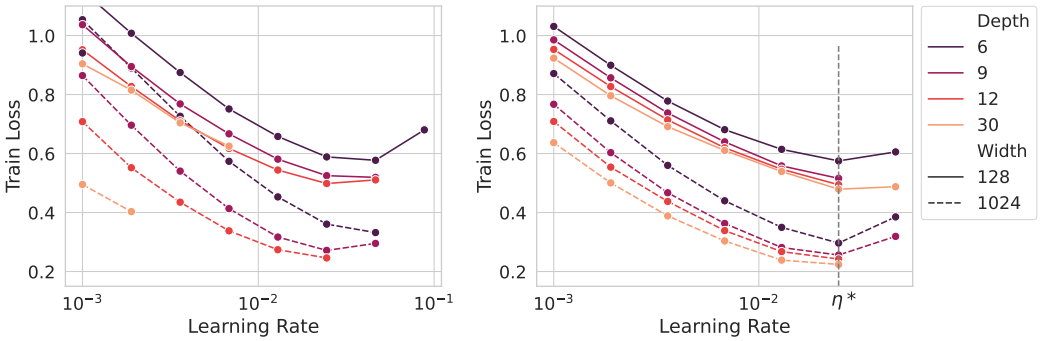
Depthwise Hyperparameter Transfer in Residual Networks: Dynamics and Scaling Limit

Blake Bordelon[*] [¶] **, Lorenzo Noci**[*] [♯] **, Mufan (Bill) Li** [§] **, Boris Hanin**[†] [§] **& Cengiz Pehlevan**[†] [¶]
[¶] Harvard University
[♯] ETH Zürich
[§] Princeton University

Abstract

We study residual networks with a residual branch scale of $1/\sqrt{\text{depth}}$ in combination with the μP parameterization. We provide experiments demonstrating that residual architectures including convolutional ResNets and Vision Transformers trained with this parameterization exhibit transfer of optimal hyperparameters across width and depth on CIFAR-10 and ImageNet. Furthermore, using recent developments in the dynamical mean field theory (DMFT) description of neural network learning dynamics, we show that this parameterization of ResNets admits a well-defined feature learning joint infinite-width and infinite-depth limit and show convergence of finite-size network dynamics towards this limit.

1 Introduction



(a) ResNet, μP

(b) This work: $\frac{1}{\sqrt{L}}$-ResNet + μP

Figure 1: The optimal learning rate η^* transfers across both depth and width in our proposed parameterization but not in μP or standard parameterization (Fig. C.2). Loss is plotted after 20 epochs on CIFAR-10. *All the missing datapoints indicate that the corresponding run diverged.*

Increasing the number of parameters in a neural network has led to often dramatic improvements in model quality [31, 26, 60, 32, 45], at the price of a costly hyperparameter tuning. To combat this, [57] proposed the so-called μP parameterization, which seeks to develop principles by which optimal hyperparameters from small networks can be reused — or *transferred* — to larger networks [55]. The μP prescription focuses on transfer from narrower to wider models, but does not always transfer across depth (see Figure 1(a)). This leads us to pose the following problem:

[*]Equal first authors.
[†]Equal senior authors.

Mathematics of Modern Machine Learning Workshop at NeurIPS 2023.

	SP (PyTorch)	μP / Mean Field	μP+$1/\sqrt{L}$-Residual (Ours)
Branch Scale β_ℓ	1	$\begin{cases} N^{-1/2}, & \ell > 0 \\ D^{-1/2}, & \ell = 0 \end{cases}$	$\begin{cases} N^{-1/2}, & \ell = L \\ (LN)^{-1/2}, & 0 < \ell < L \\ D^{-1/2}, & \ell = 0 \end{cases}$
Output Scale γ	1	$\gamma_0 N^{1/2}$	$\gamma_0 N^{1/2}$
LR Schedule $\eta(t)$	$\eta_0(t)$	$\eta_0(t)\gamma_0^2 N$	$\eta_0(t)\gamma_0^2 N$
Weight Variance σ_ℓ^2	$\begin{cases} N^{-1}, & \ell > 0 \\ D^{-1}, & \ell = 0 \end{cases}$	1	1

Table 1: In Equation 1, different choices of coefficients leads to standard parameterization (SP, PyTorch default [46]), μP / Mean Field parameterization, and our μP+$1/\sqrt{L}$-residual branch scaling.

Question: ***Can we transfer hyperparameters simultaneously across depth and width?***
In this work, we provide an affirmative answer for a particular flexible class of residual architectures (including ResNets and transformers) with residual branches scaled like $1/\sqrt{\text{depth}}$ (see Equation 1 and Table 1 for the exact parameterization). More specifically:

- We show that we the proposed model allows for transfer of learning rates across both depth and width (Equation 1 and Fig. 1(b)), with and without complex learning rate schedules (Fig. 3).
- We show that the lack of transfer across depth in μP ResNets can be partly overcome by adding normalization layers (Fig. 2), and it is improved with $1/\sqrt{\text{depth}}$ residual branches (Fig. 3 and C.5).
- We present evidence that our prescription leads to learning rate transfer in Vision Transformers (ViTs), both with and *without* normalization (Figure 3).
- We show transfer for momentum and weight decay coefficients (Fig. C.9).

The key underlying idea is that two neural networks will exhibit similar training dynamics if they are both finite size approximations to, or discretizations of, such a consistent scaling limit. Our main theoretical contributions can be summarized as follows:

- We show the infinite-width-and-depth limit of a residual network with $1/\sqrt{\text{depth}}$ residual branch scaling and μP can be characterized by a dynamical mean field theory (DMFT) (Section 4, 4.2).

2 Framework for Depthwise Hyperparameter Transfer

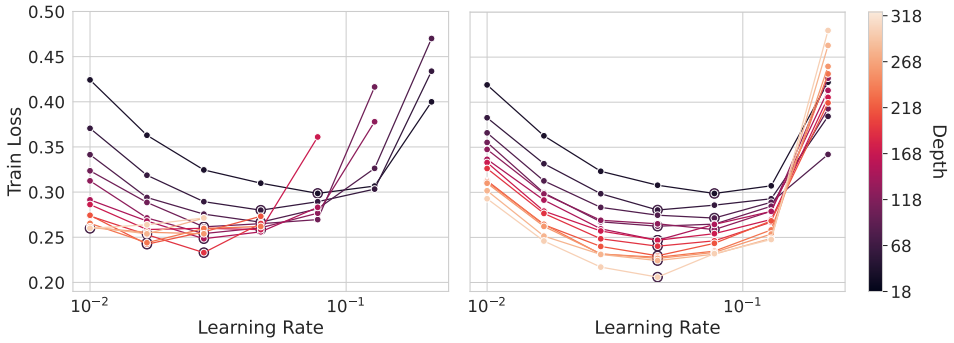
We consider the following type of residual network of width N and depth L with inputs $\boldsymbol{x} \in \mathbb{R}^D$ which are mapped to N-dimensional preactivations $\boldsymbol{h}^\ell \in \mathbb{R}^N$ and outputs $f(\boldsymbol{x}) \in \mathbb{R}$:

$$f(\boldsymbol{x}) = \frac{\beta_L}{\gamma} \boldsymbol{w}^L \cdot \phi(\boldsymbol{h}^L(\boldsymbol{x}))\,, \quad \boldsymbol{h}^{\ell+1}(\boldsymbol{x}) = \boldsymbol{h}^\ell(\boldsymbol{x}) + \beta_\ell \boldsymbol{W}^\ell \phi(\boldsymbol{h}^\ell(\boldsymbol{x}))\,, \quad \boldsymbol{h}^1(\boldsymbol{x}) = \beta_0 \boldsymbol{W}^0 \boldsymbol{x}, \quad (1)$$

where γ is a scaling factor, $\phi(\cdot)$ the activation function, and the weights are initialized as $\boldsymbol{W}_{ij}^\ell \sim \mathcal{N}(0, \sigma_\ell^2)$ with a corresponding learning rate $\eta(t)$. See Table 1. Compared to what is commonly done in practical residual architectures, our parameterization uses the $\sqrt{L}$ factor proposed in [24, 15], and the coefficient γ from μP, but do not use normalization layers (Table 1). The parameters $\boldsymbol{\theta} = \{\boldsymbol{W}^0, ..., \boldsymbol{w}^L\}$ are updated with a gradient based learning rule.

$$\boldsymbol{\theta}(t+1) = \boldsymbol{\theta}(t) - \eta(t)\, \mathbb{E}_{\boldsymbol{x} \sim \mathfrak{B}_t} \nabla_{\boldsymbol{\theta}}\, \mathcal{L}[f(\boldsymbol{\theta}, \boldsymbol{x})], \quad (2)$$

where $\mathfrak{B}_t$ is the minibatch at iteration t, $\eta(t)$ is the learning rate schedule, and the loss function $\mathcal{L}$ depends on parameters through the predictions f of the model. The choice of how $(\gamma, \eta(t))$ should be scaled as width N and depth L go to infinity determines the stability and feature learning properties of the network. In this work, we study the feature learning (*mean field*) parameterization $\gamma = \gamma_0\sqrt{N}$, $\eta(t) = \eta_0(t)\gamma_0^2\, N$, where $\gamma_0, \eta_0(t)$ are independent of N. One contribution of this work is identifying how η_0, γ_0 should be scaled with depth L in order to obtain a stable feature-learning large L limit. We argue — both theoretically and empirically — that after the $\frac{1}{\sqrt{L}}$ factors have been introduced the correct scaling is simply to take $\eta_0, \gamma_0 \sim \mathcal{O}(1)$.



