbol{h}_a^{l+1} = \boldsymbol{h}_a^l + W_2^l\,\mathrm{ReLU}\big(W_1^l \tilde{\boldsymbol{h}}_a^l\big), \tag{23}$$

we split the computation into three steps.

1. Covariance propagation through $W_1\phi(\cdot)$ or $W_1\mathrm{LayerNorm}(\cdot)$. Here $W_1 \in \mathbb{R}^{d_{\text{ff}}\times d}$ has i.i.d. entries $\mathcal{N}(0,\, \sigma_1^2/d)$. Assume the input activations have covariance matrix Σ_{ab}, $a, b \in \{1, 2\}$, with $\Sigma_{11} = \Sigma_{22} = q$ and $\Sigma_{12} = \Sigma_{21} = p$. The propagated covariance is

$$\Sigma'_{ab} = \frac{1}{d_{\text{ff}}}\mathbb{E}_\theta\left[(W_1\tilde{\boldsymbol{h}}_a)\cdot(W_1\tilde{\boldsymbol{h}}_b)\right] = \frac{\sigma_1^2}{d}\mathbb{E}_\theta\left[\tilde{\boldsymbol{h}}_a\cdot\tilde{\boldsymbol{h}}_b\right]. \tag{24}$$

For $\mathrm{LayerNorm}$, evaluating the dot product gives $q' = \sigma_1^2$ and $p' = \sigma_1^2\, p/q$. For elementwise functions, we obtain:

$$\Sigma'_{ab} = \sigma_1^2\,\mathbb{E}_{(h_1,h_2)\sim\mathcal{N}(0,\Sigma)}\left[\phi(h_a)\phi(h_b)\right]. \tag{25}$$

2. Covariance propagation through $W_2\mathrm{ReLU}(\cdot)$. Here $W_2 \in \mathbb{R}^{d\times d_{\text{ff}}}$ has i.i.d. entries $\mathcal{N}(0, \sigma_2^2/d_{\text{ff}})$. Similarly, given an input covariance matrix Σ_{ab}, the propagated covariance is

$$\Sigma'_{ab} = \sigma_2^2\,\mathbb{E}_{(h_1,h_2)\sim\mathcal{N}(0,\Sigma)}\left[\mathrm{ReLU}(h_a)\mathrm{ReLU}(h_b)\right]. \tag{26}$$

These Gaussian integrals can be evaluated explicitly, giving:

$$q' = \frac{1}{2}\sigma_2^2 q, \quad p' = \frac{1}{2}\sigma_2^2\kappa\left(\frac{p}{q}\right)q, \tag{27}$$

where

$$\kappa(\rho) = \frac{1}{\pi}\left(\sqrt{1-\rho^2} + \rho(\pi - \arccos\rho)\right). \tag{28}$$

3. Residual addition. The residual addition adds the covariance contributions from the residual stream and the residual branch, because they are independent.

Combining the results above gives the MLP contributions in Eqs. (8) and (9).

B.2 Covariance Propagation Through Attention

Given the attention-layer activation update in Eq. (1)

$$\boldsymbol{h}_a^{l+1} = \boldsymbol{h}_a^l + W_O^l W_V^l \sum_{b=1}^n A_{ab}^l\,\tilde{\boldsymbol{h}}_b^l \tag{29}$$

we split the computation into two steps.

1. Covariance propagation through attention. Define $\boldsymbol{h}'_a = W_O W_V \sum_{b=1}^n A_{ab}\,\tilde{\boldsymbol{h}}_b$. Assume the input activations have covariance matrix Σ_{ab}, $a, b = 1, \ldots, n$, with $\Sigma_{aa} = q$ and $\Sigma_{ab} = p$ for $a \neq b$. In the uniform-attention approximation [Noci et al., 2022], the propagated covariance is

$$\Sigma'_{ab} = \frac{1}{d}\mathbb{E}_\theta\left[\boldsymbol{h}'_a\cdot\boldsymbol{h}'_b\right] = \sigma_{OV}^2\sum_{t,s=1}^n \frac{1}{n^2}\mathbb{E}_\theta\left[\frac{\tilde{\boldsymbol{h}}_t\cdot\tilde{\boldsymbol{h}}_s}{d}\right]. \tag{30}$$

Let $\tilde{\Sigma}_{ts} = \mathbb{E}_\theta\left[\tilde{\boldsymbol{h}}_t \cdot \tilde{\boldsymbol{h}}_s/d\right]$, with $t, s = 1, \ldots, n$, denote the covariance after propagation through ϕ, whose entries satisfy $\tilde{\Sigma}_{tt} = \tilde{q}$ and $\tilde{\Sigma}_{ts} = \tilde{p}$ for $t \neq s$. As in the previous section, it is given by

$$\tilde{\Sigma}_{ts} = \mathbb{E}_{\boldsymbol{h}\sim\mathcal{N}(0,\Sigma)}\left[\phi(h_t)\phi(h_s)\right]. \tag{31}$$

Therefore,

$$\Sigma'_{ab} = \frac{\sigma_{OV}^2}{n^2}\sum_{t',s'=1}^{n}\tilde{\Sigma}_{t's'} = \sigma_{OV}^2\left(\frac{1}{n}\tilde{q} + \frac{n-1}{n}\tilde{p}\right). \tag{32}$$

Thus, for large context size n, one obtains approximately:

$$q' \approx \sigma_{OV}^2\tilde{p}, \quad p' \approx \sigma_{OV}^2\tilde{p}. \tag{33}$$

2. Residual addition. The residual addition adds the covariance contributions from the residual stream and the residual branch, because they are independent:

$$q^{l+1} \approx q^l + \sigma_{OV}^2\tilde{p}, \quad p^{l+1} \approx p^l + \sigma_{OV}^2\tilde{p}. \tag{34}$$

B.3 Covariance Propagation Through Multi-Head Attention

In the case of multiple attention heads, Eq. (34) remains unchanged under the standard weight initialization scheme. Define $\boldsymbol{h}'_a = \sum_{h=1}^{H} W_O^h W_V^h \sum_{b=1}^{n} A_{ab}^h \tilde{\boldsymbol{h}}_b$, where H denotes the total number of attention heads, and the attention scores now depend on the head index h due to separate query and key embeddings:

$$A_{ab}^h = \frac{\exp\left((W_Q^h\tilde{\boldsymbol{h}}_a)\cdot(W_K^h\tilde{\boldsymbol{h}}_b)/\sqrt{d_\text{h}}\right)}{\sum_{c=1}^{n}\exp\left((W_Q^h\tilde{\boldsymbol{h}}_a)\cdot(W_K^h\tilde{\boldsymbol{h}}_c)/\sqrt{d_\text{h}}\right)}. \tag{35}$$

Here $d_\text{h} = d/H$, and $W_Q^h \in \mathbb{R}^{d_\text{h}\times d}$, $W_K^h \in \mathbb{R}^{d_\text{h}\times d}$, $W_V^h \in \mathbb{R}^{d_\text{h}\times d}$, and $W_O^h \in \mathbb{R}^{d\times d_\text{h}}$ have i.i.d. Gaussian entries with zero mean and variances σ_Q^2/d, σ_K^2/d, σ_V^2/d, and σ_O^2/d, respectively. Note that, in standard Transformer implementations, the variance of $W_O^h \in \mathbb{R}^{d\times d_\text{h}}$ scales as $1/d$, which is crucial for this derivation. Let us denote $\tilde{\tilde{\boldsymbol{h}}}_a^h = \sum_{b=1}^{n} A_{ab}^h \tilde{\boldsymbol{h}}_b$. Then, in the uniform-attention approximation, the propagated covariance is

$$\begin{aligned}\Sigma'_{ab} &= \frac{1}{d}\mathbb{E}_\theta\left[\boldsymbol{h}'_a\cdot\boldsymbol{h}'_b\right] = \sum_{h=1}^{H}\frac{\sigma_O^2}{Hd_\text{h}}\mathbb{E}_\theta\left[\left(W_V\tilde{\tilde{\boldsymbol{h}}}_a^h\right)\cdot\left(W_V\tilde{\tilde{\boldsymbol{h}}}_b^h\right)\right] = \frac{1}{H}\sum_{h=1}^{H}\frac{\sigma_{OV}^2}{d}\mathbb{E}_\theta\left[\tilde{\tilde{\boldsymbol{h}}}_a^h\cdot\tilde{\tilde{\boldsymbol{h}}}_b^h\right] = \\ &= \frac{\sigma_{OV}^2}{H}\sum_{h=1}^{H}\sum_{t,s=1}^{n}\frac{1}{n^2}\mathbb{E}_\theta\left[\frac{\tilde{\boldsymbol{h}}_t\cdot\tilde{\boldsymbol{h}}_s}{d}\right] = \sigma_{OV}^2\sum_{t,s=1}^{n}\frac{1}{n^2}\mathbb{E}_\theta\left[\frac{\tilde{\boldsymbol{h}}_t\cdot\tilde{\boldsymbol{h}}_s}{d}\right].\end{aligned} \tag{36}$$

In the chain of equations above, we performed the weight averages sequentially to keep track of the factors of d, d_h, and H: first over W_O, and then over W_V. Thus, we obtain an equation analogous to Eq. (30). This proves that the covariance update for multi-head attention is the same as for single-head attention in Eq. (34), in the uniform-attention approximation.

C APJN Recurrence Relations

This section states and derives the full APJN recurrence relations, including the direct attention contribution and the cross-positional Jacobian-correlation terms; see Results C.1 and C.2. The simplified main-text recurrence in Result 3.2 follows by neglecting the direct attention contribution and the cross-positional Jacobian-correlation terms.

C.1 Statement of the APJN Recurrences

C.1.1 Forward APJN Recurrence Relations

Result C.1 (Full forward APJN and cross-positional correlation recurrences). Under Assumption 2.1, the forward APJN satisfies

$$\mathcal{J}^{l+1,0} = \begin{cases} \left(1 + \frac{\sigma_{OV}^2}{n}\hat{q}^l\right)\mathcal{J}^{l,0} + \sigma_{OV}^2 \hat{p}^l \mathcal{K}^{l,0}, & \text{if } l \text{ is even (attn)}, \\ \left(1 + \frac{1}{2}\sigma_{21}^2 \hat{q}^{\,l}\right)\mathcal{J}^{l,0}, & \text{if } l \text{ is odd (MLP)}. \end{cases} \tag{37}$$

Here, $\mathcal{K}^{l,0}$ denotes the forward cross-positional Jacobian correlation,

$$\mathcal{K}^{l,0} = \frac{1}{n(n-1)d}\,\mathbb{E}_\theta\Big[\sum_{\substack{a\neq f,\ b,\\ \mu\nu}} \frac{\partial h_{a\mu}^l}{\partial h_{b\nu}^0}\frac{\partial h_{f\mu}^l}{\partial h_{b\nu}^0}\Big]. \tag{38}$$

This quantity satisfies

$$\mathcal{K}^{l+1,0} = \begin{cases} \left(1 + \sigma_{OV}^2 \hat{p}^l\right)\mathcal{K}^{l,0} + \frac{\sigma_{OV}^2}{n}\hat{q}^l \mathcal{J}^{l,0}, & \text{if } l \text{ is even (attn)}, \\ \left(1 + \sigma_{21}^2 \hat{\kappa}^l \hat{p}^l\right)\mathcal{K}^{l,0}, & \text{if } l \text{ is odd (MLP)}. \end{cases} \tag{39}$$

In Eqs. (37) and (39), $\hat{q}^l$ and $\hat{p}^l$ denote the covariance components obtained by propagating the covariance through either the derivative of the nonlinearity ϕ or the derivative of LayerNorm. For an elementwise nonlinearity ϕ,

$$\hat{\Sigma}_{st}^l = \mathbb{E}_{(h_1,h_2)\sim\mathcal{N}(0,\Sigma^l)}\big[\phi'(h_s)\phi'(h_t)\big], \qquad s,t \in \{1,2\}, \tag{40}$$

where $\hat{\Sigma}_{11}^l = \hat{\Sigma}_{22}^l = \hat{q}^l$, $\hat{\Sigma}_{12}^l = \hat{\Sigma}_{21}^l = \hat{p}^l$, and $\Sigma_{11}^l = \Sigma_{22}^l = q^l$, $\Sigma_{12}^l = \Sigma_{21}^l = p^l$. In the LayerNorm case, $\hat{q}^l = \hat{p}^l = 1/q^l$. The quantity $\hat{\kappa}^l$ arises from covariance propagation through the derivative of ReLU and is given by

$$\hat{\kappa}^l = \frac{1}{4} + \frac{1}{2\pi}\arcsin\left(\frac{\tilde{p}^l}{\tilde{q}^l}\right). \tag{41}$$

The recurrence relations in Eqs. (37) and (39) are supplemented with the initial conditions

$$\mathcal{J}^{0,0} = 1, \qquad \mathcal{K}^{0,0} = 0. \tag{42}$$

Because $\mathcal{K}^{0,0} = 0$, the cross-positional Jacobian correlation can be neglected when σ_{OV}^2/n is sufficiently small, producing the simplified APJN recurrence given in Eq. (12) in the main text.