(a) ResNet + μP, BatchNorm (b) $\frac{1}{\sqrt{L}}$ ResNet + μP, BatchNorm

Figure 2: Effect of batch normalization on a standard Resnet18-type architecture [25]. The above examples are taken after 20 epochs on CIFAR-10. Normalization layers can slightly improve consistency and trainability across depths in standard ResNets, but consistency is much more reliable with the $1/\sqrt{L}$ scaling (precisely, we use $\beta_\ell = 3/\sqrt{L}$ here to increase feature learning at finite depth). Runs that exceed a target loss of 0.5 are removed from the plot for visual clarity.

3 Experimental Verification of Hyperparameter Transfer

We report here a variety of experiments on hyperparameter transfer in residual networks. We begin with a basic modification of equation 1 by replacing fully connected layers with convolutional layers (Fig. 1, C.9, C.2 and C.1). We also experiment with a practical ResNet (in figures 2, C.4, C.5) (see Appendix B.2). Finally, we also perform selected experiments on Vision Transformers [14] (Figure 3). Below we list the the main takeaways from our experiments.

μP does not Transfer at Large Depths. While μP exhibits transfer over network widths, it does not immediately give transfer over network depths. In Figure 1(a), we show the train loss after 20 epochs on CIFAR-10. Notice how the network transfers over widths but not depths for μP. For completeness, in Fig. C.2 we show that also SP does not transfer.

$1/\sqrt{L}$-scaled Residual Networks Transfer. We repeat the same experiment, but now scale the residual branch by $1/\sqrt{L}$. The networks transfer over both width and depth (Figure 1 (b)).

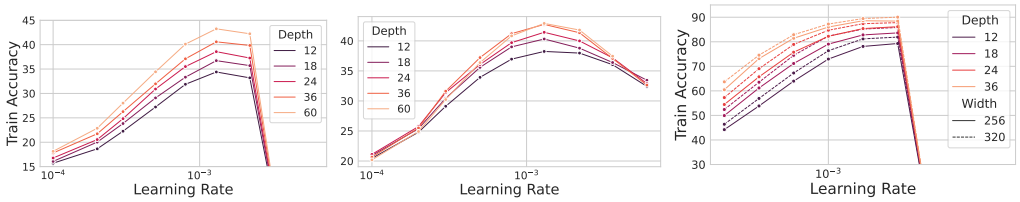
The Role of Normalization Layers. Normalization layers, such as LayerNorm and BatchNorm, are commonly used instead of the $1/\sqrt{L}$-scaling in residual architectures. In Fig. 2, we repeat the learning rate transfer experiment with BatchNorm placed after the convolutional layer. In Fig. C.4 we repeat the same experiment with LayerNorm. In both cases, while it is known that the presence of normalization layers certainly helps the trainability of the model at larger depth [13, 30], we observe that the $1/\sqrt{L}$-scaled version exhibits a more consistent transfer across both width and depth.

Empirical Verification on Vision Transformers with Adam. We empirically test whether the transfer results can be extended to Vision Transformers (ViTs) trained with μP-parameterized Adam optimizer [55]. In particular, we consider Pre-LN Transformers [53], and adapt it to our setting by scaling all the residual branches by $1/\sqrt{L}$ (as in [43]), and make the architecture compatible with the μP framework of [55] (see Appendix B.3 for details). We test the variants both with and without LayerNorm. Results for 20 epochs training with the Tiny ImageNet dataset are shown in Fig. 3 (a-c), where we also see a good transfer across all the settings.

Other Hyperparameters and Learning Rate Schedules also Transfer. Given the standard practice of using learning rate schedulers in optimizing Transformers [14, 51], we show that linear warm-up (Fig 3(a-c)), optionally followed by a cosine decay (Fig 3(c)) also transfer consistently. Finally, in Figure C.9, we plot the learning dynamics for momentum, weight decay, and the feature learning rate γ_0, which interpolates between the kernel and feature learning regimes [6].

4 Convergence to the Large Width and Depth Limit

We argued that width/depth-invariant feature and prediction updates are crucial for hyperparameter transfer. To formalize this, let $f_{N,L}(t)$ be the network output after t steps of optimization, where we explicitly write the dependence on width N and depth L. For hyperparameter transfer we want $f_{N,L}(t)$ to be close to $f_{N',L'}(t)$. One strategy to achieve this is to adopt a parameterization which



(a) $1/\sqrt{L}$ ViT + μP (b) ViT + μP, LayerNorm (c) $\frac{1}{\sqrt{L}}$ ViT + μP, Cosine Annealing

Figure 3: ViTs trained with Adam also exhibit learning rate transfer, with or without LayerNorm and $1/\sqrt{L}$-scaling. The above examples are taken after 20 epochs on Tiny ImageNet. In (a)-(c), the learning rate is linearly increased in the first 1000 steps. In (c), after 2000 warm-up steps, the learning rate is decreased to zero with a cosine schedule in 100 epochs.

makes finite networks as close as possible to a limit $f_{\infty,\infty}(t)$. The existence of the limit for the forward pass at initialization $t = 0$ was shown in [23, 12]. Here, we study the learning dynamics in $1/\sqrt{L}$ ResNets with μP parameterization, establish that the limit exists and and characterize the limit throughout training in App. D, E, F and finite-size approximations in App. H and I.

4.1 Primer on the DMFT in the $N \to \infty$ Limit

To derive our theoretical results, we use a technique known as dynamical mean field theory (DMFT) [6]. DMFT describes the dynamics of neural networks at (or near) infinite width (preserving feature learning). In this setting one tracks a set of kernel order parameters

$$\Phi^\ell(\boldsymbol{x}, \boldsymbol{x}'; t, s) = \frac{1}{N}\phi(\boldsymbol{h}^\ell(\boldsymbol{x}, t)) \cdot \phi(\boldsymbol{h}^\ell(\boldsymbol{x}', s)) \,, \quad G^\ell(\boldsymbol{x}, \boldsymbol{x}'; t, s) = \frac{1}{N}\boldsymbol{g}^\ell(\boldsymbol{x}, t) \cdot \boldsymbol{g}^\ell(\boldsymbol{x}', s), \tag{3}$$

where $\boldsymbol{g}^\ell(\boldsymbol{x}, t) \equiv N\gamma_0 \frac{\partial f(\boldsymbol{x},t)}{\partial \boldsymbol{h}^\ell(\boldsymbol{x},t)}$ are the back-propagation signals. DMFT reveals the following facts: (1) $\{\Phi^\ell, G^\ell\}$ and outputs f concentrate over random initializations of the weights throughout training. (2) The preactivations and gradient fields of each neuron $\{h_i^\ell, g_i^\ell\}$ become i.i.d. random variables drawn from a *single-site* (i.e. neuron marginal) density which depends on the kernels. (3) The kernels can be computed as averages over this *single-site* density.

4.2 The Large-Width-and-Depth Limit DMFT Equations

From the finite-L DMFT equations (App. D), we calculate the large depth $L \to \infty$ limit, generating a continuum process over the *layer time* $\tau = \frac{\ell}{L} \in [0, 1]$, a concept introduced in Li et al. [35].

Proposition 1 (Informal) *Consider training dynamics of a ResNet as in Equation 1 with our μP $\frac{1}{\sqrt{L}}$ scaling in the infinite width and depth limit. The preactivation $h(\tau; \boldsymbol{x}, t)$ drawn from the density of neurons at layer-time τ, data point $\boldsymbol{x}$ and training time t obeys a stochastic integral equation*

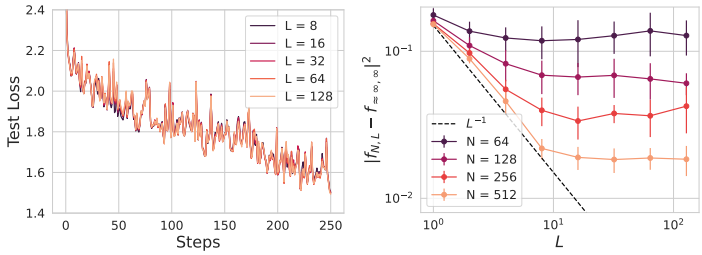
$$h(\tau; \boldsymbol{x}; t) = h(0; \boldsymbol{x}; t) + \int_0^\tau du(\tau'; \boldsymbol{x}; t) + \eta_0\gamma_0 \int_0^\tau d\tau' \int_0^t ds \int d\boldsymbol{x}' C_h(\tau'; \boldsymbol{x}, \boldsymbol{x}'; t, s) g(\tau'; \boldsymbol{x}'; s) \tag{4}$$

where $du(\tau; \boldsymbol{x}; t)$ is zero mean Brownian motion. The covariance of du is the feature kernel Φ and the deterministic operator C_h can be computed from the deterministic limiting kernels Φ, G. The gradients $g(\tau)$ satisfy an analogous integral equation (See App. E.7 for formulas for C_h and g). The stochastic variables h, g and the kernels Φ, G satisfy a closed system of equations.

A full proposition can be found in Appendix E.7. We derive this limit in the Appendix E, F. These equations are challenging to solve in the general case, as the h and g distributions are non-Gaussian. Its main utility in the present work is demonstrating that there is a well-defined feature-learning infinite width and depth limit and that, in the limit, the predictions, and kernels evolve by $\mathcal{O}(1)$.

4.3 Special Exactly Solveable Cases of the $N, L \to \infty$ DMFT equations

Though the resulting DMFT equations for the kernels are intractable in the general case, we can provide analytical solutions in special cases. We first write down the training dynamics in the lazy limit by characterizing the neural tangent kernel (NTK) at initialization. We discuss deep linear ResNets where the feature learning dynamics gives ODEs that can be closed while preserving Gaussianity of preactivations. Due to space limitation, these results are deferred to Appendix G.