C.1.2 Backward APJN Recurrence Relations

Result C.2 (Full backward APJN and cross-positional correlation recurrences). Under Assumption 2.1, the backward APJN satisfies

$$\mathcal{J}^{L,l} = \begin{cases} \left(1 + \frac{\sigma_{OV}^2}{n}\hat{q}^l\right)\mathcal{J}^{L,l+1} + \sigma_{OV}^2 \hat{q}^l \mathcal{K}^{L,l+1}, & \text{if } l \text{ is even (attn)}, \\ \left(1 + \frac{1}{2}\sigma_{21}^2 \hat{q}^{\,l}\right)\mathcal{J}^{L,l+1}, & \text{if } l \text{ is odd (MLP)}. \end{cases} \tag{43}$$

Here, $\mathcal{K}^{L,l}$ denotes the backward cross-positional Jacobian correlation,

$$\mathcal{K}^{L,l} = \frac{1}{n(n-1)d}\,\mathbb{E}_\theta\Big[\sum_{\substack{b\neq f,\ a,\\ \mu\nu}} \frac{\partial h_{a\mu}^L}{\partial h_{b\nu}^l}\frac{\partial h_{a\mu}^L}{\partial h_{f\nu}^l}\Big]. \tag{44}$$

This quantity satisfies

$$\mathcal{K}^{L,l} = \begin{cases} \left(1 + \sigma_{OV}^2 \hat{p}^l\right)\mathcal{K}^{L,l+1} + \frac{\sigma_{OV}^2}{n}\hat{p}^l \mathcal{J}^{L,l+1}, & \text{if } l \text{ is even (attn)}, \\ \left(1 + \sigma_{21}^2 \hat{\kappa}^l \hat{p}^l\right)\mathcal{K}^{L,l+1}, & \text{if } l \text{ is odd (MLP)}. \end{cases} \tag{45}$$

The quantities $\hat{q}^l$, $\hat{p}^l$, and $\hat{\kappa}^l$ are defined as in Result C.1. The recurrence relations in Eqs. (43) and (45) are supplemented by the initial conditions

$$\mathcal{J}^{L,L} = 1, \qquad \mathcal{K}^{L,L} = 0. \tag{46}$$

Analogously to the forward APJN case, when σ_{OV}^2/n is sufficiently small, the cross-positional Jacobian correlation can be neglected, giving the simplified backward APJN recurrence:

$$\mathcal{J}^{L,l} \approx \begin{cases} \mathcal{J}^{L,l+1}, & \text{if } l \text{ is even (attn)}, \\ \left(1 + \frac{1}{2}\sigma_{21}^2 \hat{q}^l\right) \mathcal{J}^{L,l+1}, & \text{if } l \text{ is odd (MLP)}. \end{cases} \tag{47}$$

In this case, the backward APJN is approximately the inverse of the forward APJN, up to the overall factor $\mathcal{J}^{L,0}$:

$$\mathcal{J}^{L,l} \approx \mathcal{J}^{L,0}/\mathcal{J}^{l,0}. \tag{48}$$

C.2 Derivation of APJN Recurrence Relations

In this section, we derive the APJN recurrence relations in Eqs. (37) and (39). To clearly distinguish positional and feature indices, we use Latin letters for the former and Greek letters for the latter. The μth component of the activation vector at position a in layer l is denoted by $h_{a\mu}^l$. We begin by separating the contribution of the last layer from the rest in the APJN:

$$\begin{aligned} \mathcal{J}^{l,0} &= \frac{1}{nd}\mathbb{E}_\theta \|J^{l,0}\|_F^2 = \frac{1}{nd}\mathbb{E}_\theta \sum_{ab,\mu\nu} \left(\frac{\partial h_{a\mu}^l}{\partial h_{b\nu}^0}\right)^2 = \frac{1}{nd}\mathbb{E}_\theta \sum_{\substack{abcd,\\ \mu\nu\xi\rho}} \frac{\partial h_{a\mu}^l}{\partial h_{c\xi}^{l-1}} \frac{\partial h_{a\mu}^l}{\partial h_{d\rho}^{l-1}} \frac{\partial h_{c\xi}^{l-1}}{\partial h_{b\nu}^0} \frac{\partial h_{d\rho}^{l-1}}{\partial h_{b\nu}^0} = \\ &= \frac{1}{nd} \sum_{\substack{abcd,\\ \mu\nu\xi\rho}} \mathbb{E}_\theta \left[\frac{\partial h_{a\mu}^l}{\partial h_{c\xi}^{l-1}} \frac{\partial h_{a\mu}^l}{\partial h_{d\rho}^{l-1}}\right] \mathbb{E}_\theta \left[\frac{\partial h_{c\xi}^{l-1}}{\partial h_{b\nu}^0} \frac{\partial h_{d\rho}^{l-1}}{\partial h_{b\nu}^0}\right]. \end{aligned} \tag{49}$$

In the last equality, we used the infinite-width factorization property: although the local Jacobian $\partial h^l/\partial h^{l-1}$ depends on h^{l-1}, the conditional second moment of its entries self-averages to a deterministic function of the layer-$(l-1)$ activation order parameters, such as the self-overlap q^{l-1} and the cross-positional overlap p^{l-1}. Analogously, one can separate the contribution of layer l from the rest in the cross-positional Jacobian correlation $\mathcal{K}^{l,0}$:

$$\mathcal{K}^{l,0} = \frac{1}{n^2 d}\mathbb{E}_\theta \sum_{\substack{a\neq f,\ b,\\ \mu\nu}} \frac{\partial h_{a\mu}^l}{\partial h_{b\nu}^0} \frac{\partial h_{f\mu}^l}{\partial h_{b\nu}^0} = \frac{1}{n^2 d} \sum_{\substack{a\neq f,\ bcd,\\ \mu\nu\xi\rho}} \mathbb{E}_\theta \left[\frac{\partial h_{a\mu}^l}{\partial h_{c\xi}^{l-1}} \frac{\partial h_{f\mu}^l}{\partial h_{d\rho}^{l-1}}\right] \mathbb{E}_\theta \left[\frac{\partial h_{c\xi}^{l-1}}{\partial h_{b\nu}^0} \frac{\partial h_{d\rho}^{l-1}}{\partial h_{b\nu}^0}\right]. \tag{50}$$

Here and in what follows, we approximate $n(n-1) \approx n^2$, assuming that the context size n is sufficiently large. Our goal is now to compute the first expectation in Eqs. (49) and (50), which will allow us to derive the recurrence relations in Eqs. (37) and (39).

C.2.1 $\mathcal{J}^{l,0}$ Recurrence for MLP

Consider the activation vector update in Eq. (23) due to the MLP. Let ψ denote a vector-valued function that is either an elementwise nonlinearity or $\mathrm{LayerNorm}$, i.e., $\psi \in \{\phi, \mathrm{LayerNorm}\}$. Since MLP layers do not mix activation vectors across positions, the APJN is a sum of identical contributions from each position. It therefore suffices to consider a single positional index:

$$\frac{\partial h_\mu^{l+1}}{\partial h_\xi^l} = \delta_{\mu\xi} + \sum_{\alpha\beta} (W_2)_{\mu\alpha}^l \mathrm{ReLU}'\left(\left(W_1 \tilde{\boldsymbol{h}}^l\right)_\alpha\right) (W_1)_{\alpha\beta}^l \frac{\partial \psi(\boldsymbol{h}^l)_\beta}{\partial h_\xi^l}, \tag{51}$$

where, for an elementwise nonlinearity ϕ,

$$\frac{\partial \psi(\boldsymbol{h}^l)_\beta}{\partial h_\xi^l} = \phi'(h_\beta^l)\delta_{\beta\xi}, \tag{52}$$

whereas, for $\mathrm{LayerNorm}$,

$$\frac{\partial \psi(\boldsymbol{h}^l)_\beta}{\partial h^l_\xi} = \frac{\sqrt{d}}{\|\boldsymbol{h}^l\|}\left(\delta_{\beta\xi} - \frac{h^l_\beta h^l_\xi}{\|\boldsymbol{h}^l\|^2}\right). \tag{53}$$

The second term in this expression is negligible in the large-d limit. Taking the expectation of a product of terms (51), as required for Eq. (49),

$$\mathbb{E}_\theta\left[\frac{\partial h^{l+1}_\mu}{\partial h^l_\xi}\frac{\partial h^{l+1}_\mu}{\partial h^l_\rho}\right] = \delta_{\mu\xi}\delta_{\mu\rho} + \delta_{\xi\rho}\frac{\sigma^2_{21}}{d}\left(\frac{1}{d_{\text{ff}}}\mathrm{ReLU}'\left(W^l_1\tilde{\boldsymbol{h}}^l\right)\cdot\mathrm{ReLU}'\left(W^l_1\tilde{\boldsymbol{h}}^l\right)\right)\mathbb{E}_\theta\left[\phi'\left(h^l_\rho\right)^2\right], \tag{54}$$

where the prime denotes the derivative. The dot product in parentheses is the activation variance propagated through ReLU' (i.e., the Heaviside step function), which does not depend on the input and equals $1/2$. Moreover, the expectation $\mathbb{E}_\theta\left[\phi'\left(h^l_\rho\right)^2\right]$ does not depend on the index ρ. For $\mathrm{LayerNorm}$, this expectation should effectively be replaced by $d/\|\boldsymbol{h}^l\|^2 = 1/q^l$, as follows from Eq. (53) in the large-d limit. Substituting the above expression into Eq. (49) and restoring the positional indices, we obtain

$$\mathcal{J}^{l+1,0} = \frac{1}{nd}\sum_{ab,\rho\nu}\left(1 + \frac{1}{2}\sigma^2_{21}\mathbb{E}_\theta\left[\phi'\left(h^l\right)^2\right]\right)\mathbb{E}_\theta\left[\left(\frac{\partial h^l_{a\rho}}{\partial h^0_{b\nu}}\right)^2\right] \tag{55}$$

Thus, for the MLP, we have shown that $\mathcal{J}^{l+1,0} = \chi^l_{\mathcal{J}}\mathcal{J}^{l,0}$, where

$$\chi^l_{\mathcal{J}} = 1 + \frac{\sigma^2_{21}}{2}\hat{q}^l, \tag{56}$$

and, for elementwise nonlinearities, we define

$$\hat{q}^l = \mathbb{E}_{h\sim\mathcal{N}(0,\,q^l)}\left[\phi'(h)^2\right]. \tag{57}$$

For $\mathrm{LayerNorm}$, we set $\hat{q}^l = 1/q^l$, following the discussion above.

C.2.2 $\mathcal{K}^{l,0}$ Recurrence for MLP

Restoring the positional indices in Eq. (54), we obtain

$$\begin{aligned}&\mathbb{E}_\theta\left[\frac{\partial h^{l+1}_{a\mu}}{\partial h^l_{c\xi}}\frac{\partial h^{l+1}_{f\mu}}{\partial h^l_{d\rho}}\right] = \\ &= \delta_{ac}\delta_{fd}\left[\delta_{\mu\xi}\delta_{\mu\rho} + \delta_{\xi\rho}\frac{\sigma^2_{21}}{d}\left(\frac{1}{d_{\text{ff}}}\mathrm{ReLU}'\left(W^l_1\tilde{\boldsymbol{h}}^l_c\right)\cdot\mathrm{ReLU}'\left(W^l_1\tilde{\boldsymbol{h}}^l_d\right)\right)\mathbb{E}_\theta\left[\phi'\left(h^l_{c\rho}\right)\phi'\left(h^l_{d\rho}\right)\right]\right].\end{aligned} \tag{58}$$

Substituting this expression into Eq. (50), we find that only the terms with $c \neq d$ contribute to the sum, which yields the recurrence relation

$$\mathcal{K}^{l+1,0} = \left(1 + \sigma^2_{21}\hat{\kappa}^l\hat{p}^l\right)\mathcal{K}^{l,0}, \tag{59}$$

where $\hat{p}^l$ is the off-diagonal counterpart of $\hat{q}^l$ in Eq. (57):

$$\hat{p}^l = \mathbb{E}_{(h_1,\,h_2)\sim\mathcal{N}(0,\,\Sigma^l)}\left[\phi'(h_1)\phi'(h_2)\right], \tag{60}$$

and $\hat{\kappa}^l$ arises from covariance propagation through the derivative of ReLU:

$$\hat{\kappa}^l = \mathbb{E}_{(h_1,\,h_2)\sim\mathcal{N}(0,\,\tilde{\Sigma}^l)}\left[\mathrm{ReLU}'(h_1)\mathrm{ReLU}'(h_2)\right] = \frac{1}{4} + \frac{1}{2\pi}\arcsin\left(\frac{\tilde{p}^l}{\tilde{q}^l}\right). \tag{61}$$

Here, Σ^l is the covariance matrix of the activations $\boldsymbol{h}^l$, whereas $\tilde{\Sigma}^l$ is the covariance matrix of the normalized activations $\tilde{\boldsymbol{h}}^l$. Note that for LayerNorm, the expectation $\mathbb{E}_\theta\big[\phi'\left(h^l_{c\rho}\right)\phi'\left(h^l_{d\rho}\right)\big]$ is replaced by $d/\|\boldsymbol{h}^l\|^2 = 1/q^l$, analogously to the derivation in the previous subsection. Therefore, for LayerNorm, we set $\hat{p}^l = 1/q^l$.

C.2.3 $\mathcal{J}^{l,0}$ and $\mathcal{K}^{l,0}$ Recurrence for Attention

Consider the activation vector update in Eq. (29) due to attention. As argued in Cowsik et al. [2024], the terms arising from differentiating the attention scores are subleading in $1/d$. Therefore, the computation is similar to the MLP case: we differentiate the nonlinearity and additionally approximate the product of attention scores as $1/n^2$. This gives

$$\mathbb{E}_\theta\left[\frac{\partial h^{l+1}_{a\mu}}{\partial h^{l}_{c\xi}}\frac{\partial h^{l+1}_{f\nu}}{\partial h^{l}_{d\rho}}\right] = \delta_{ac}\delta_{fd}\delta_{\mu\xi}\delta_{\nu\rho} + \frac{\sigma^2_{OV}}{d}\delta_{\mu\nu}\delta_{\xi\rho}A_{ac}A_{fd}\mathbb{E}_\theta\left[\phi'(h^\rho_c)\phi'(h^\rho_d)\right]. \tag{62}$$

Under the uniform-attention approximation, the second term does not depend on the indices a and f and therefore contributes equally to the recurrences for $\mathcal{J}^{l,0}$ and $\mathcal{K}^{l,0}$, as can be seen from Eqs. (49) and (50). Substituting this expression into Eq. (49) and splitting the sum into the contributions with $c = d$ and $c \neq d$, we obtain

$$\mathcal{J}^{l+1,0} = \mathcal{J}^{l,0} + \frac{\sigma^2_{OV}}{n}\hat{q}^l\frac{1}{nd}\sum_{c\xi,\, b\nu}\left(\frac{\partial h^l_{c\xi}}{\partial h^0_{b\nu}}\right)^2 + \sigma^2_{OV}\hat{p}^l\frac{1}{n^2d}\sum_{\substack{c\neq d,\, b,\\ \rho\nu}}\frac{\partial h^l_{c\rho}}{\partial h^0_{b\nu}}\frac{\partial h^l_{d\rho}}{\partial h^0_{b\nu}}. \tag{63}$$

Here, we recover the APJN recurrence in Eq. (37), as claimed. Analogously, substituting Eq. (62) into Eq. (50) and again splitting the sum into the contributions with $c = d$ and $c \neq d$, we obtain

$$\mathcal{K}^{l+1,0} = \mathcal{K}^{l,0} + \frac{\sigma^2_{OV}}{n}\hat{q}^l\frac{1}{nd}\sum_{c\xi,\, b\nu}\left(\frac{\partial h^l_{c\xi}}{\partial h^0_{b\nu}}\right)^2 + \sigma^2_{OV}\hat{p}^l\frac{1}{n^2d}\sum_{\substack{c\neq d,\, b,\\ \rho\nu}}\frac{\partial h^l_{c\rho}}{\partial h^0_{b\nu}}\frac{\partial h^l_{d\rho}}{\partial h^0_{b\nu}}. \tag{64}$$

This proves the recurrence for the cross-positional Jacobian correlation in Eq. (39).

Multi-head attention. In the case of multi-head attention, under the same assumptions as in Sec. B.3, Eq. (62) is modified as follows:

$$\mathbb{E}_\theta\left[\frac{\partial h^{l+1}_{a\mu}}{\partial h^{l}_{c\xi}}\frac{\partial h^{l+1}_{f\nu}}{\partial h^{l}_{d\rho}}\right] = \delta_{ac}\delta_{fd}\delta_{\mu\xi}\delta_{\nu\rho} + \frac{1}{H}\sum_{h=1}^{H}\frac{\sigma^2_{OV}}{d}\delta_{\mu\nu}\delta_{\xi\rho}A^h_{ac}A^h_{fd}\mathbb{E}_\theta\left[\phi'(h^\rho_c)\phi'(h^\rho_d)\right]. \tag{65}$$

Approximating the product of the attention scores by $1/n^2$, this equation reduces to Eq. (62), leading to the same APJN recurrence relations.

C.2.4 Derivation of Backward APJN Recurrence Relations

The derivation of the recurrence relations for the backward quantities, $\mathcal{J}^{L,l}$ and $\mathcal{K}^{L,l}$, in Eqs. (43) and (45) closely parallels the derivation given above for the forward quantities. We begin by separating the contribution of layer $l+1$ from the rest in the backward APJN:

$$\mathcal{J}^{L,l} = \frac{1}{nd}\mathbb{E}_\theta\sum_{\substack{ab,\\ \mu\nu}}\left(\frac{\partial h^L_{a\mu}}{\partial h^l_{b\nu}}\right)^2 = \frac{1}{nd}\sum_{\substack{abcd,\\ \mu\nu\xi\rho}}\mathbb{E}_\theta\left[\frac{\partial h^L_{a\mu}}{\partial h^{l+1}_{c\xi}}\frac{\partial h^L_{a\mu}}{\partial h^{l+1}_{d\rho}}\right]\mathbb{E}_\theta\left[\frac{\partial h^{l+1}_{c\xi}}{\partial h^l_{b\nu}}\frac{\partial h^{l+1}_{d\rho}}{\partial h^l_{b\nu}}\right]. \tag{66}$$

Analogously, we separate the contribution of layer $l+1$ from the rest in the cross-positional Jacobian correlation $\mathcal{K}^{L,l}$:

$$\mathcal{K}^{L,l} = \frac{1}{n^2d}\mathbb{E}_\theta\sum_{\substack{b\neq f,\, a,\\ \mu\nu}}\frac{\partial h^L_{a\mu}}{\partial h^l_{b\nu}}\frac{\partial h^L_{a\mu}}{\partial h^l_{f\nu}} = \frac{1}{n^2d}\sum_{\substack{b\neq f,\, acd,\\ \mu\nu\xi\rho}}\mathbb{E}_\theta\left[\frac{\partial h^L_{a\mu}}{\partial h^{l+1}_{c\xi}}\frac{\partial h^L_{a\mu}}{\partial h^{l+1}_{d\rho}}\right]\mathbb{E}_\theta\left[\frac{\partial h^{l+1}_{c\xi}}{\partial h^l_{b\nu}}\frac{\partial h^{l+1}_{d\rho}}{\partial h^l_{f\nu}}\right]. \tag{67}$$

Since the rest of the derivation is identical to that given above, we omit it here.

D Asymptotic Signal Propagation

From the covariance propagation recurrence relations, Eqs. (8) and (9), it follows that q^l and p^l increase with l. As a result, for $\mathrm{LayerNorm}$ or $\tanh$-like nonlinearities, $\tilde{q}^l$ and $\tilde{p}^l$ saturate, which in

turn leads to asymptotically linear growth of both q^l and p^l at large l. Therefore, the output of the bth transformer block has the following covariance components:

$$\begin{aligned} q^{2b} &\sim b\left(\frac{1}{2}\sigma_{21}^2\tilde{q}_\star + \sigma_{OV}^2\tilde{p}_\star\right), \\ p^{2b} &\sim b\left(\frac{1}{2}\sigma_{21}^2\kappa\left(\frac{\tilde{p}_\star}{\tilde{q}_\star}\right)\tilde{q}_\star + \sigma_{OV}^2\tilde{p}_\star\right), \end{aligned} \tag{68}$$

where $\tilde{q}_\star$ and $\tilde{p}_\star$ are the limiting values of $\tilde{q}^{2b}$ and $\tilde{p}^{2b}$ as $b \to \infty$, respectively. Assuming that σ_{OV}^2/n is sufficiently small, so that the cross-positional Jacobian correlation terms $\mathcal{K}^{l,0}$ and $\mathcal{K}^{L,l}$ are negligible, only the odd (MLP) layers contribute to the APJN:

$$\mathcal{J}^{2b,0} = \prod_{k=0}^{b-1}\chi_{\mathcal{J}}^{2k+1} = \prod_{k=0}^{b-1}\left(1 + \frac{\sigma_{21}^2}{2}\hat{q}^{2k+1}\right). \tag{69}$$

D.1 Asymptotic Behavior of APJN

This section derives Results 3.4 and 3.5.

D.1.1 LayerNorm

From Eq. (69) and using the asymptotic expression for q^{2k} in Eq. (68), we obtain:

$$\chi_{\mathcal{J}}^{2k+1} = 1 + \frac{\sigma_{21}^2}{2}\frac{1}{q^{2k+1}} \sim 1 + \frac{\frac{1}{2}\sigma_{21}^2}{k\left(\frac{1}{2}\sigma_{21}^2 + \sigma_{OV}^2\tilde{p}_\star\right)} = 1 + \frac{\zeta}{k}, \tag{70}$$

where ζ is defined in Eq. (17) in the main text. Thus, for large transformer block index b,

$$\log \mathcal{J}^{2b,0} \sim \sum_{k=K}^{b}\log\left(1 + \frac{\zeta}{k}\right) = \sum_{k=K}^{b}\left[\frac{\zeta}{k} + O(k^{-2})\right] \sim \zeta \log b, \tag{71}$$

where K is a sufficiently large integer whose choice does not affect the asymptotics. Therefore, the APJN exhibits power-law growth:

$$\mathcal{J}^{2b,0} \sim b^{\zeta}. \tag{72}$$

D.1.2 $\tanh$-like nonlinearities

First, for $\tanh$-like nonlinearities, Eq. (13) implies the following large-q^l asymptotic form for $\hat{q}^l$:

$$\hat{q}^l \approx \frac{C}{\sqrt{q^l}}, \qquad C = \frac{1}{\sqrt{2\pi}}\int_{\mathbb{R}}\phi'(h)^2\,dh. \tag{73}$$

Next, proceeding as in the previous subsection, we obtain the following asymptotic expression for $\chi_{\mathcal{J}}^{2k+1}$:

$$\chi_{\mathcal{J}}^{2k+1} = 1 + \frac{\sigma_{21}^2}{2}\frac{C}{\sqrt{q^{2k+1}}} \sim 1 + \frac{\frac{1}{2}\sigma_{21}^2 C}{\sqrt{k\left(\frac{1}{2}\sigma_{21}^2 + \sigma_{OV}^2\tilde{p}_\star\right)}} = 1 + \frac{1}{2\sqrt{k\lambda}}, \tag{74}$$

where λ is defined in Eq. (19) in the main text. Thus, for large transformer block index b,

$$\log \mathcal{J}^{2b,0} \sim \sum_{k=K}^{b}\log\left(1 + \frac{1}{2\sqrt{k\lambda}}\right) = \sum_{k=K}^{b}\left[\frac{1}{2\sqrt{k\lambda}} - \frac{1}{8k\lambda} + O(k^{-3/2})\right] \sim \sqrt{\frac{b}{\lambda}} - \frac{1}{8\lambda}\log b, \tag{75}$$

where K is a sufficiently large integer. Therefore, the APJN grows stretched-exponentially with b, with a power-law prefactor:

$$\mathcal{J}^{2b,0} \sim b^{-\frac{1}{8\lambda}}e^{\sqrt{b/\lambda}}. \tag{76}$$

This power-law prefactor modifies the asymptotic behavior of the backward APJN in Eq. (18) to

$$\mathscr{J}^{B,\,b} \sim \left(\frac{B}{b}\right)^{-\frac{1}{8\lambda}} e^{\left(\sqrt{B}-\sqrt{b}\right)/\sqrt{\lambda}}. \tag{77}$$

D.2 Limiting Value of $c = p/q$

This section derives the equation for the limiting value of the inter-token cosine similarity $c = p/q$ used in Result 3.3, analyzes the rate at which c approaches this limiting value, and shows that $c_\star = 1$ is unstable for $\tanh$-like nonlinearities.