(a) $N = 512$ SGD Dynamics (b) Convergence in L, N

Figure 4: Approximation of the $N \to \infty$, $L \to \infty$ limit requires sufficiently large N and L. CNNs are compared after 250 steps on CIFAR-10 with batchsize 32. As we cannot compute the infinite width and depth predictor, we use a proxy $f_{\approx\infty,\infty}$ ($N = 512$ and $L = 128$). (a) SGD dynamics at large width are consistent across depths. (b) The convergence of $f_{N,L}$ to a large width and depth proxy $f_{\approx\infty,\infty}$ is bottlenecked by width N for small N while for large N, it decays like $\mathcal{O}(L^{-1})$.

5 Discussion

In this work, we have proposed a simple parameterization of residual networks. In this parameterization, we have seen empirically that hyperparameter transfer is remarkably consistent across both width and depth for a variety of architectures, hyperparameters, and datasets. The experimental evidence is supported by theory, which shows that dynamics of all hidden layers are both non-trivial and independent of width and depth while not vanishing in the limit. We believe that these results have the potential to reduce computational costs of hyperparameter tuning, allowing the practitioner to train a large depth and width model only once with near optimal hyperparameters.

References

[1] Joseph M Antognini. Finite size corrections for neural network gaussian processes. *arXiv preprint arXiv:1908.10030*, 2019.

[2] David Balduzzi, Marcus Frean, Lennox Leary, JP Lewis, Kurt Wan-Duo Ma, and Brian McWilliams. The shattered gradients problem: If resnets are the answer, then what is the question? In *International Conference on Machine Learning*, pp. 342–350. PMLR, 2017.

[3] Jeremy Bernstein, Arash Vahdat, Yisong Yue, and Ming-Yu Liu. On the distance between two neural networks and the stability of learning. *Advances in Neural Information Processing Systems*, 33:21370–21381, 2020.

[4] Jeremy Bernstein, Chris Mingard, Kevin Huang, Navid Azizan, and Yisong Yue. Automatic gradient descent: Deep learning without hyperparameters. *arXiv preprint arXiv:2304.05187*, 2023.

[5] Blake Bordelon and Cengiz Pehlevan. The influence of learning rule on representation dynamics in wide neural networks. *arXiv preprint arXiv:2210.02157*, 2022.

[6] Blake Bordelon and Cengiz Pehlevan. Self-consistent dynamical field theory of kernel evolution in wide neural networks. *Advances in Neural Information Processing Systems*, 35:32240–32256, 2022.

[7] Blake Bordelon and Cengiz Pehlevan. Dynamics of finite width kernel and prediction fluctuations in mean field neural networks. *arXiv preprint arXiv:2304.03408*, 2023.

[8] Lenaic Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. *Advances in neural information processing systems*, 31, 2018.

[9] Lenaic Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on Learning Theory*, pp. 1305–1338. PMLR, 2020.

[10] Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in neural information processing systems*, 32, 2019.

[11] Youngmin Cho and Lawrence Saul. Kernel methods for deep learning. *Advances in neural information processing systems*, 22, 2009.

[12] Nicola Muca Cirone, Maud Lemercier, and Cristopher Salvi. Neural signature kernels as infinite-width-depth-limits of controlled resnets. *arXiv preprint arXiv:2303.17671*, 2023.

[13] Hadi Daneshmand, Amir Joudaki, and Francis Bach. Batch normalization orthogonalizes representations in deep random networks. *Advances in Neural Information Processing Systems*, 34:4896–4906, 2021.

[14] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

[15] Kirsten Fischer, David Dahmen, and Moritz Helias. Optimal signal propagation in resnets through residual scaling. *arXiv preprint arXiv:2305.07715*, 2023.

[16] Cedric Gerbelot, Emanuele Troiani, Francesca Mignacco, Florent Krzakala, and Lenka Zdeborova. Rigorous dynamical mean field theory for stochastic gradient descent methods. *arXiv preprint arXiv:2210.06591*, 2022.

[17] Boris Hanin. Which neural net architectures give rise to exploding and vanishing gradients? *Advances in neural information processing systems*, 31, 2018.

[18] Boris Hanin. Correlation functions in random fully connected neural networks at finite width. *arXiv preprint arXiv:2204.01058*, 2022.

[19] Boris Hanin and Mihai Nica. Finite depth and width corrections to the neural tangent kernel. *arXiv preprint arXiv:1909.05989*, 2019.

[20] Boris Hanin and Mihai Nica. Products of many large random matrices and gradients in deep neural networks. *Communications in Mathematical Physics*, 376(1):287–322, 2020.

[21] Boris Hanin and Alexander Zlokapa. Bayesian interpolation with deep linear networks. *Proceedings of the National Academy of Sciences*, 120(23):e2301345120, 2023.

[22] Soufiane Hayou. On the infinite-depth limit of finite-width neural networks. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL `https://openreview.net/forum?id=RbLsYz1Az9`.

[23] Soufiane Hayou and Greg Yang. Width and depth limits commute in residual networks. *arXiv preprint arXiv:2302.00453*, 2023.

[24] Soufiane Hayou, Eugenio Clerico, Bobby He, George Deligiannidis, Arnaud Doucet, and Judith Rousseau. Stable resnet. In *International Conference on Artificial Intelligence and Statistics*, pp. 1324–1332. PMLR, 2021.

[25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

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

[27] Xiao Shi Huang, Felipe Perez, Jimmy Ba, and Maksims Volkovs. Improving transformer optimization through better initialization. In *International Conference on Machine Learning*, pp. 4475–4483. PMLR, 2020.

[28] Kiyosi Itô. 109. stochastic integral. *Proceedings of the Imperial Academy*, 20(8):519–524, 1944.

[29] Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

[30] Amir Joudaki, Hadi Daneshmand, and Francis Bach. On the impact of activation and normalization in obtaining isometric embeddings at initialization. *arXiv preprint arXiv:2305.18399*, 2023.

[31] Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

[32] Tobit Klug and Reinhard Heckel. Scaling laws for deep learning based image reconstruction. In *The Eleventh International Conference on Learning Representations*, 2023. URL `https://openreview.net/forum?id=op-ceGueqc4`.

[33] Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

[34] Mufan Li, Mihai Nica, and Dan Roy. The future is log-gaussian: Resnets and their infinite-depth-and-width limit at initialization. *Advances in Neural Information Processing Systems*, 34: 7852–7864, 2021.

[35] Mufan Li, Mihai Nica, and Dan Roy. The neural covariance sde: Shaped infinite depth-and-width networks at initialization. *Advances in Neural Information Processing Systems*, 35: 10795–10808, 2022.

[36] Qianyi Li and Haim Sompolinsky. Statistical mechanics of deep linear neural networks: The backpropagating kernel renormalization. *Physical Review X*, 11(3):031059, 2021.

[37] Paul Cecil Martin, ED Siggia, and HA Rose. Statistical dynamics of classical systems. *Physical Review A*, 8(1):423, 1973.

[38] Alexander G de G Matthews, Mark Rowland, Jiri Hron, Richard E Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*, 2018.

[39] Song Mei, Theodor Misiakiewicz, and Andrea Montanari. Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. In *Conference on Learning Theory*, pp. 2388–2464. PMLR, 2019.

[40] Francesca Mignacco, Florent Krzakala, Pierfrancesco Urbani, and Lenka Zdeborová. Dynamical mean-field theory for stochastic gradient descent in gaussian mixture classification. *Advances in Neural Information Processing Systems*, 33:9540–9550, 2020.

[41] Radford M Neal. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media, 2012.

[42] Lorenzo Noci, Gregor Bachmann, Kevin Roth, Sebastian Nowozin, and Thomas Hofmann. Precise characterization of the prior predictive distribution of deep relu networks. *Advances in Neural Information Processing Systems*, 34:20851–20862, 2021.

[43] 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. *Advances in Neural Information Processing Systems*, 35:27198–27211, 2022.

[44] Lorenzo Noci, Chuning Li, Mufan Bill Li, Bobby He, Thomas Hofmann, Chris Maddison, and Daniel M Roy. The shaped transformer: Attention models in the infinite depth-and-width limit. *arXiv preprint arXiv:2306.17759*, 2023.

[45] OpenAI. Gpt-4 technical report, 2023.

[46] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019.

[47] Walter Rudin. *Principles of mathematical analysis*. 1953.

[48] Kai Segadlo, Bastian Epping, Alexander van Meegen, David Dahmen, Michael Krämer, and Moritz Helias. Unified field theoretical approach to deep and recurrent neuronal networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2022(10):103401, 2022.

[49] Masato Taki. Deep residual networks and weight initialization. *arXiv preprint arXiv:1709.02956*, 2017.

[50] Wojciech Tarnowski, Piotr Warchoł, Stanisław Jastrzebski, Jacek Tabor, and Maciej Nowak. Dynamical isometry is achieved in residual networks in a universal way for any activation function. In *The 22nd International Conference on Artificial Intelligence and Statistics*, pp. 2221–2230. PMLR, 2019.

[51] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: open and efficient foundation language models, 2023. *URL https://arxiv. org/abs/2302.13971*.

[52] Nikhil Vyas, Alexander Atanasov, Blake Bordelon, Depen Morwani, Sabarish Sainathan, and Cengiz Pehlevan. Feature-learning networks are consistent across widths at realistic scales, 2023.

[53] Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tieyan Liu. On layer normalization in the transformer architecture. In *International Conference on Machine Learning*, pp. 10524–10533. PMLR, 2020.

[54] Sho Yaida. Non-gaussian processes and neural networks at finite widths. In *Mathematical and Scientific Machine Learning*, pp. 165–192. PMLR, 2020.

[55] Ge Yang, Edward Hu, Igor Babuschkin, Szymon Sidor, Xiaodong Liu, David Farhi, Nick Ryder, Jakub Pachocki, Weizhu Chen, and Jianfeng Gao. Tuning large neural networks via zero-shot hyperparameter transfer. *Advances in Neural Information Processing Systems*, 34:17084–17097, 2021.