D.2.1 Equation for the Limiting Value of $c = p/q$

Let us denote by $c^b = p^{2b}/q^{2b}$ the ratio of the covariance components, which is equal to the cosine similarity between activation vectors at different positions in the output of transformer block b. For large b, the increments of q^{2b} and p^{2b} per transformer block are approximately

$$\begin{aligned} \Delta q(c) &= \frac{1}{2}\sigma_{21}^2 + \sigma_{OV}^2 \tilde{p}(c), \\ \Delta p(c) &= \frac{1}{2}\sigma_{21}^2 \kappa(\tilde{p}(c)) + \sigma_{OV}^2 \tilde{p}(c), \end{aligned} \tag{78}$$

evaluated at $c = c^b$. Here, $\tilde{p}(c)$ denotes the asymptotic form of $\tilde{p}$ for large q and p, as follows from Eq. (10):

$$\tilde{p}(c) = \frac{2}{\pi} \arcsin c \tag{79}$$

for an elementwise $\tanh$-like nonlinearity, and

$$\tilde{p}(c) = c \tag{80}$$

for $\mathrm{LayerNorm}$. Therefore, the asymptotic dynamics of c^b, induced by those of p^{2b} and q^{2b}, are given by

$$c^{b+1} = c^b + \frac{\Delta p(c^b)}{q^{2b}} - c^b \frac{\Delta q(c^b)}{q^{2b}} + o\left(\frac{1}{q^{2b}}\right). \tag{81}$$

For convenience, let us define

$$g(c) = \Delta p(c) - c\Delta q(c). \tag{82}$$

The fixed points of the asymptotic dynamics are given by the zeros of $g(c)$. Thus, any fixed point $c_\star$ satisfies $g(c_\star) = 0$, or equivalently, $c_\star = \Delta p(c_\star)/\Delta q(c_\star)$, i.e.,

$$c_\star = \frac{\frac{1}{2}\sigma_{21}^2 \kappa(\tilde{p}(c_\star)) + \sigma_{OV}^2 \tilde{p}(c_\star)}{\frac{1}{2}\sigma_{21}^2 + \sigma_{OV}^2 \tilde{p}(c_\star)}. \tag{83}$$

D.2.2 Rate of Convergence of c to the Limiting Value

At a fixed point, $g(c_\star) = 0$, so we expand $g(c)$ in Eq. (81) to first order in $c - c_\star$. This yields

$$\Delta c \sim \frac{g'(c_\star)(c - c_\star)}{b\Delta q(c_\star)}. \tag{84}$$

From this equation, we obtain power-law convergence to the fixed point,

$$|c - c_\star| = b^{-\mu}, \tag{85}$$

with exponent

$$\mu = -\frac{g'(c_\star)}{\Delta q(c_\star)}. \tag{86}$$

The derivative $g'(c)$ follows from the definition of $g(c)$:

$$g'(c) = \tilde{p}'(c)\left[\frac{1}{2}\sigma_{21}^2 \kappa'(\tilde{p}(c)) + (1-c)\sigma_{OV}^2\right] - \frac{1}{2}\sigma_{21}^2 - \sigma_{OV}^2 \tilde{p}(c). \tag{87}$$

For $\mathrm{LayerNorm}$, the exponent simplifies to $\mu = \sigma_{OV}^2 / \left(\sigma_{OV}^2 + \frac{1}{2}\sigma_{21}^2\right)$.

D.2.3 Instability of the Asymptotic Fixed Point $c_\star = 1$ for $\tanh$-Like Nonlinearities

From Eq. (81), a transformer block increases the alignment between activation vectors, as measured by c, if $g(c) > 0$, and decreases it if $g(c) < 0$. Let us show that for $c = 1 - \varepsilon$ with sufficiently small ε, $g(c)$ is negative for $\tanh$-like nonlinearities. For $\tanh$-like nonlinearities, from Eq. (79), $\tilde{p}(1-\varepsilon) = 1 - \frac{2\sqrt{2}}{\pi}\sqrt{\varepsilon} + O(\varepsilon^{3/2})$. Additionally, from Eq. (28), for small x, $\kappa(1-x) = 1 - x + o(x)$. Thus, from Eqs. (82) and (78), a straightforward computation yields:

$$g(1-\varepsilon) = -\frac{\sqrt{2}}{\pi}\sigma_{21}^2\sqrt{\varepsilon} + o(\sqrt{\varepsilon}), \tag{88}$$

which is negative for sufficiently small ε. This implies that c decreases according to Eq. (81).

E Explicit Expressions for erf-Nonlinearity

For $\phi(h) = \mathrm{erf}(\alpha h)$, the covariance components propagated through ϕ, denoted by $\tilde{q}^l$ and $\tilde{p}^l$ and defined in Eq. (10), can be computed analytically as

$$\tilde{q}^l = \frac{2}{\pi}\arcsin\left(\frac{2\alpha^2 q^l}{1+2\alpha^2 q^l}\right), \qquad \tilde{p}^l = \frac{2}{\pi}\arcsin\left(\frac{2\alpha^2 p^l}{1+2\alpha^2 q^l}\right). \tag{89}$$

Similarly, the covariance components propagated through ϕ', denoted by $\hat{q}^l$ and $\hat{p}^l$ and defined in Eq. (40), are given by

$$\hat{q}^l = \frac{4\alpha^2}{\pi\sqrt{1+4\alpha^2 q^l}}, \qquad \hat{p}^l = \frac{4\alpha^2}{\pi\sqrt{(1+2\alpha^2 q^l)^2 - 4\alpha^4 (p^l)^2}}. \tag{90}$$

In particular, these expressions explicitly show why smaller values of α in Derf lead to slower growth of both q^l, according to Eqs. (8) and (9), and the APJN, according to Eqs. (11) and (12).

F Role of the Final Normalization Layer

Pre-LN transformers typically use a final LayerNorm, whereas DyT/Derf models use the corresponding $\tanh$-like nonlinearity at the output [Zhu et al., 2025, Chen et al., 2025]. This final normalization layer contributes an additional factor $\hat{Q}^B$ to the backward APJN: specifically, the APJN from block b to the output of the final normalization layer satisfies

$$\mathscr{J}^{\mathrm{out},b} = \hat{Q}^B \mathscr{J}^{B,b}. \tag{91}$$

For LayerNorm,

$$\hat{Q}^B = 1/Q^B, \tag{92}$$

whereas for a final $\tanh$-like nonlinearity parameterized by α, $\hat{Q}^B$ has the same form as in Eq. (57):

$$\hat{Q}^B = \mathbb{E}_{h\sim\mathcal{N}(0,\,Q^B)}\left[\phi'(h)^2\right], \tag{93}$$

In particular, for a final erf nonlinearity parameterized by α,

$$\hat{Q}^B = \frac{4\alpha^2}{\pi\sqrt{1+4\alpha^2 Q^B}}. \tag{94}$$

This factor is governed by the same mechanism as the layer-to-layer APJN growth discussed in the main text. At fixed α in DyT/Derf, increasing model depth or weight scale increases the last-block variance Q^B; once Q^B is sufficiently large, $\hat{Q}^B$ is larger for DyT/Derf than for LayerNorm. At fixed model depth, however, decreasing α suppresses $\hat{Q}^B$ and can make it smaller than in pre-LN. Thus, replacing the final LayerNorm with DyT/Derf can further increase backward gradient amplification in deep networks, though sufficiently small α can weaken or reverse this effect.

G Supplementary Results on Signal Propagation

Activation covariance components Q^b and P^b, and their ratio P^b/Q^b. Fig. 6 shows these quantities as computed by the theory and measured in a ViT model on a permutation-symmetric input token configuration with $(q^0,\, p^0) = (1.0,\, 0.2)$, averaged over 16 weight initializations.

Backward-APJN comparison examples. Fig. 7 shows example comparisons between measured and theoretical backward-APJN curves for two randomly selected samples from the experiment shown in Fig. 1(b). As discussed in the main text, the case $(\sigma_{21}, \sigma_{OV}) = (0.6,\, 1.2)$ shows worse agreement, likely because larger attention weights make the APJNs more sensitive to deviations from the permutation-symmetry assumption of the theory.

Backward APJN: ViT with CIFAR-100 inputs vs. theory at additional values of $(\sigma_{21}, \sigma_{OV})$. Fig. 8 repeats the experiments in Fig. 1 from the main text for backward APJNs at additional, smaller values of $(\sigma_{21}, \sigma_{OV})$. Note that smaller weight variances cause the activations to preserve a complex distribution of dot products up to deeper transformer blocks. As a result, the theoretical assumptions become applicable only in deeper layers, and one should not expect the theory to accurately capture the APJN curve shape in this regime, as illustrated, for example, by the case $(\sigma_{21}, \sigma_{OV}) = (0.15,\, 0.07)$ in Fig. 8(b).

Forward APJNs. Fig. 9 repeats the experiments in Fig. 1 from the main text for forward APJNs.

Fig. 9(a) compares the forward APJNs $\mathscr{J}^{b,0}$ measured in ViT model variants on several synthetic permutation-symmetric inputs with $(q^0, p^0) = (1.0,\ 0.2)$ against the APJNs predicted by the theoretical recurrence relations in Eqs. (8), (9), and (12).

Fig. 9(b) shows per-sample GMFE scores for forward APJN values, comparing theoretical predictions with measurements on randomly selected CIFAR-100 samples. For this comparison, we shift the reference block to $b_0 = 8$ and compare the measured $\mathscr{J}^{b,b_0}$ with theory. We do so because the theory does not accurately capture the overall scale of the measured pre-LN forward APJN values when the reference is taken at the input; this scale is determined by the first few transformer blocks. This discrepancy may arise because forward APJNs involve differentiation with respect to the inputs, whose statistics differ substantially from those assumed in the theory. By contrast, activations in later blocks are expected to more closely satisfy the theoretical assumptions: after a few transformer blocks, the variances of the self-dot products and cross-dot products among activation vectors become smaller relative to their mean values, as illustrated in Figs. 13 and 15.

Comparison of backward APJNs on CIFAR-100 inputs for Derf with $\alpha = 0.3$ and $\alpha = 1.9$ against the theory. Fig. 10 is analogous to Fig. 2 in the main text for different values of α. The asymptotic expression for the APJN is less accurate for models with smaller α, even in networks with more than one hundred transformer blocks. This is because the derivation of the asymptotic APJN assumes that l is large enough for $\tilde{q}^l$ and $\tilde{p}^l$ in Eqs. (8) and (9) to approach their asymptotic values. However, the layer at which this occurs depends on the nonlinearity parameter α and may lie beyond the network depth. Indeed, as discussed in the main text, smaller α leads to slower growth of q^l with l. Consequently, the asymptotic APJN expressions in Eqs. (16) and (18) become valid only in deeper layers for smaller α and may therefore not apply in finite-depth networks. Nevertheless, the full APJN theory remains accurate at small α, as illustrated in Fig. 10(a) for $\alpha = 0.3$ in Appendix G.

Ratio of the cross-positional Jacobian-correlation term to the APJN in the full recurrence relations of Eqs. (37) and (43); shown in Fig. 11 for several values of $(\sigma_{21}, \sigma_{OV})$. The cross-positional Jacobian-correlation terms, $\mathscr{K}^{B,b}$ and $\mathscr{K}^{b,0}$, can become non-negligible for larger values of σ_{OV}, as illustrated by the case $(\sigma_{21}, \sigma_{OV}) = (0.6, 1.2)$. The quantity $\mathscr{K}^{b',b}$ is defined analogously to the other block-level quantities in the text: $\mathscr{K}^{b',b} = \mathcal{K}^{2b',2b}$.

Input statistics and the evolution of activation statistics across transformer blocks. Fig. 12 shows the input statistics for four random CIFAR-100 samples, namely the normalized activation-vector norms and cross-positional dot products. These statistics clearly violate the simplifying assumption of permutation invariance among activation vectors used in the theoretical derivation in the main text.

Figs. 13–16 show how the activation statistics evolve across transformer blocks in the pre-LN ViT model and the Derf ViT model with $\alpha = 1$, based on 1000 CIFAR-100 input samples. With a slight abuse of notation, $\bar{Q}$ and $\bar{P}$ denote the averages over positions of the normalized self-dot products and cross-positional dot products of activation vectors, respectively, at a given transformer

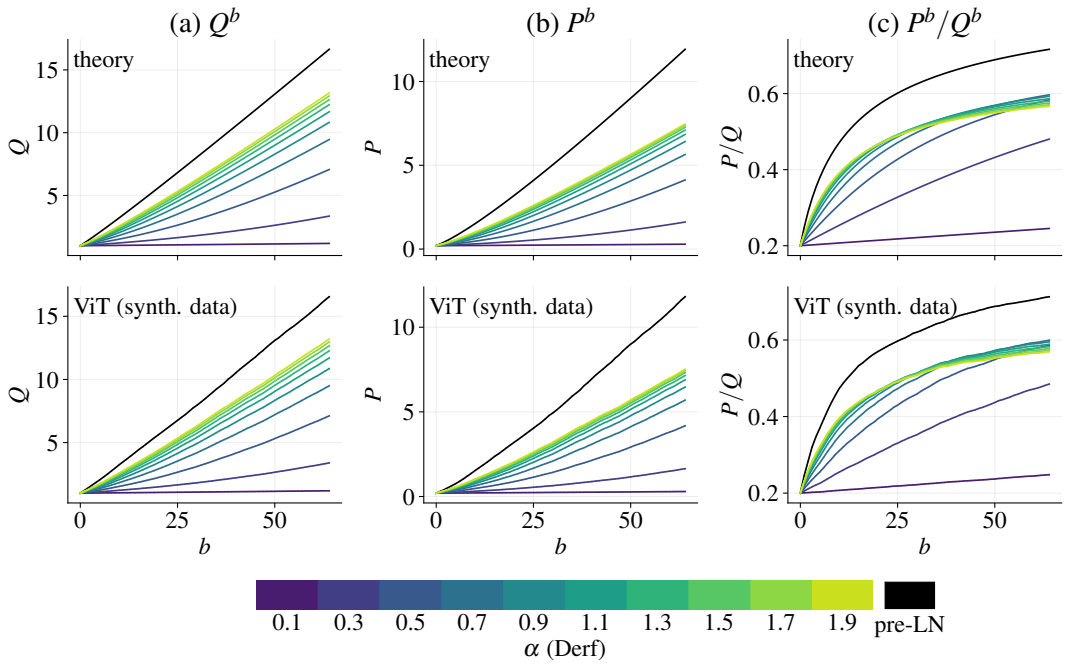

Figure 6: The activation covariance components **(a)** Q^b and **(b)** P^b, as well as their ratio **(c)** P^b/Q^b. **Upper row**: theoretical values predicted by the recurrence relations (8) and (9); **lower row**: values measured in a ViT model for a permutation-symmetric input token configuration with $(q^0, p^0) = (1.0,\ 0.2)$.

block for a fixed input sample. δQ and δP denote the corresponding standard deviations; these are therefore within-sample quantities. In turn, $\langle \bar{Q} \rangle$ and $\langle \bar{P} \rangle$ denote averages across input samples, with corresponding standard deviations $\delta \bar{Q}$ and $\delta \bar{P}$.

Specifically, Figs. 13 and 15 show how the distribution of the within-sample relative dot-product fluctuations, $\delta Q/\bar{Q}$ and $\delta P/\bar{P}$, evolves across transformer blocks. For both self-dot products and cross-positional dot products, these fluctuations decrease with depth. This suggests that the theoretical assumptions become more accurate for activations in deeper layers. In particular, the theory provides a better description of the empirically measured APJN $\mathscr{J}^{b',b}$ at larger values of b and b'.

Additionally, Figs. 14 and 16 show how the sample-to-sample relative fluctuations of these mean quantities, $\delta \bar{Q}/\langle \bar{Q} \rangle$ and $\delta \bar{P}/\langle \bar{P} \rangle$, evolve across transformer blocks. For both self-dot products and cross-positional dot products, these relative fluctuations decrease with depth. This suggests that the dependence on the specific input statistics gradually weakens.

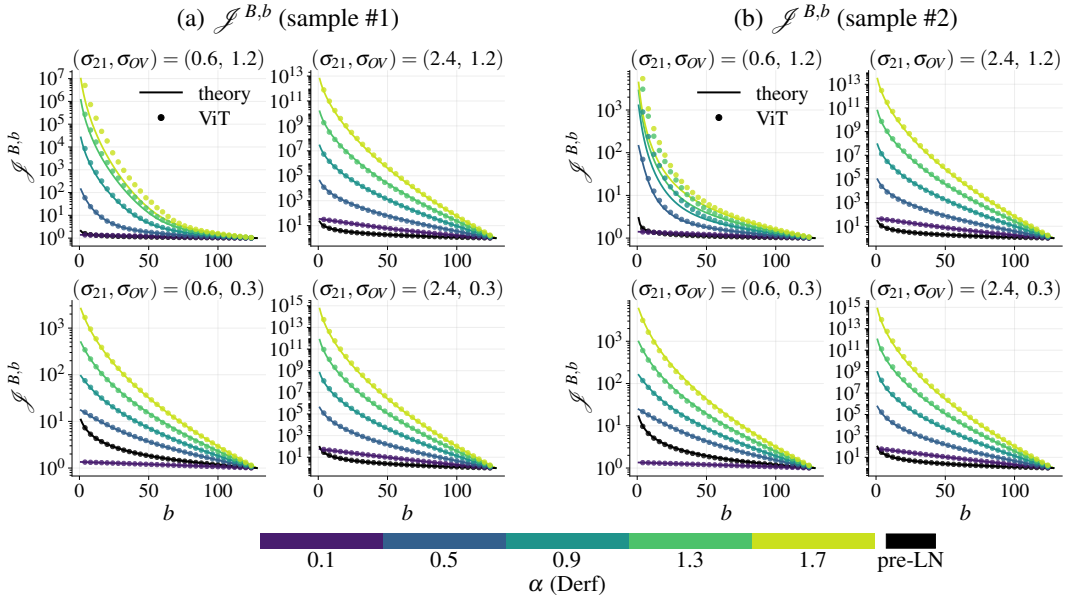

Figure 7: Backward APJN $\mathscr{J}^{B,b}$: ViT on CIFAR-100 inputs versus theory comparison. **(a)** and **(b)** show two different CIFAR-100 input samples. Circles denote APJN values measured in ViT models (the pre-LN variant and Derf variants with multiple values of α). Solid lines denote the backward APJN values predicted by the theoretical model in Eqs. (8)–(12).

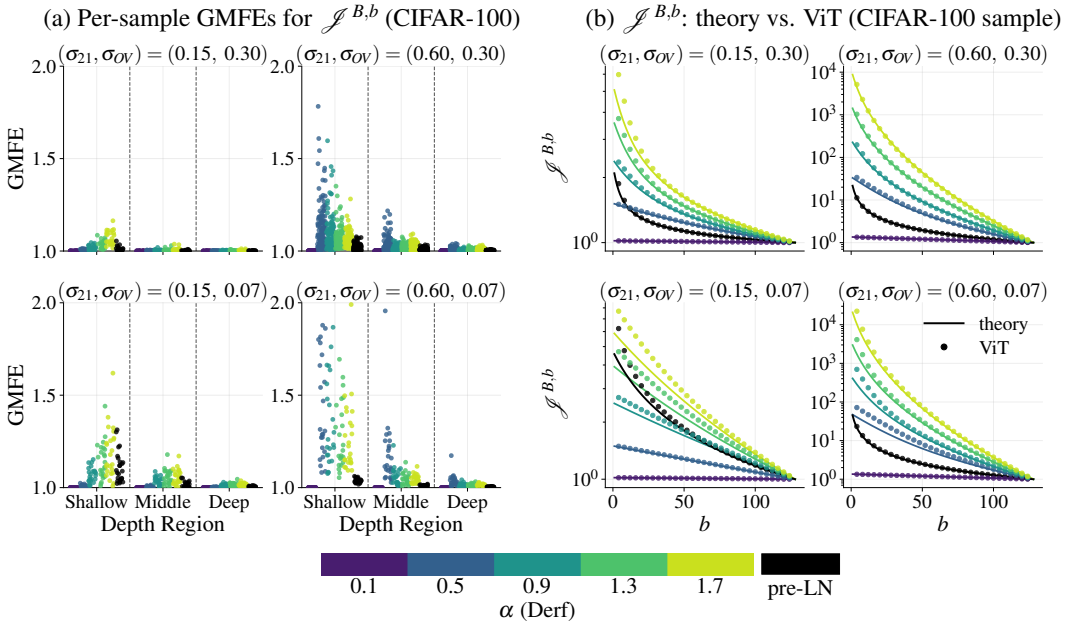

Figure 8: Backward APJN: ViT with CIFAR-100 inputs vs. theory at additional values of $(\sigma_{21}, \sigma_{OV})$. **(a)** Per-sample GMFE values between the theoretical values of $\mathscr{J}^{B,b}$ and those measured in a ViT with $B = 128$ blocks on CIFAR-100 inputs, evaluated in early, middle, and deep layers. **(b)** Backward APJN $\mathscr{J}^{B,b}$ for a ViT on a random CIFAR-100 input, compared with theory. Circles denote APJN values measured in ViT models, and solid lines denote the backward APJN values predicted by the theoretical model in Eqs. (8)–(12).

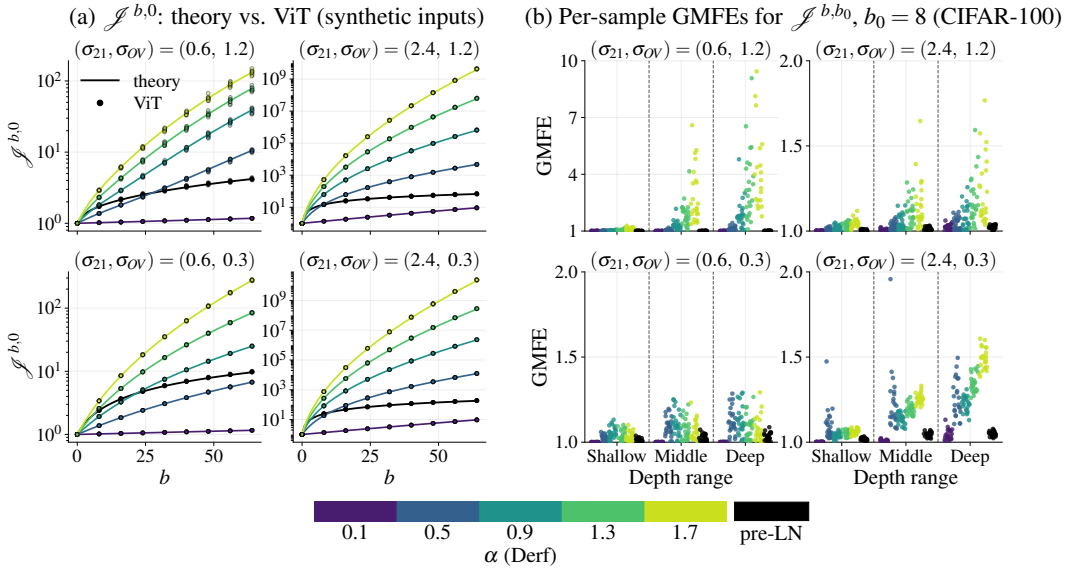

Figure 9: **(a)** Forward APJN $\mathcal{J}^{b,0}$: theory vs. ViT on synthetic permutation-symmetric inputs. Circles denote forward APJN values measured in the ViT model on several permutation-symmetric inputs, for both the pre-LN variant and Derf variants with several values of α. Solid lines denote the forward APJN values predicted by the theoretical model in Eqs. (8)–(12). **(b)** Per-sample GMFE values between the theoretical values of $\mathcal{J}^{b,b_0}$, $b_0 = 8$ and those measured in a ViT with $B = 128$ blocks on 24 CIFAR-100 inputs, in early, middle, and deep layers.

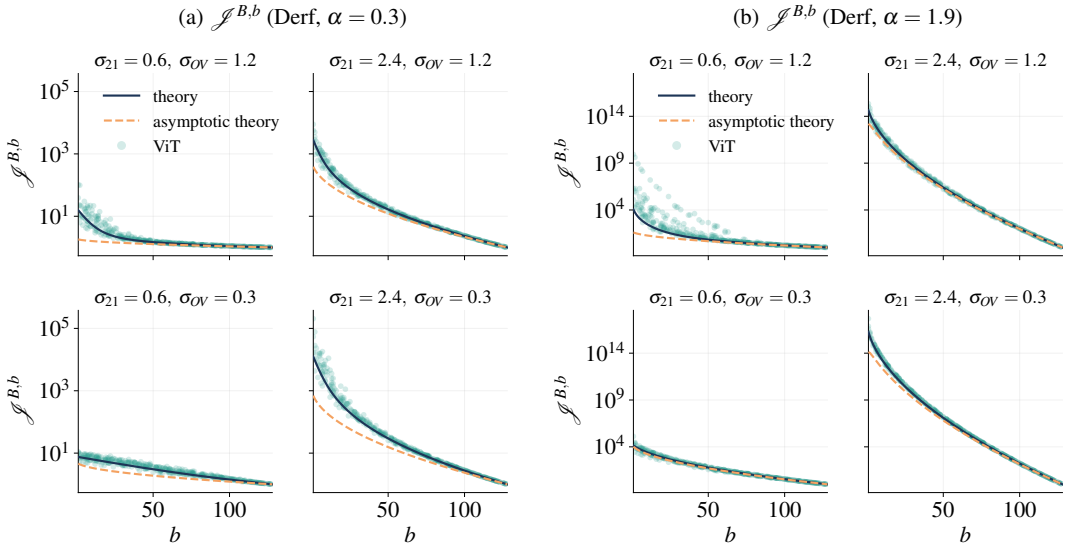

Figure 10: Backward APJN $\mathcal{J}^{B,b}$ for ViT with Derf, together with the full-theory prediction and the asymptotic prediction from Eq. (18). **(a)** $\alpha = 0.3$; **(b)** $\alpha = 1.9$. Teal circles indicate backward APJN values measured in ViT on 100 random training samples from CIFAR-100. Black solid lines show the full-theory predictions obtained from the recurrence relations (8), (9), and (12) with $(q^0, p^0) = (0.5, 0.25)$, while orange dashed lines show the large-b asymptotic predictions.