[56] Greg Yang. Tensor programs ii: Neural tangent kernel for any architecture. *arXiv preprint arXiv:2006.14548*, 2020.

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

[58] Jacob Zavatone-Veth and Cengiz Pehlevan. *Advances in Neural Information Processing Systems*, 34:3364–3375, 2021.

[59] Jacob Zavatone-Veth, Abdulkadir Canatar, Ben Ruben, and Cengiz Pehlevan. Asymptotics of representation learning in finite bayesian neural networks. *Advances in neural information processing systems*, 34:24765–24777, 2021.

[60] Xiaohua Zhai, Alexander Kolesnikov, Neil Houlsby, and Lucas Beyer. Scaling vision transformers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12104–12113, 2022.

[61] Hongyi Zhang, Yann N Dauphin, and Tengyu Ma. Fixup initialization: Residual learning without normalization. *arXiv preprint arXiv:1901.09321*, 2019.

Appendix

A Related Works

The $1/\sqrt{L}$-scaled residual model [25] adopted in this work belongs to a rapidly expanding research domain. This approach has several advantageous characteristics in signal propagation, in terms of dynamical isometry [50], as well as stable backward and forward kernels [2, 24, 15]. As a result, this has paved the way for the development of reliable initialization methods [49, 61], including Transformers without normalization layers [27, 43].

The study of neural network scaling limits started with MLPs in the infinite-width kernel regime [29, 41, 38, 33], where the preactivations are also Gaussian with deterministic covariance. However, the joint depth and width limit has been shown to be non-Gaussian [20, 42] and have a stochastic covariance [34]. In particular, the feature covariance of particular classes of networks have a limit described by stochastic differential equations (SDEs) [35, 44], where the layer time is the depth-to-width ratio in this case. On the other hand, [23] study the forward signal propagation in $1/\sqrt{L}$-scaled ResNets at initialization, where the preactivations converge to a Gaussian distribution with a *deterministic* covariance described by an ODE.

Transfer across width and depth was achieved on non-residual networks with the Automatic Gradient Descent algorithm proposed by [3, 4]. Successful hyperparameter transfer across network widths in μP was shown in [55]. Width-consistent early time prediction and representation dynamics were also observed in an empirical study of [52]. The infinite width limits of feature learning networks have been characterized with a mean-field PDE in two layer networks [8, 39, 9], Tensor programs [56, 57], and DMFT techniques [6, 5, 7], which have been proven useful to understand the gradient dynamics in other settings as well [40, 16]. This present work extends the DMFT technique to ResNets to analyze their feature learning behavior at large depth L.

Understanding feature learning in non-residual feedforward networks close to the asymptotic regime has also been studied with finite width and depth corrections from the corresponding infinite limit, both in the lazy [10] NTK [17, 19] and NNGP limits [1, 54, 36, 59, 48, 18], and more recently in the rich feature learning regime [7]. At any finite width and depth the analysis is even more challenging and has been studied in special cases using combinatorial approaches and hypergeometric functions [42, 58, 21].

B Experimental Details

B.1 Simplified ResNet

We design a residual architecture following the footprint of Eq. 1, and replacing the weight matrices with 1-strided convolutional layers with 3×3 kernel size. We max-pool the feature map at three selected layers equally distributed across depth. For instance, for a depth=12 model, we perform average pooling after the third, 6-th and 9-th layer. After each average pooling, we apply an extra convolutional layer to double the number of channels, which play the role of width in a convolutional network. Hence, also the width doubled three times across the depth (one time per average pooling). The final feature map is then flattened and passed through a final readout layer. The reported width in all the figures is the last width before the flattening operations. In the experiment with the $1/\sqrt{L}$-scaling, we divide by the total number of layers instead of the number of residual blocks (i.e. $L = 3 +$ # blocks). This extra factor does not affect the scaling limit.

B.2 ResNet

Compared to the simplified ResNet architecture of the previous section, the ResNet family as proposed in [25] are augmented with two convolutional layers per residual block, each followed by batch normalization normalization. The subsampling is performed by 2-strided convolutional layers, for a total of 4 times across the depth. For further details, we refer to [25]. In some experiments (e.g. Fig. C.4), we replace batch norm with LayerNorm applied across each two dimensional channel.

B.3 Vision Transformer

We use the standard ViT architecture as in [14], i.e. we tokenize the input images by splitting it into patches of fixed size, and embed it to N dimension using a linear layer. Then we apply a number of the usual Transformer blocks, each one composed of two residual blocks, a Softmax-based self attention block, and an MLP block with GeLU activation function. If used, we place LayerNorm at the beginning of each residual branch (Pre-LN configuration). Finally, we apply the following modification to make it compatible with μP [55]:

- We re-scale the Softmax logits (i.e. the product of queries and keys) of each attention layer by $1/N_q$. The usual convention would be to use instead $1/\sqrt{N_q}$.
- Initialize the queries to zero.
- Initialize all the other weights to have standard deviation of $1/\sqrt{N}$, where N here is the model dimension.

B.4 Details

We always use data augmentation, in particular the following random transformations:

- CIFAR-10: horizontal flips, 10-degree rotations, affine transformations, color jittering.
- Tiny ImageNet and ImageNet: crops, horizontal flips.

Across all the experiments, we fix the batch size to 64. Unless stated otherwise, all the hyperparameters that are not tuned are set to a default value: $\beta, \gamma_0, \sigma_\ell = 1$. For ViTs, we use a patch size We use the μP implementation of the μ-Readout layer, and optimizers (SGD, and Adam under μP parametrizstion) as in the released μP package.

C Additional Experiments

C.1 Depth Width Trade-Off

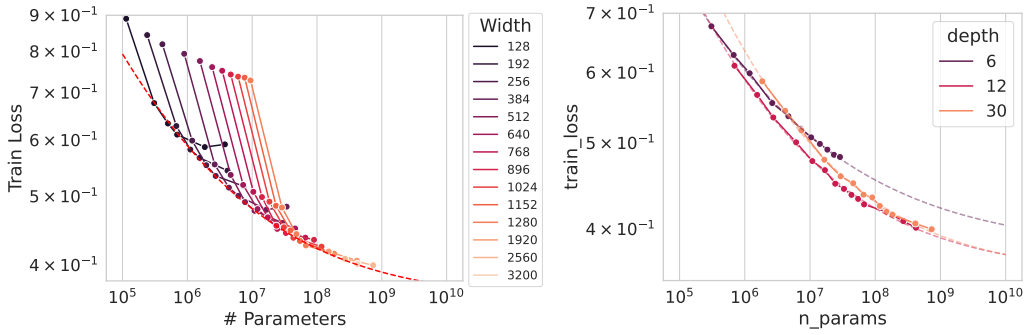
While tuning the architecture at small widths/depths, it is interesting to study along which direction — width or depth — the architecture should be scaled. To test this, we train the residual convolutional model (Sec. B.1) at relatively large scale (up to almost a billion parameters), for 20 epochs on CIFAR-10 with fixed learning rate of 0.046 (using data augmentation, see Sec. B). We report the results in Fig. C.1, where we plot the train loss as a function of parameters. In particular, in Fig. C.1(a) we highlight models of same widths, while in Fig. C.1(a) models of equal depth. We find that the Pareto optimal curve coincides by the depth 12 model for the range of widths/depths used for the experiment. As shown in Fig. C.1, we then fit power laws to the curves of constant depth, which seems to predict an optimal depth 30 at very large widths. Notice that we fix the number of epochs. We hypothesize the optimal curve to shift to different trade-offs as the number of epochs increases. We also expect different trade-offs for different datasets and models. However, extensively studying width-depth trade-off is beyond the scope of this work, thus we leave it as a potential direction for future work.

C.2 How Long Should you Train for Accurate Transfer?

An important question is after how many steps the optimal hyperparameters are fixed and can hence be selected without training for longer. In Fig. C.3(a), we plot the training loss profiles for two different depths and 4 different learning rates for a residual convolutional network of width 512 trained on CIFAR-10. Notice how after the first few epochs it is already possible to decide the optimal learning rate. We confirm this for a ViT trained on Tiny ImageNet in Fig. C.3(b-c), where the optimal learning rate at two selected epochs (3 and 9) is the same.

C.3 Other Experiments

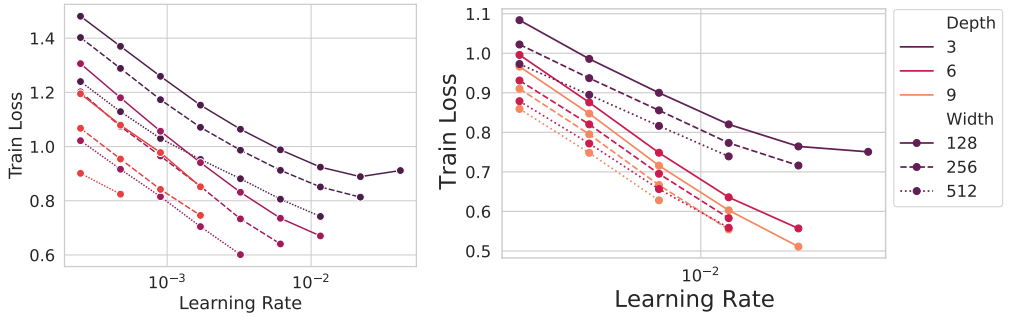
In Fig. C.2, we show how the learning rate does not transfer under SP for the residual model considered in this work, both with and without the $1/\sqrt{L}$-scaling. In Fig. 2 and Fig. C.4, we train



(a) Envelope of Depth-Width Trade-off

(b) Optimal Intermediate Depths

Figure C.1: Since all models have hyperparameter transfer across depths and widths, one can explore a trade-off between depth and width at the optimal learning rate. We illustrate this idea with a simple experiment on CIFAR-10. (a) We plot the final train loss as a function of parameters for different widths. (b) intermediate depth of $\sim 9 - 12$ is preferred at fixed parameter count.