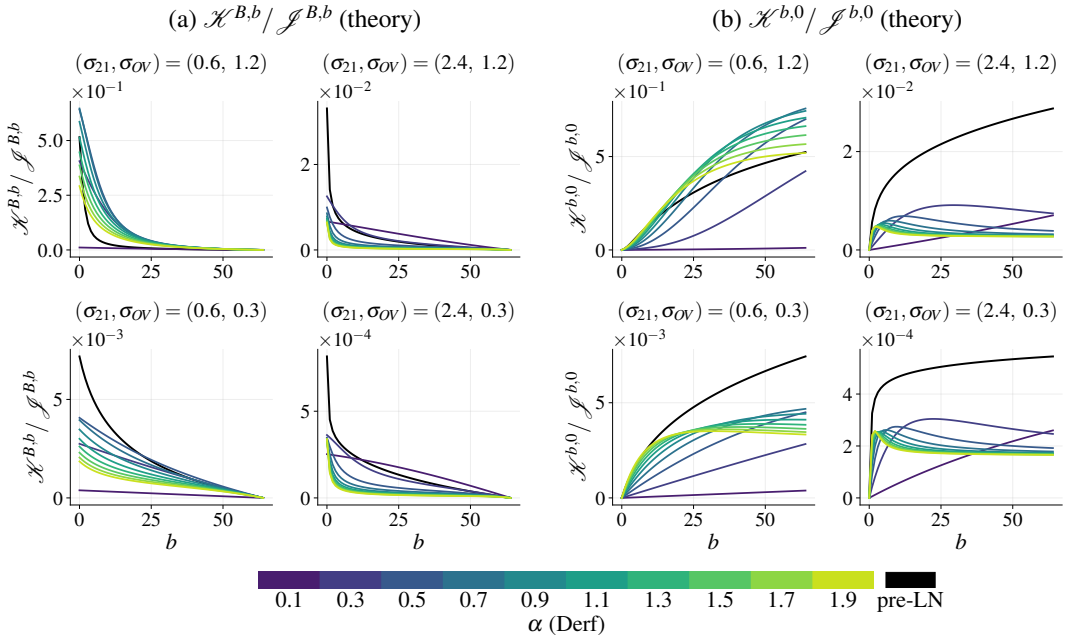

Figure 11: Ratio of the cross-positional Jacobian correlation to the APJN, computed from theory for multiple values of $(\sigma_{21}, \sigma_{OV})$. **(a)** Ratio $\mathscr{K}^{B,b}/\mathscr{J}^{B,b}$. **(b)** Ratio $\mathscr{K}^{b,0}/\mathscr{J}^{b,0}$. The cross-positional Jacobian correlation becomes non-negligible for larger values of σ_{OV}.

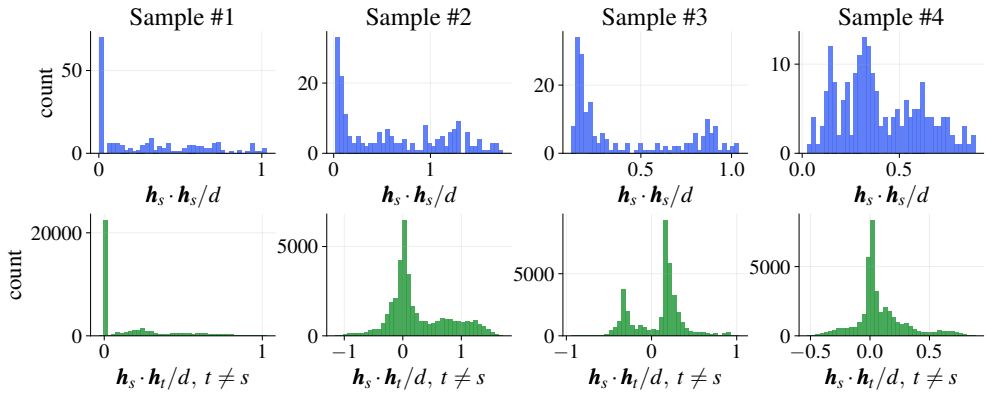

Figure 12: Histograms of normalized activation vector norms and normalized dot products between activation vectors at different positions for four CIFAR-100 training samples, measured after random augmentation, feature extraction, and positional encoding, immediately before the first transformer block.

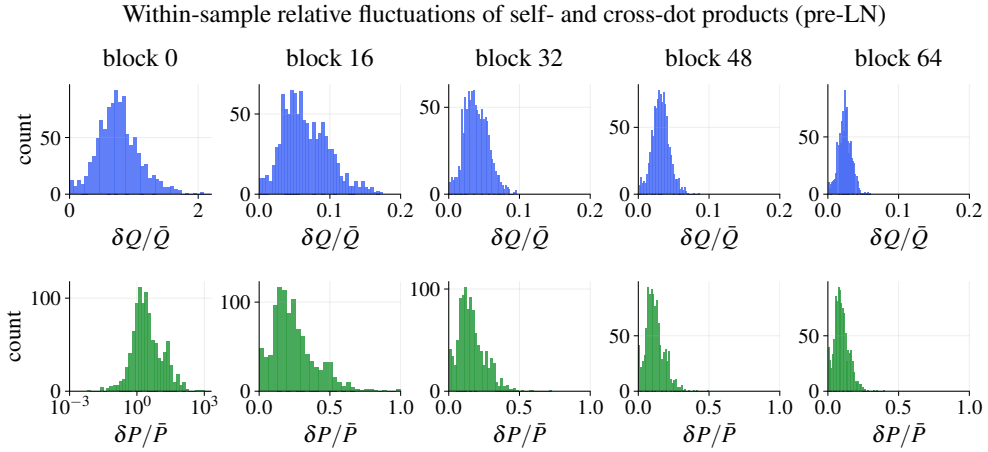

Figure 13: Pre-LN: Histograms of within-sample relative fluctuations of self-dot products (**upper row**) and cross-positional dot products between activation vectors (**lower row**) at multiple transformer blocks.

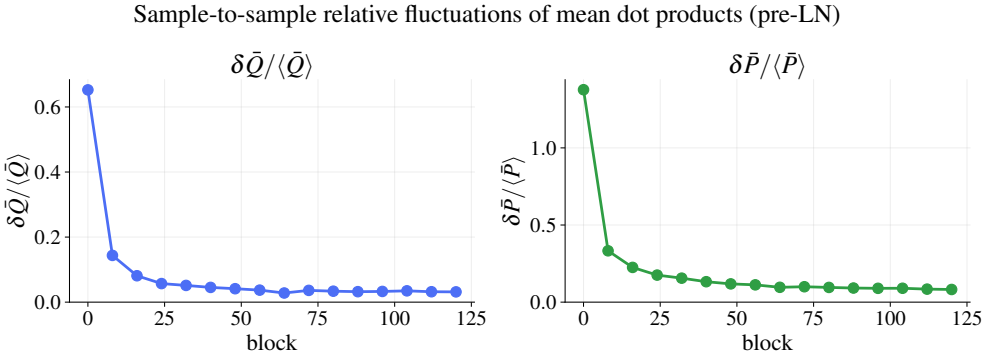

Figure 14: Pre-LN: Sample-to-sample relative fluctuations of mean self-dot products (**left**) and mean cross-positional dot products between activation vectors (**right**) across transformer blocks.

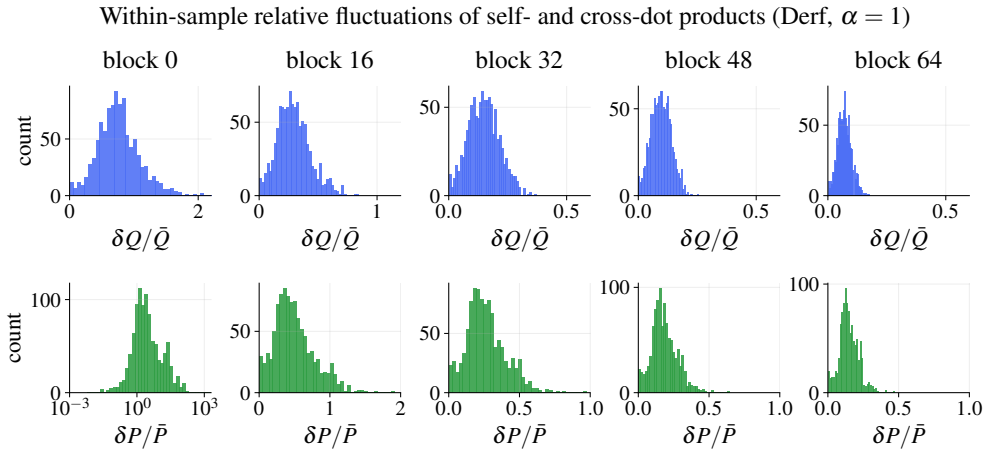

Figure 15: Derf, $\alpha = 1$: Histograms of within-sample relative fluctuations of self-dot products (**upper row**) and cross-positional dot products between activation vectors (**lower row**) at multiple transformer blocks.

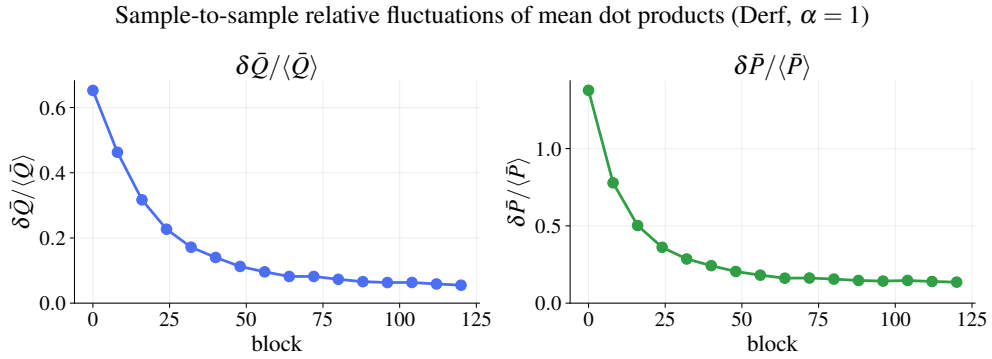

Figure 16: Derf, $\alpha = 1$: Sample-to-sample relative fluctuations of mean self-dot products (**left**) and mean cross-positional dot products between activation vectors (**right**) across transformer blocks.

H Supplementary Results on Training Stability Experiments

H.1 Gradient Amplification in ViT: Measurements and Analysis

Fig. 17 shows the gradient amplification in the ViT model from the last transformer block to block b, $\mathbb{E}\|g_b\|^2/\mathbb{E}\|g_B\|^2$ (upper row), and from the final normalization layer to block b, $\mathbb{E}\|g_b\|^2/\mathbb{E}\|g_{\text{out}}\|^2$ (lower row), where g_b denotes the gradient with respect to the output of the bth transformer block, and g_{out} denotes the gradient with respect to the output of the final normalization layer. The expectations are estimated using a batch of 32 CIFAR-100 training examples.

Fig. 17(a) shows that, for $B = 12$, the gradient amplification of pre-LN approximately matches that of Derf with $\alpha \approx 0.9$ in the deeper layers, while its overall gradient amplification is closer to that of Derf with $\alpha \approx 1.3$. Fig. 17(b) shows that, at greater depth ($B = 36$), the pre-LN baseline matches the gradient amplification of Derf variants with smaller values of α, namely $\alpha \approx 0.5$–0.7, whereas Derf variants with larger values of α exhibit stronger gradient amplification. Both statements hold for gradient amplification computed either from the last transformer block or from the output of the final normalization layer. This behavior qualitatively agrees with our theory and may explain why Derf variants with larger values of α become unstable at smaller depth as model depth increases.

However, we also observe that equal gradient amplification does not imply equal convergence rate. Even when Derf is matched to pre-LN in terms of gradient amplification at a given depth, it still tends to converge more slowly. For example, at depth 12, Derf with $\alpha = 0.8$ converges more slowly than pre-LN in Fig. 4(a), whereas $\alpha = 0.6$ yields a convergence rate closer to that of pre-LN, and $\alpha = 0.4$ converges faster. In Fig. 25, we show that tuning the initialization parameter γ in pre-LN can partially close this gap.

Fig. 17(c) shows that increasing the weight scale leads to significantly stronger gradient amplification in the Derf variants than in pre-LN.

H.2 Supplementary Experiments

Training loss and test performance curves. Figs. 18 and 19 show the training loss and test performance curves for the experiments presented in Figs. 4(a) and 4(b), respectively.

Training stability comparison between Derf and pre-LN under varying depth with other hyperparameter settings. Fig. 20 repeats the training stability comparison between Derf and pre-LN shown in Fig. 4(a), using a lower learning rate and a larger batch size than in the main experiment. Fig. 21 repeats this experiment with a longer warmup and without weight decay.

Lowering the learning rate or increasing the batch size shifts the Derf instability boundary toward larger α and greater depth, while increasing the number of warmup epochs improves performance near the boundary without substantially shifting it. By contrast, weight decay has little effect on training stability in this setting.

Warmup and learning-rate sweep experiments for Derf in other settings. Fig. 22 repeats the experiments in Figs. 5(a) and (b) without weight decay in Adam, showing that removing weight decay does not significantly affect training stability. Fig. 23 repeats the experiments in Figs. 5(a) and (b) with a non-trainable parameter α fixed at its initialization value. Comparing with Fig. 5 suggests that making α untrainable worsens training convergence.