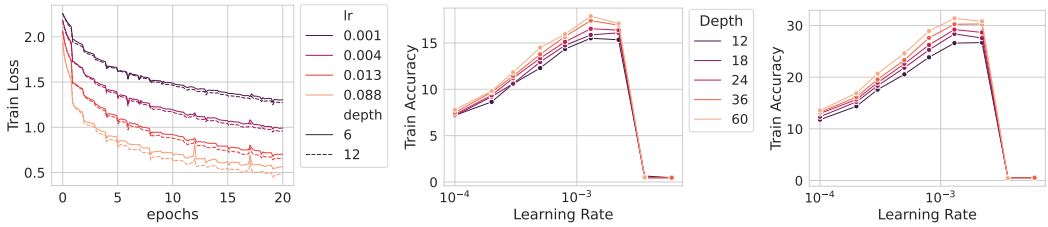


(a) ResNet, SP

(b) $\frac{1}{\sqrt{L}}$ ResNet SP

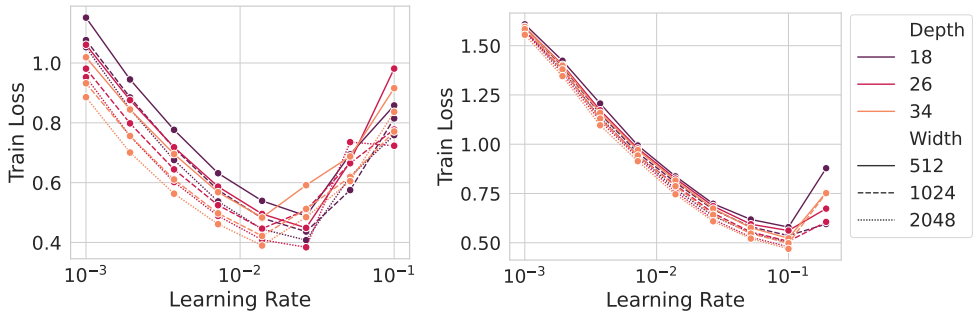
Figure C.2: Learning rates transfer across both depth and width in our proposed parameterization but not in standard parameterization (SP) or μP. Loss is plotted after 20 epochs on CIFAR-10. *All the missing datapoints indicate that the corresponding run diverged.* In particular, all depth 30 runs diverged when using μP without $1/\sqrt{\text{depth}}$ scaling.

models from the ResNets family, both with BatchNorm and with LayerNorm, concluding that the version with $1/\sqrt{L}$ scaling exhibits a better transfer. In Fig. C.5 we show our results on learning rate transfers for ImageNet, and in Fig. C.9 and Fig. C.10, we show how also other hyperparameters, such as weight decay, transfer under the proposed $1/\sqrt{L}$-scaling and have very consistent dynamics. The model used is the simplified residual convolutional model, trained for 20 epochs on CIFAR-10.



(a) LR dynamics (b) Epoch 3 (c) Epoch 9

Figure C.3: (a) dynamics of train loss for different depths and learning rates. The network is the residual convolutional network trained on CIFAR-10. (b-c) learning rate tranfer plots for the same ViT at two different checkpoints (epoch 3 and 5). Notice how the optimal learning rate is determined after the first few epochs of training.



(a) ResNet + μP, LayerNorm (b) $\frac{1}{\sqrt{L}}$ ResNet + μP, LayerNorm

Figure C.4: Effect of normalization layers. The above examples are taken after 20 epochs on CIFAR-10. Normalization layers can slightly improve consistency across depths in standard ResNets, but consistency is much more reliable with the $1/\sqrt{L}$ scaling.

D Derivation of the $N \to \infty$ limit at fixed L

In this section, we characterize the feature learning dynamics of the $N \to \infty$ limit of fixed depth L residual networks. Our derivation is an extension of the approach presented in [6]. Before taking the limit, we first identify our order parameters and weight space dynamics. Then we take the $N \to \infty$ limit using a technique known as Dynamical Mean Field Theory (DMFT).

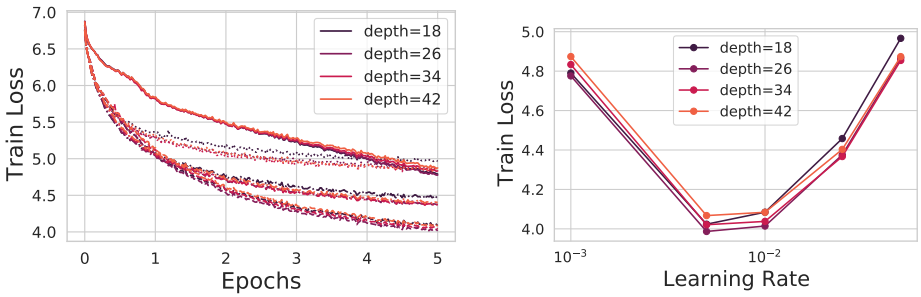
D.1 Finite N, L Function and Weight Dynamics

We start by computing the neural tangent kernel (NTK). For the sake of brevity, we will first write the equations where the read-in $\boldsymbol{W}^0$ weights are held fixed and will add their contribution to the dynamics in Appendix K.3. Excluding these two weight matrices from the present discussion does not alter any of the scalings with width N or depth L. The dynamics of the prediction f under gradient flow take the form

$$\frac{d}{dt} f(\boldsymbol{x}; t) = \eta_0 \, \mathbb{E}_{\boldsymbol{x}'} K(\boldsymbol{x}, \boldsymbol{x}'; t) \Delta(\boldsymbol{x}'; t) \tag{5}$$

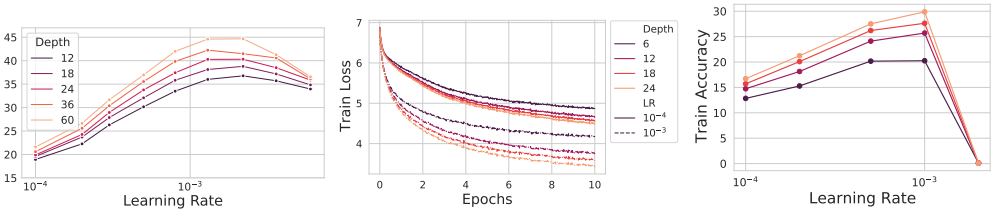
where $\Delta(\boldsymbol{x}; t) = -\frac{\partial \mathcal{L}(f(\boldsymbol{x}, t))}{\partial f(\boldsymbol{x}, t)}$ is an error signal and K is the NTK of this model, which takes the form

$$K(\boldsymbol{x}, \boldsymbol{x}') = \gamma_0^2 N \sum_{\ell=1}^{L} \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{W}^\ell} \cdot \frac{\partial f(\boldsymbol{x}')}{\partial \boldsymbol{W}^\ell}. \tag{6}$$



(a) Dynamics (b) Learning Rate Transfer

Figure C.5: Learning rate + depth sweep on ImageNet for 5 epochs with depth extrapolations of ResNet-18 *without any normalization layers*. Training is performed with SGD with momentum 0.9 and batchsize 128. (a) The loss curves for different learning rates (each learning rate is a distinct linestyle) and for different depths (colors). Dynamics are strikingly consistent across depths for each learning rate. (b) The optimal learning rate is preserved across depths.



(a) $1/\sqrt{L}$ ViT + μP, LayerNorm (b) $1/\sqrt{L}$ ViT ImageNet Dynamics (c) $1/\sqrt{L}$ ViT + μP, ImageNet

Figure C.6: ViTs trained with Adam also exhibit learning rate transfer, with or without LayerNorm and $1/\sqrt{L}$-scaling. The above examples are taken after 20 epochs on Tiny ImageNet for (a) and ImageNet after 10 epochs for (b)-(c). In (a), the learning rate is linearly increased in the first 1000 steps.

Explicitly calculating the gradients of the output with respect to the weights $\boldsymbol{W}^\ell$ gives

$$N\gamma_0 \frac{\partial f(\boldsymbol{x})}{\partial \boldsymbol{W}^\ell} = \frac{1}{\sqrt{NL}} \boldsymbol{g}^{\ell+1}(\boldsymbol{x}) \phi(\boldsymbol{h}^\ell(\boldsymbol{x}))^\top \ , \ g_i^\ell \equiv N\gamma_0 \frac{\partial f(\boldsymbol{x})}{\partial h_i^\ell(\boldsymbol{x})} \sim \mathcal{O}_{N,L}(1) \tag{7}$$

Using this above formula, we find that the NTK is $\mathcal{O}(1)$ with respect to N, L

$$K(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{L} \sum_{\ell=1}^{L} G^{\ell+1}(\boldsymbol{x}, \boldsymbol{x}') \Phi^\ell(\boldsymbol{x}, \boldsymbol{x}') \tag{8}$$

where we introduced the $\mathcal{O}(1)$ feature kernel Φ^ℓ and gradient kernels G^ℓ which have the form

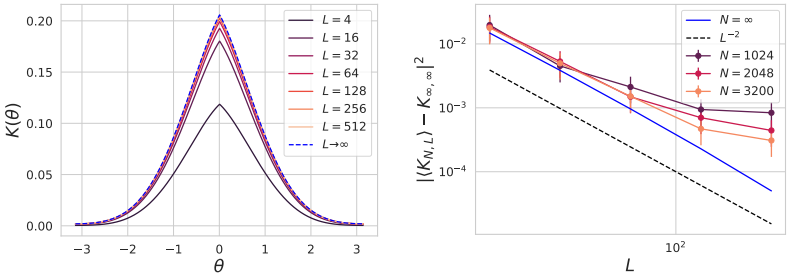
$$\Phi^\ell(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{N} \phi(\boldsymbol{h}^\ell(\boldsymbol{x})) \cdot \phi(\boldsymbol{h}^\ell(\boldsymbol{x}')) \ , \ G^\ell(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{N} \boldsymbol{g}^\ell(\boldsymbol{x}) \cdot \boldsymbol{g}^\ell(\boldsymbol{x}') \tag{9}$$

The base cases for the feature and gradient kernels are given by

$$\Phi^0(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{D} \boldsymbol{x} \cdot \boldsymbol{x}' \ , \ G^{L+1}(\boldsymbol{x}, \boldsymbol{x}') = 1. \tag{10}$$

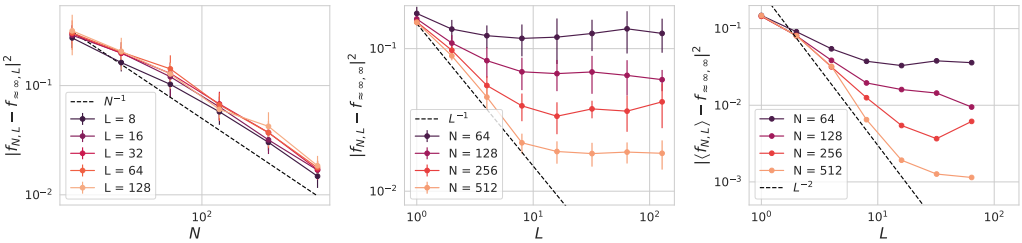
We next start by computing the dynamics of the weights under gradient flow

$$\frac{d}{dt} \boldsymbol{W}^\ell = \frac{\eta_0 \gamma_0}{\sqrt{LN}} \, \mathbb{E}_{\boldsymbol{x}} \, \Delta(\boldsymbol{x}; t) \boldsymbol{g}^{\ell+1}(\boldsymbol{x}, t) \phi(\boldsymbol{h}^\ell(\boldsymbol{x}, t))^\top. \tag{11}$$



(a) ReLU NTKs at Initialization (b) Initial NTK Convergence in L

Figure C.7: Convergence of the depth L NTK to the limiting kernel at initialization. (a) The infinite width NTK for a ReLU residual multilayer perpectron (MLP) for two inputs $\boldsymbol{x}, \boldsymbol{x}'$ separated by angle $\theta \in [-\pi, \pi]$ over varying depth L. The kernel converges as $L \to \infty$. (b) The convergence rate of the initialization averaged NTK $\langle K \rangle$ with network widths N and network depths L at initialization. For sufficiently large N, the *initial* NTK will converge at rate $\mathcal{O}(L^{-2})$ in square error.



(a) Fixed L Convergence in N (b) Average Square Error (c) Error of Ensembled Logits

Figure C.8: Further numerical support of our finite N, L error decomposition in Appendix I through ensembling logits over $E = 15$ random seeds in the same experimental setting as Figure 4. (a) At each depth L, the finite width N network converges in square error as $\mathcal{O}_{N,L}(N^{-1})$, independent of L. (b) The full expected square error from Figure 4 shows a convergence rate of $\mathcal{O}(L^{-1})$ for sufficiently large N, consistent with theory. (c) The ensemble averaged logits $\langle f_{L,N} \rangle$ converge to the infinite depth proxy at a faster rate, which is consistent with the theoretical $\mathcal{O}(L^{-2})$ for sufficiently large N.

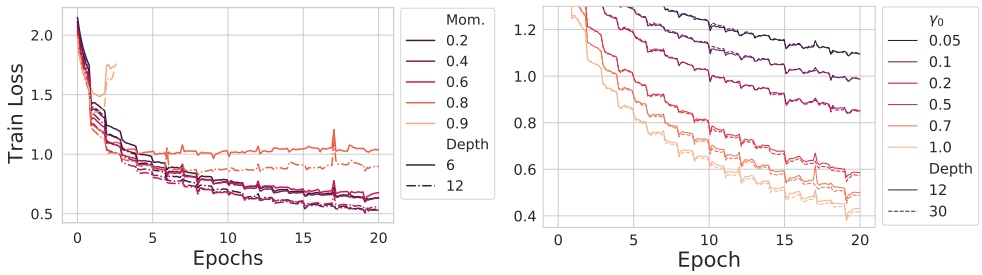
Integrating this equation from time 0 to time t, we get the following

$$\begin{aligned} \boldsymbol{h}^{\ell+1}(\boldsymbol{x}, t) =& \boldsymbol{h}^{\ell}(\boldsymbol{x}, t) + \frac{1}{\sqrt{NL}} \boldsymbol{W}^{\ell}(0) \phi(\boldsymbol{h}^{\ell}) \\ &+ \frac{\eta_0 \gamma_0}{L} \int_0^t ds \mathbb{E}_{\boldsymbol{x}'} \Delta(\boldsymbol{x}', s) \boldsymbol{g}^{\ell+1}(\boldsymbol{x}', s) \Phi^{\ell}(\boldsymbol{x}, \boldsymbol{x}'; t, s) \end{aligned} \tag{12}$$

where $\Phi^{\ell}(\boldsymbol{x}, \boldsymbol{x}'; t, s) = \frac{1}{N} \phi(\boldsymbol{h}^{\ell}(\boldsymbol{x}, t)) \cdot \phi(\boldsymbol{h}^{\ell}(\boldsymbol{x}'; s))$ is the double time-index generalization of the feature kernel. Similarly, we obtain a backward pass relation of the form

$$\begin{aligned} \boldsymbol{g}^{\ell}(\boldsymbol{x}, t) =& \boldsymbol{g}^{\ell+1}(\boldsymbol{x}, t) + \frac{1}{\sqrt{NL}} \dot{\phi}(\boldsymbol{h}^{\ell}(\boldsymbol{x}, t)) \odot \boldsymbol{W}^{\ell}(0)^{\top} \boldsymbol{g}^{\ell+1} \\ &+ \frac{\eta_0 \gamma_0}{L} \dot{\phi}(\boldsymbol{h}^{\ell}(\boldsymbol{x}, t)) \odot \int_0^t ds \mathbb{E}_{\boldsymbol{x}'} \Delta(\boldsymbol{x}', s) G^{\ell+1}(\boldsymbol{x}, \boldsymbol{x}'; t, s) \phi(\boldsymbol{h}^{\ell}(\boldsymbol{x}', s)) \end{aligned} \tag{13}$$

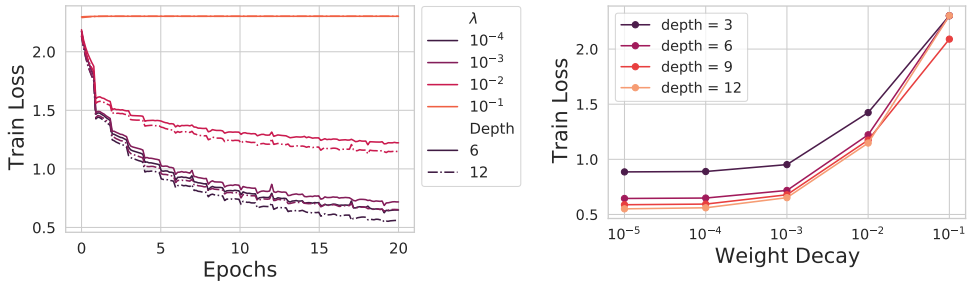
These equations define update rules for the preactivations $\boldsymbol{h}^{\ell}$ and the gradients $\boldsymbol{g}^{\ell}$ in terms of the feature and gradient kernels $\{\Phi^{\ell}, G^{\ell}\}$. However the above equations explicitly depend on the random initial weights $\boldsymbol{W}^{\ell}(0)$ through which time-varying signals propagate. Our ultimate goal is to gain insight into the effective dynamics over random initializations in the large N limit. To do that, we will invoke dynamical mean field theory. In the next section, we characterize how these terms behave at large N using a saddle point argument.



(a) Momentum

(b) Feature Learning Rate γ_0

Figure C.9: Other hyperparameters also transfer. This example shows training dynamics during the first 20 epochs on CIFAR-10 (architecture details in Appendix B). The dynamics at two different depths are provided for (a) momentum (b) feature learning rate γ_0.



(a) Weight Decay Dynamics

(b) Final Loss vs Depth

Figure C.10: Weight decay dynamics also transfer across depths.

D.2 Intuitive Explanation of Large Width Feature Learning Limit

In the limit of $N \to \infty$ with γ_0 held fixed, the above equations can be simplified. The two key effects that occur at large width are

- Kernels (feature kernels, gradient kernels, NTK) and output logits concentrate by a law of large numbers.
- At each layer, each of the hidden neuron's activities becomes independent draws from a *single site stochastic process* which is characterized by the kernels.

Once the single site density is known, the kernels can be computed as averages over the single site density. The next section derives this result formally from a Martin-Siggia-Rose Path Integral derivation of the Dynamical Mean Field Theory [37].

D.3 Path-integral and Saddle Point Equations in $N \to \infty$ Limit

This computation sketches the DMFT derivation of the limiting large N process. Examples of detailed computations of the DMFT action $\mathcal{S}$ can be found in the Appendix of [6, 5]. Our notation is chosen to match the derivations found in their Appendix.