Depth sweep for DyT. We repeat the depth sweep of Fig. 4(a) for DyT models. Fig. 24 shows qualitatively similar behavior to Derf: larger initial values of α lead to training instability at smaller depths.

Small-α sweep experiment in other settings. Fig. 26 repeats the experiment from Fig. 5(c) under several alternative settings: smaller and larger learning rates, no weight decay, and no replacement of the final LayerNorm with Derf. Across these settings, the results suggest that choosing α too small in Derf can slow convergence.

Pre-LN γ sweep vs. Derf α sweep. Fig. 25 compares pre-LN models with different values of the initialization parameter γ against Derf models with different values of α. Here, $\boldsymbol{\gamma}$ is defined as $\boldsymbol{\gamma} = \gamma\mathbf{1}$. Note that, in this experiment, the inputs in the pre-LN variant are additionally normalized with LayerNorm before the first transformer block, unlike in the other experiments. As a result, the

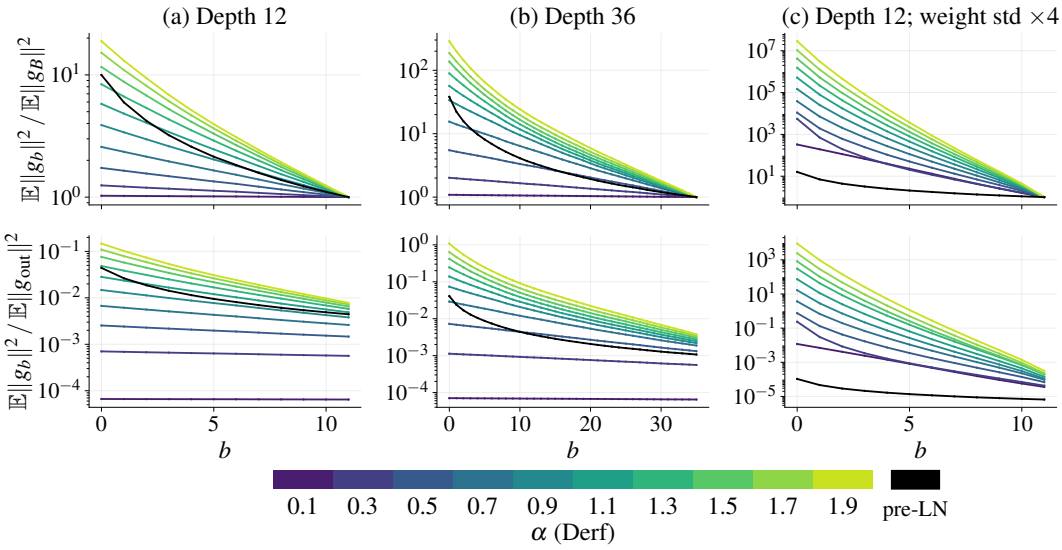

Figure 17: Gradient amplification in the ViT model from **(upper row)** the last transformer block to earlier blocks, and from **(lower row)** the final normalization layer to earlier blocks: Derf vs. pre-LN. **(a)** Depth $B = 12$. **(b)** Depth $B = 36$. As the network depth increases, the gradient amplification in pre-LN matches that of Derf models with smaller α. In addition, smaller α results in smaller overall gradient magnitudes because of gradient propagation through the final normalization layer. **(c)** Depth $B = 12$, with the MLP and attention weight standard deviations increased by a factor of 4, leading to substantially stronger gradient amplification in Derf than in pre-LN.

training curves and test accuracy for the $\gamma = 1$ variant may differ from those in the other experiments, such as those shown in Figs. 18(a) and 19(a).

H.3 Preliminary Experiments on Language Modeling

As a preliminary empirical investigation, we conducted additional next-token-prediction experiments on the TinyStories dataset. After one training epoch, a 12-block pre-LN model outperformed both a 12-block DyT model and a 36-block DyT model, suggesting that DyT models may also benefit from further hyperparameter tuning in this setting.

However, these experiments did not identify a clear training-instability boundary as a function of depth and α, comparable to that observed for ViTs trained on CIFAR-100 in Fig. 4(a). In particular, the 36-block DyT model, initialized with a larger value of α, exhibited better convergence and performance than the 12-block DyT model initialized with a smaller value of α.

These preliminary results leave open whether the same instability pattern occurs in causal language models. Broader sweeps over learning rates, model depths, and α, together with an extension of the theory to causal attention, could help clarify this question.

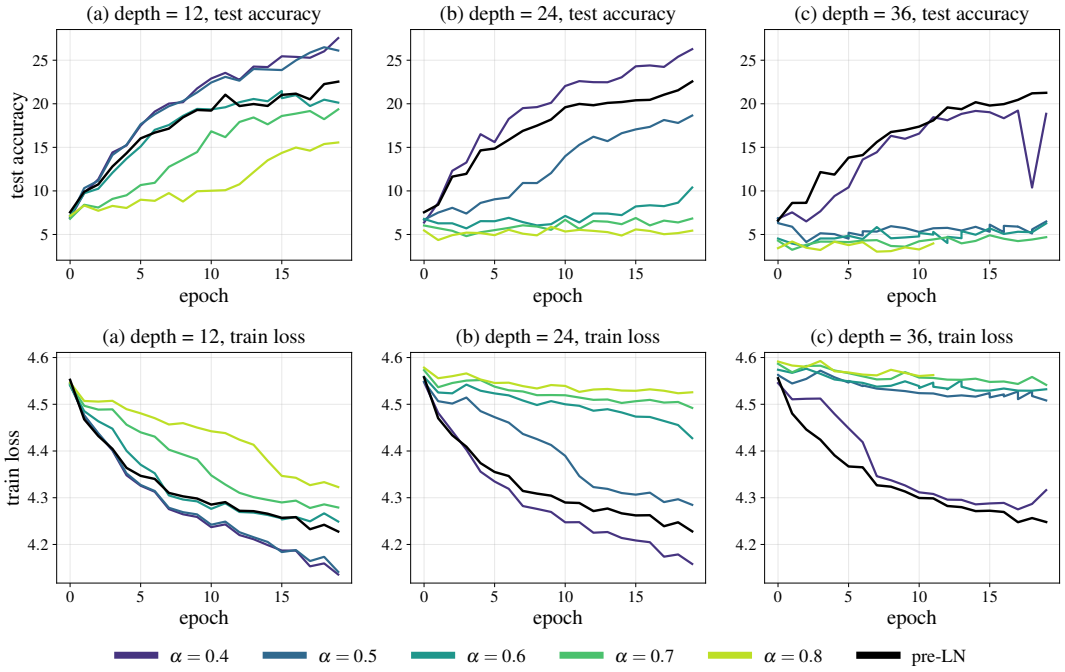

Figure 18: **Upper:** test performance curves; **lower:** training curves for the training-stability comparison between Derf and pre-LN when varying depth, shown in Fig. 4(a).

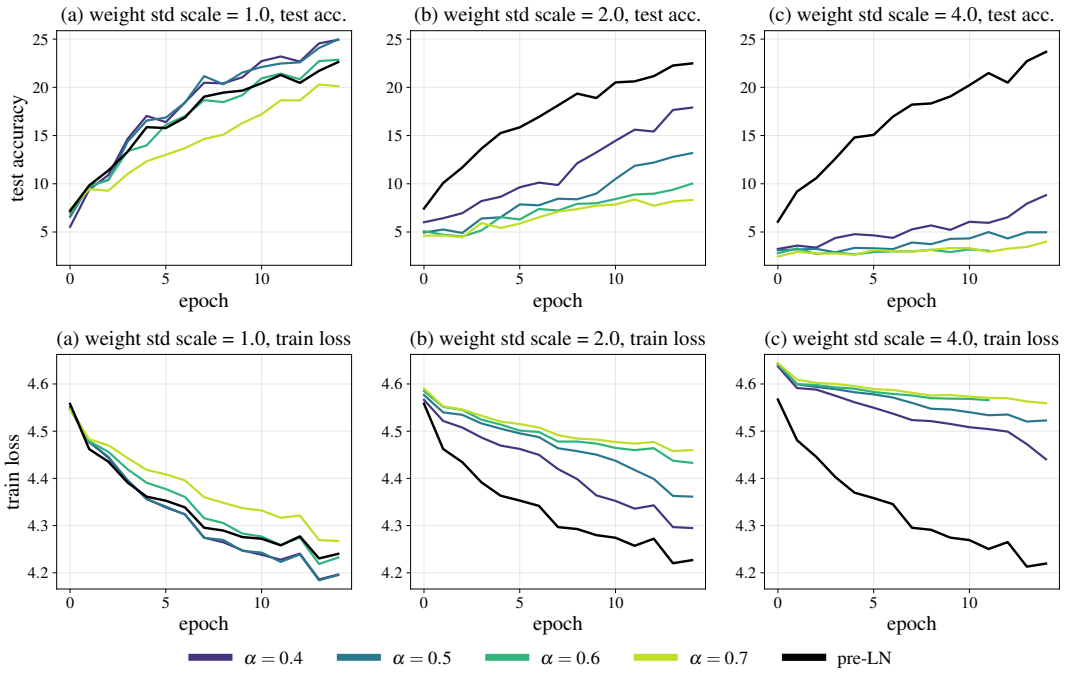

Figure 19: **Upper:** test performance curves; **lower:** training curves for the training-stability comparison between Derf and pre-LN when varying the weight standard deviation, shown in Fig. 4(b).

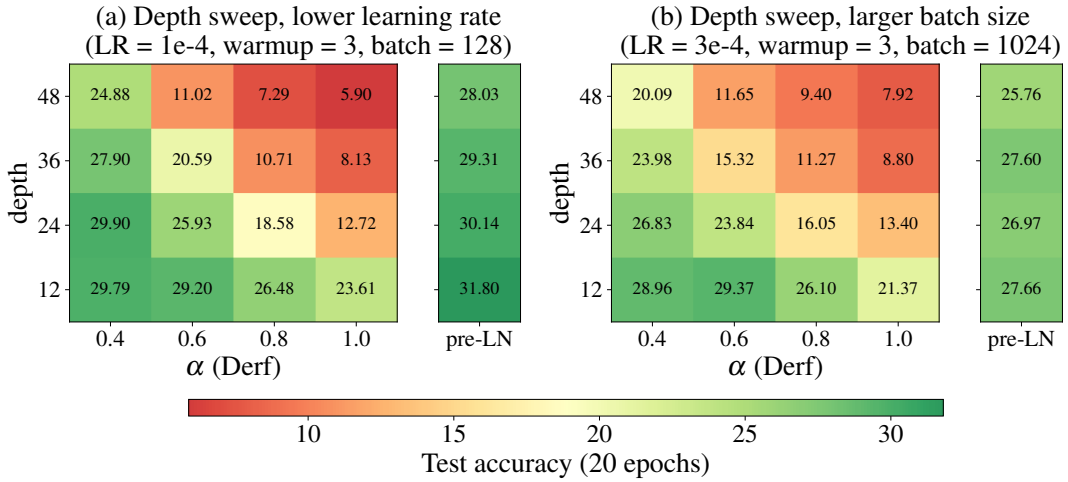

Figure 20: Training stability comparison between Derf and pre-LN as depth varies, using **(a)** a lower learning rate and **(b)** a larger batch size than in the main experiment in Fig. 4(a). All other parameters are kept identical. Both panels show test accuracy on CIFAR-100 after 20 training epochs versus depth. Compared to Fig. 4(a), both decreasing the learning rate and increasing the batch size shift the instability boundary toward larger α and larger depth.

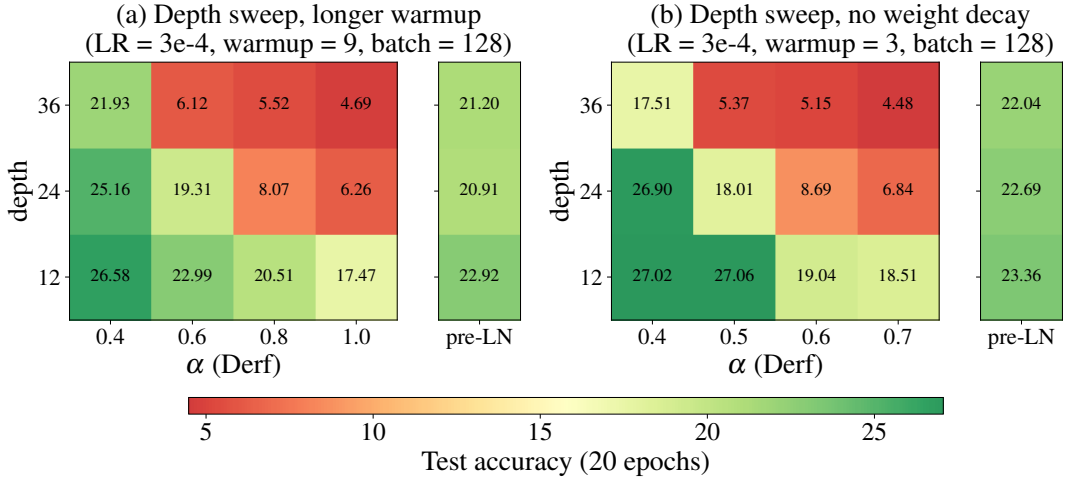

Figure 21: Training stability comparison between Derf and pre-LN as depth varies, using **(a)** a longer warmup and **(b)** no weight decay, relative to the main experiment shown in Fig. 4(a). All other parameters are kept identical. Both panels show test accuracy on CIFAR-100 after 20 training epochs as a function of depth. Compared to Fig. 4(a), increasing the number of warmup epochs improves performance near the instability boundary without substantially shifting it, while removing weight decay does not substantially affect the training stability.