To characterize the effect of the random initial weights, we will attempt to characterize the distribution of $\boldsymbol{h}^\ell(\boldsymbol{x}, t), \boldsymbol{g}^\ell(\boldsymbol{x}, t)$ by tracking the moment generating functional of these fields

$$Z[\{\boldsymbol{j}^h, \boldsymbol{j}_g\}] = \mathbb{E}_{\{\boldsymbol{W}^\ell(0)\}} \exp\left(\sum_{\ell=1}^{L} \int d\boldsymbol{x} \int dt \, [\boldsymbol{j}_h^\ell(\boldsymbol{x}, t) \cdot \boldsymbol{h}^\ell(\boldsymbol{x}, t) + \boldsymbol{j}_g(\boldsymbol{x}, t) \cdot \boldsymbol{g}^\ell(\boldsymbol{x}, t)]\right) \quad (14)$$

Moments of the fields can be obtained with differentiation near zero source, for example

$$\lim_{\{\boldsymbol{j}_h^\ell,\boldsymbol{j}_g^\ell\}\to 0} \frac{\delta}{\delta j^\ell_{h,i_1}(\boldsymbol{x}_1,t_1)} ... \frac{\delta}{\delta j^\ell_{h,i_k}(\boldsymbol{x}_k,t_k)} Z[\{\boldsymbol{j}^h,\boldsymbol{j}_g\}] = \left\langle h^{\ell_1}_{i_1}(\boldsymbol{x}_1,t_1)...h^{\ell_k}_{i_k}(\boldsymbol{x}_k,t_k)\right\rangle . \tag{15}$$

Now, following [6], we introduce the following useful auxiliary fields

$$\boldsymbol{\chi}^{\ell+1}(\boldsymbol{x},t) = \frac{1}{\sqrt{N}}\boldsymbol{W}^\ell(0)\phi(\boldsymbol{h}^\ell(\boldsymbol{x},t)) \ , \ \boldsymbol{\xi}^\ell(\boldsymbol{x},t) = \frac{1}{\sqrt{N}}\boldsymbol{W}^\ell(0)^\top \boldsymbol{g}^{\ell+1}(\boldsymbol{x},t) \tag{16}$$

The choice to introduce these auxiliary fields is apparent once one realizes that the h and g fields can be regarded as functions of χ, ξ and the kernels Φ, G. We enforce the constraints defining these fields with Dirac-delta functions by repeatedly multiplying by unity

$$\begin{aligned} 1 &= \int d\boldsymbol{\chi}^{\ell+1}(\boldsymbol{x},t)\ \delta\left(\boldsymbol{\chi}^{\ell+1}(\boldsymbol{x},t) - \frac{1}{\sqrt{N}}\boldsymbol{W}^\ell(0)\phi(\boldsymbol{h}^\ell(\boldsymbol{x},t))\right) \\ &= \int \frac{d\hat{\boldsymbol{\chi}}^{\ell+1}(\boldsymbol{x},t)d\boldsymbol{\chi}^{\ell+1}(\boldsymbol{x},t)}{(2\pi)^N} \exp\left(i\hat{\boldsymbol{\chi}}^{\ell+1}(\boldsymbol{x},t)\cdot\left[\boldsymbol{\chi}^{\ell+1}(\boldsymbol{x},t) - \frac{1}{\sqrt{N}}\boldsymbol{W}^\ell(0)\phi(\boldsymbol{h}^\ell(\boldsymbol{x},t))\right]\right) \end{aligned} \tag{17}$$

for all $\ell, \boldsymbol{x}, t$. And similarly for the $\boldsymbol{g}^\ell$ fields. After this insertion, the averages over $\{\boldsymbol{W}^\ell(0)\}$ can be performed with Gaussian integration. Concretely, we obtain the following average over Gaussian initialization $\boldsymbol{W}^\ell(0)$

$$\begin{aligned} &\ln \mathbb{E}\exp\left(-\frac{i}{\sqrt{N}}\text{Tr}\ \boldsymbol{W}^\ell(0)^\top \int d\boldsymbol{x}\int dt\ \left[\hat{\boldsymbol{\chi}}^{\ell+1}(\boldsymbol{x},t)\phi(\boldsymbol{h}^\ell(\boldsymbol{x},t))^\top + \boldsymbol{g}^{\ell+1}(\boldsymbol{x},t)\boldsymbol{\xi}^\ell(\boldsymbol{x},t)^\top\right]\right) \\ &= -\frac{1}{2}\int dtdsd\boldsymbol{x}d\boldsymbol{x}'\ \hat{\boldsymbol{\chi}}^{\ell+1}(\boldsymbol{x},t)\cdot\hat{\boldsymbol{\chi}}^{\ell+1}(\boldsymbol{x}',s)\Phi^\ell(\boldsymbol{x},\boldsymbol{x}';t,s) \\ &\quad -\frac{1}{2}\int dtdsd\boldsymbol{x}d\boldsymbol{x}'\ \hat{\boldsymbol{\xi}}^\ell(\boldsymbol{x},t)\cdot\hat{\boldsymbol{\xi}}^\ell(\boldsymbol{x}',s)G^{\ell+1}(\boldsymbol{x},\boldsymbol{x}';t,s) \\ &\quad - i\int dtdsd\boldsymbol{x}d\boldsymbol{x}'\ \hat{\boldsymbol{\chi}}^{\ell+1}(\boldsymbol{x},t)\cdot\boldsymbol{g}^{\ell+1}(\boldsymbol{x}',s)\ A^\ell(\boldsymbol{x},\boldsymbol{x}';t,s) \end{aligned} \tag{18}$$

where we introduced the following order parameters

$$\Phi^\ell(\boldsymbol{x},\boldsymbol{x}';t,s) = \frac{1}{N}\phi(\boldsymbol{h}^\ell(\boldsymbol{x},t))\cdot\phi(\boldsymbol{h}^\ell(\boldsymbol{x}',s)) \ , \ G^\ell(\boldsymbol{x},\boldsymbol{x}';t,s) = \frac{1}{N}\boldsymbol{g}^\ell(\boldsymbol{x},t)\cdot\boldsymbol{g}^\ell(\boldsymbol{x}',s) \tag{19}$$

$$A^\ell(\boldsymbol{x},\boldsymbol{x}';t,s) = -\frac{i}{N}\phi(\boldsymbol{h}^\ell(\boldsymbol{x},t))\cdot\hat{\boldsymbol{\xi}}^\ell(\boldsymbol{x}',s) \tag{20}$$

As we did with the $\boldsymbol{\chi}, \boldsymbol{\xi}$ fields, we can enforce the definition of the above order parameters with integral representations of Dirac-delta functions. For example, the Φ kernels can be enforced by inserting

$$\begin{aligned} 1 = \int \frac{d\Phi^\ell(\boldsymbol{x},\boldsymbol{x}';t,s)d\hat{\Phi}^\ell(\boldsymbol{x},\boldsymbol{x}';t,s)}{2\pi i N^{-1}} &\exp\left(N\Phi^\ell(\boldsymbol{x},\boldsymbol{x}';t,s)\hat{\Phi}^\ell(\boldsymbol{x},\boldsymbol{x}';t,s)\right) \\ &\exp\left(-\hat{\Phi}^\ell(\boldsymbol{x},\boldsymbol{x}';t,s)\phi(\boldsymbol{h}^\ell(\boldsymbol{x},t))\cdot\phi(\boldsymbol{h}^\ell(\boldsymbol{x}',t'))\right) \end{aligned} \tag{21}$$

where the $\hat{\Phi}$ integral is performed along the imaginary axis $(-i\infty, i\infty)$ in the complex plane. The same trick is performed for G^ℓ and A^ℓ which have conjugate variables $\hat{G}^\ell, B^\ell$ respectively. After inserting these new variables, we find that the moment generating function can be expressed as an integral over a collection of order parameters $\boldsymbol{q}$,

$$\boldsymbol{q} = \text{Vec}\{\Phi^\ell(\boldsymbol{x},\boldsymbol{x}';t,s), \hat{\Phi}^\ell(\boldsymbol{x},\boldsymbol{x}';t.s), G^\ell(\boldsymbol{x},\boldsymbol{x}',t,s), \hat{G}^\ell(\boldsymbol{x},\boldsymbol{x}',t,s), A^\ell(\boldsymbol{x},\boldsymbol{x}',t,s)B^\ell(\boldsymbol{x},\boldsymbol{x}',t,s)\}, \tag{22}$$

which is vectorized over all layers ℓ time points t, s and samples $\boldsymbol{x}, \boldsymbol{x}'$. The moment generating function takes the form

$$Z[\{\boldsymbol{j}_h,\boldsymbol{j}_g\}] = \int d\boldsymbol{q}\exp\left(N\mathcal{S}[\boldsymbol{q},\boldsymbol{j}_h,\boldsymbol{j}_g]\right) \tag{23}$$

where $\mathcal{S}$ is the $\mathcal{O}_N(1)$ DMFT action which has the following form

$$\mathcal{S} = \sum_\ell \int d\boldsymbol{x} d\boldsymbol{x}' dt ds \left[\Phi^\ell(\boldsymbol{x},\boldsymbol{x}';t,s)\hat{\Phi}^\ell(\boldsymbol{x},\boldsymbol{x}';t.s) + G^\ell(\boldsymbol{x},\boldsymbol{x}',t,s)\hat{G}^\ell(\boldsymbol{x},\boldsymbol{x}',t,s)\right]$$
$$- \sum_\ell \int d\boldsymbol{x} d\boldsymbol{x}' dt ds\, A^\ell(\boldsymbol{x},\boldsymbol{x}',t,s)B^\ell(\boldsymbol{x}',\boldsymbol{x},s,t) + \frac{1}{N}\sum_{i=1}^N \mathcal{Z}_i[\boldsymbol{q},\boldsymbol{j}_h,\boldsymbol{j}_g] \tag{24}$$

The functions $\mathcal{Z}_i$ are *single-site* moment generating functions (MGFs) which express the distribution of h_i, g_i for a given value of the order parameters $\boldsymbol{q}$ and sources $\boldsymbol{j}_h, \boldsymbol{j}_g$.