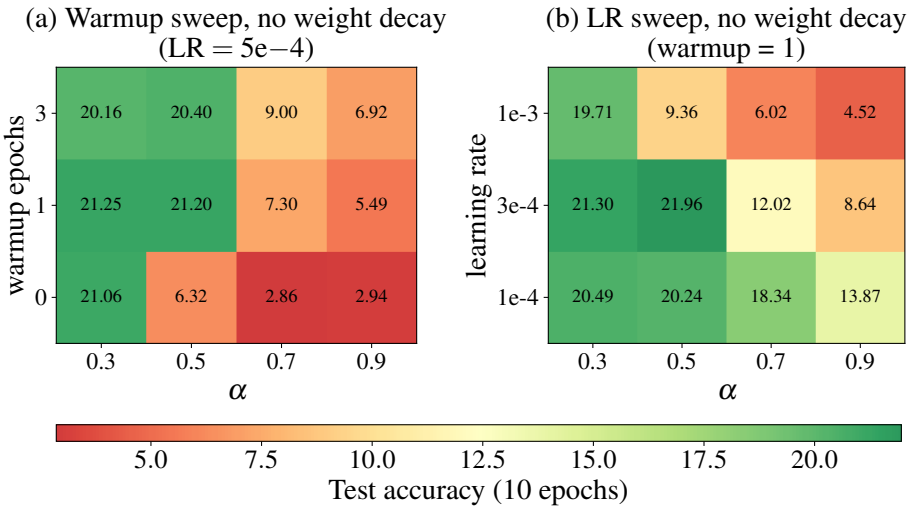

Figure 22: **(a)** Effect of the number of warmup epochs and **(b)** effect of the learning rate on the training stability of Derf for different α, *without weight decay*. Comparison to Fig. 5 suggests that moderate weight decay, or its absence, does not significantly affect the training-stability boundary.

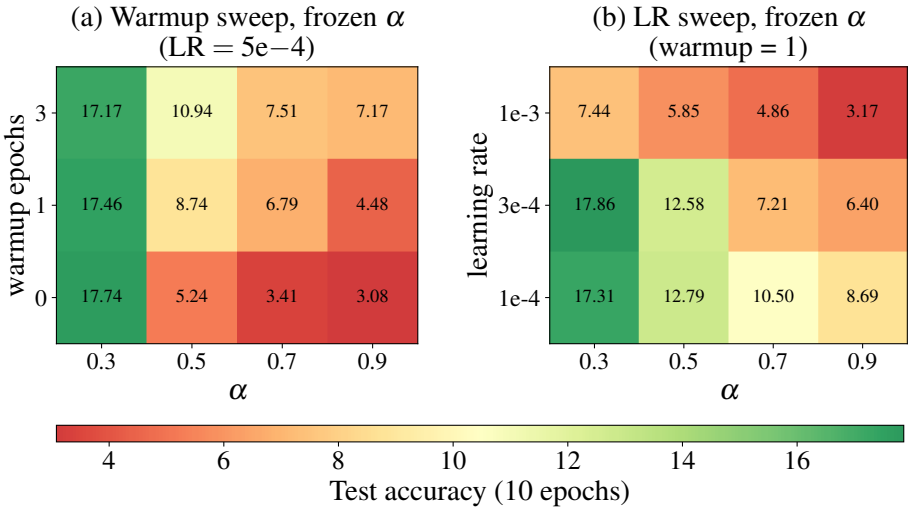

Figure 23: **(a)** Effect of the number of warmup epochs and **(b)** effect of the learning rate on the training stability of Derf for different α, *with α frozen during training*. Comparison to Fig. 5 suggests that making α trainable accelerates training convergence and improves stability at larger α and higher learning rates.

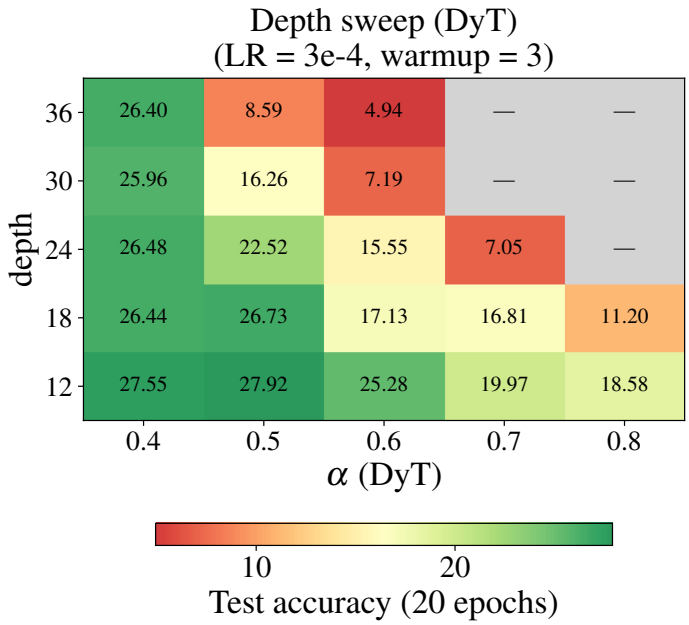

Figure 24: Test accuracy on CIFAR-100 after 20 training epochs for DyT models at different depths and initial values of α. Larger α leads to instability at smaller depths, consistent with the qualitative behavior observed for Derf.

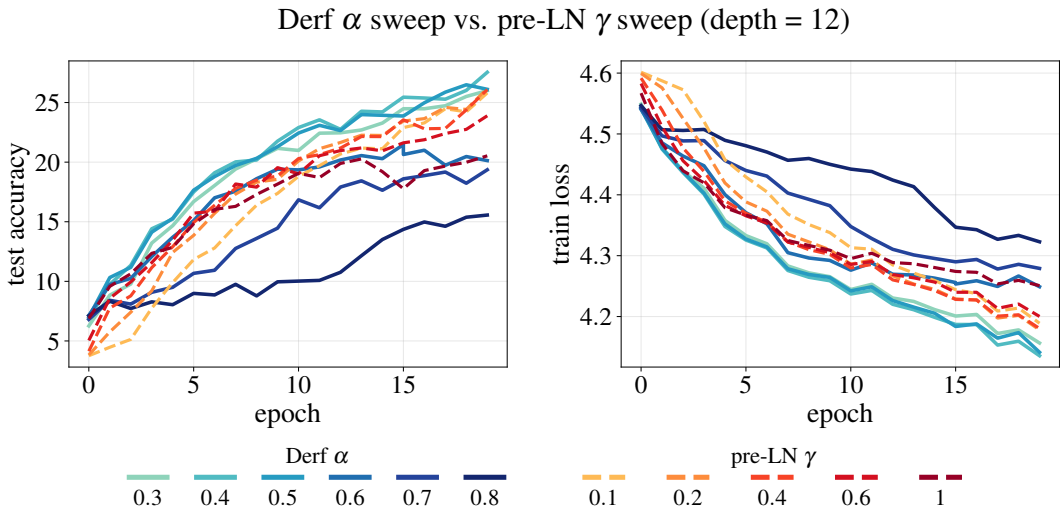

Figure 25: Comparison of a Derf α sweep and a pre-LN γ sweep at depth 12, showing test accuracy and training loss over 20 training epochs.

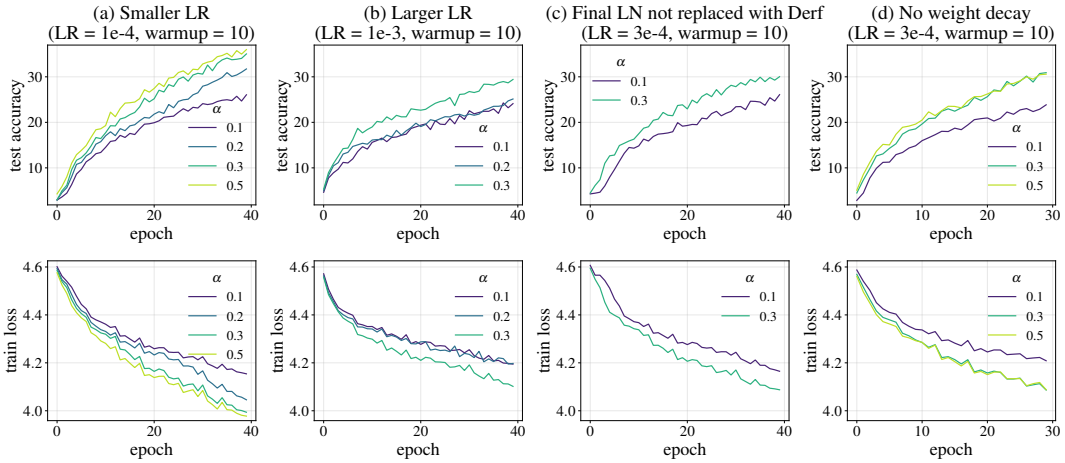

Figure 26: Effect of initializing Derf with an overly small α. **(a)** Smaller learning rate and **(b)** larger learning rate than in the main experiment in Fig. 5(c). **(c)** Same setup as Fig. 5(c), but with the final LayerNorm not replaced by Derf, to rule out an effect of small α acting directly at the output. **(d)** No weight decay. All four panels suggest that choosing α too small in Derf can slow convergence.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction accurately summarize the paper's scope: APJN-based signal-propagation theory for transformers, ViT APJN measurements, and training-stability experiments for Derf/DyT-like LayerNorm replacements. The introduction also states the main theoretical assumptions.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The paper discusses limitations in the main text and in the Limitations paragraph of the Conclusion. Sec. 3 and Appendix G discuss the limited applicability of the simplified theoretical assumptions to real inputs, while Sec. 4 and the Limitations paragraph discuss the empirical scope of the work.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: The theoretical results explicitly state the assumptions in Assumption 2.1. The main results are stated in Sec. 3, with derivations and proofs provided in Appendices B, C, and D.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The paper provides the model architecture, initialization, training hyperparameters, data preprocessing, APJN estimation procedure, and compute details needed to reproduce the main experiments in Appendix A. Additional details are provided in Appendices G and H.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: We attach code for reproducing the main results, including APJN computation in ViT models, comparison of measured APJNs to the theoretical recurrence predictions, and code to launch the training-stability runs. We will make this code available on GitHub upon paper acceptance.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: The paper specifies the ViT architecture, dataset preprocessing and augmentation, optimizer, learning-rate schedule, batch sizes, warmup lengths, weight decay, initialization scales, and the hyperparameters used for each main experiment in Appendix A. The APJN estimation protocol, including Hutchinson draws and numbers of weight initializations, is also described there.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [No]

Justification: The APJN experiments report variability across inputs, including per-sample GMFE values. For the training-stability sweeps, we do not report run-to-run variability over seeds; these experiments are intended to show large qualitative instability trends and approximate boundaries rather than statistically significant small accuracy differences. We note this limitation in Appendix A.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Appendix A reports the compute resources used, for APJN experiments and for training-stability experiments, approximate wall-clock time for representative runs, and an approximate total compute time.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research uses standard public datasets and architectures, does not involve human subjects, private data, or high-risk deployment, and is intended as a theoretical and empirical analysis of signal propagation and training stability. To the best of our knowledge, the work conforms with the NeurIPS Code of Ethics.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: The work is foundational research on signal propagation and initialization-time training stability rather than a deployed system or application-specific method. We do not identify direct societal impacts beyond generic downstream effects of improving understanding and stability of neural-network training.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper does not release pretrained models, scraped datasets, generative systems, or other assets with a high risk of misuse. The released code is intended only to reproduce APJN computations and training-stability experiments.

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper properly credits the existing assets used in the experiments, including the CIFAR-100 dataset, the `timm` ViT implementation, and the related code and model components on which our experiments build.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The main new asset released with the submission is code for reproducing the APJN computations, theoretical comparisons, and training-stability experiments. The attached code includes documentation and will be made available on GitHub upon paper acceptance.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

Justification: The paper does not involve crowdsourcing, user studies, or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [N/A]

Justification: The paper does not involve crowdsourcing, user studies, or research with human subjects, so IRB approval or equivalent review is not applicable.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [N/A]

Justification: LLMs are not used as an important, original, or non-standard component of the core research methodology. Any use of LLMs is limited to writing, editing, or formatting and does not affect the scientific contribution.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.