$$\mathcal{Z}_i = \int \prod_{\ell,\boldsymbol{x},t} d\hat{\chi}^\ell(\boldsymbol{x},t) d\hat{\chi}^\ell(\boldsymbol{x},t) d\hat{\chi}^\ell(\boldsymbol{x},t) d\hat{\chi}^\ell(\boldsymbol{x},t) \exp\left(-\sum_\ell \mathcal{H}_i^\ell[\hat{\chi},\hat{\xi},\chi,\xi,j_{h,i},j_{g,i}]\right) \tag{25}$$

where $\mathcal{H}_i^\ell$ are the single-site Hamiltonians

$$\begin{aligned}\mathcal{H}_i^\ell =& \frac{1}{2}\int d\boldsymbol{x} d\boldsymbol{x}' dt ds \left[\Phi^{\ell-1}(\boldsymbol{x},\boldsymbol{x}';t,s)\hat{\chi}^\ell(\boldsymbol{x},t)\hat{\chi}^\ell(\boldsymbol{x}',s) + G^{\ell+1}(\boldsymbol{x},\boldsymbol{x}';t,s)\hat{\xi}^\ell(\boldsymbol{x},t)\hat{\xi}^\ell(\boldsymbol{x}',s)\right] \\ &+ \int d\boldsymbol{x} d\boldsymbol{x}' dt ds \left[\hat{\Phi}^\ell(\boldsymbol{x},\boldsymbol{x}';t,s)\phi(h^\ell(\boldsymbol{x},t))\phi(h^\ell(\boldsymbol{x}',t')) + \hat{G}^\ell(\boldsymbol{x},\boldsymbol{x}';t,s)g^\ell(\boldsymbol{x},t)g^\ell(\boldsymbol{x}',s)\right] \\ &+ i\int d\boldsymbol{x} d\boldsymbol{x}' dt ds \left[A^{\ell-1}(\boldsymbol{x},\boldsymbol{x}';t,s)\hat{\chi}^\ell(\boldsymbol{x},t)g^\ell(\boldsymbol{x}',s) + B^\ell(\boldsymbol{x},\boldsymbol{x}';t,s)\hat{\xi}^\ell(\boldsymbol{x},t)\phi(h^\ell(\boldsymbol{x}',s))\right] \\ &- \int d\boldsymbol{x} dt \left[i\hat{\chi}^\ell(\boldsymbol{x},t)\chi^\ell(\boldsymbol{x},t) + i\hat{\xi}^\ell(\boldsymbol{x},t)\xi^\ell(\boldsymbol{x},t) + j_{h,i}^\ell(\boldsymbol{x},t)h^\ell(\boldsymbol{x},t) + j_{g,i}^\ell(\boldsymbol{x},t)g^\ell(\boldsymbol{x},t)\right]\end{aligned} \tag{26}$$

In the above formulas, the h, g fields should be regarded as functions of the χ, ξ fields.

Since the moment generating function has the form $Z = \int d\boldsymbol{q} \exp(N\mathcal{S}[\boldsymbol{q}])$, the $N \to \infty$ limit can thus be characterized by the steepest-descent method (saddle-point integration over $\boldsymbol{q}$). The result is that all of the order parameters take on definite values (concentrate) at infinite width. The saddle point equations $\frac{\partial \mathcal{S}}{\partial \boldsymbol{q}} = 0$ state that each of the kernels takes on definite values

$$\begin{aligned}\Phi^\ell(\boldsymbol{x},\boldsymbol{x}';t,s) &= \frac{1}{N}\sum_{i=1}^N \left\langle \phi(h^\ell(\boldsymbol{x},t))\phi(h^\ell(\boldsymbol{x}',s))\right\rangle_i \\ G^\ell(\boldsymbol{x},\boldsymbol{x}';t,s) &= \frac{1}{N}\sum_{i=1}^N \left\langle g^\ell(\boldsymbol{x},t)g^\ell(\boldsymbol{x}',s)\right\rangle_i \\ A^\ell(\boldsymbol{x},\boldsymbol{x}';t,s) &= -\frac{i}{N}\sum_{i=1}^N \left\langle \phi(h^\ell(\boldsymbol{x},t))\hat{\xi}^\ell(\boldsymbol{x}',s)\right\rangle_i \\ B^\ell(\boldsymbol{x},\boldsymbol{x}';t,s) &= -\frac{i}{N}\sum_{i=1}^N \left\langle g^{\ell+1}(\boldsymbol{x},t)\hat{\chi}^{\ell+1}(\boldsymbol{x}',s)\right\rangle_i \\ \hat{\Phi}^\ell(\boldsymbol{x},\boldsymbol{x}';t,s) &= -\frac{1}{2N}\sum_{i=1}^N \left\langle \hat{\chi}^\ell(\boldsymbol{x},t)\hat{\chi}^\ell(\boldsymbol{x}',s)\right\rangle_i \\ \hat{G}^\ell(\boldsymbol{x},\boldsymbol{x}';t,s) &= -\frac{1}{2N}\sum_{i=1}^N \left\langle \hat{\xi}^\ell(\boldsymbol{x},t)\hat{\xi}^\ell(\boldsymbol{x}',s)\right\rangle_i\end{aligned} \tag{27}$$

where $\langle\rangle_i$ denotes averaging over a *single site stochastic process* defined by the moment generating function $\mathcal{Z}_i$. At zero source $\boldsymbol{j} \to 0$ all of the single site MGF functions $\mathcal{Z}_i$ are identical so all averages $\langle\cdot\rangle_i$ are also identical. At zero source, the kernels have the simple expressions such as $\Phi^\ell(\boldsymbol{x},\boldsymbol{x}';t,s) = \left\langle \phi(h^\ell(\boldsymbol{x},t))\phi(h^\ell(\boldsymbol{x}',s))\right\rangle$.

D.3.1 Hubbard-Stratonovich transform

Now that we know the kernels Φ^ℓ, G^ℓ take on definite values at $N \to \infty$, we can perform a Hubbard-Stratonovich transformation to introduce Gaussian sources

$$\begin{aligned}
&\exp\left(-\frac{1}{2}\int d\boldsymbol{x}d\boldsymbol{x}'dtds\ \hat{\chi}^\ell(\boldsymbol{x},t)\hat{\chi}^\ell(\boldsymbol{x}',t')\Phi^{\ell-1}(\boldsymbol{x},\boldsymbol{x}';t,s)\right) \\
&= \left\langle \exp\left(-i\int d\boldsymbol{x}dt\ \hat{\chi}^\ell(\boldsymbol{x},t)u^\ell(\boldsymbol{x},t)\right)\right\rangle_{u^\ell\sim\mathcal{GP}(0,\Phi^{\ell-1})} \qquad (28)\\
&\exp\left(-\frac{1}{2}\int d\boldsymbol{x}d\boldsymbol{x}'dtds\ \hat{\xi}^\ell(\boldsymbol{x},t)\hat{\xi}^\ell(\boldsymbol{x}',t')G^{\ell+1}(\boldsymbol{x},\boldsymbol{x}';t,s)\right) \\
&= \left\langle \exp\left(-i\int d\boldsymbol{x}dt\ \hat{\xi}^\ell(\boldsymbol{x},t)r^\ell(\boldsymbol{x},t)\right)\right\rangle_{r^\ell\sim\mathcal{GP}(0,G^{\ell+1})} \qquad (29)
\end{aligned}$$

After introducing these Gaussian sources u^ℓ, r^ℓ, we have linearized the single site Hamiltonians $\mathcal{H}_i^\ell$ with respect to $\hat{\chi}$ and $\hat{\xi}$. Integration over these variables gives us the following Dirac-Delta function constraints

$$\begin{aligned}
\chi^\ell(\boldsymbol{x},t) &= u^\ell(\boldsymbol{x},t) + \int d\boldsymbol{x}ds\ A^{\ell-1}(\boldsymbol{x},\boldsymbol{x}';t,s)g^\ell(\boldsymbol{x}',s) \\
\xi^\ell(\boldsymbol{x},t) &= r^\ell(\boldsymbol{x},t) + \int d\boldsymbol{x}ds\ B^\ell(\boldsymbol{x},\boldsymbol{x}';t,s)\phi(h^\ell(\boldsymbol{x}',s)) \qquad (30)
\end{aligned}$$

Using similar manipulations, we can simplify the expressions for $A^\ell, B^\ell, \hat{\Phi}, \hat{G}$ at zero source $\boldsymbol{j}=0$

$$\begin{aligned}
A^\ell(\boldsymbol{x},\boldsymbol{x}';t,s) &= \left\langle \frac{\delta\phi(h^\ell(\boldsymbol{x},t))}{\delta r^\ell(\boldsymbol{x}',s)}\right\rangle \\
B^\ell(\boldsymbol{x},\boldsymbol{x}';t,s) &= \left\langle \frac{\delta g^{\ell+1}(\boldsymbol{x},t)}{\delta u^\ell(\boldsymbol{x}',s)}\right\rangle \\
\hat{\Phi}^\ell(\boldsymbol{x},\boldsymbol{x}';t,s) &= 0\ ,\ \hat{G}^\ell(\boldsymbol{x},\boldsymbol{x}';t,s) = 0 \qquad (31)
\end{aligned}$$

Plugging the dynamics for χ, ξ into the formulas for h^ℓ, g^ℓ, we obtain the following final equations for the stochastic process of h^ℓ, g^ℓ.

$$\boxed{\begin{aligned}
h^{\ell+1}(\boldsymbol{x},t) &= h^\ell(\boldsymbol{x},t) + \frac{1}{\sqrt{L}}u^{\ell+1}(\boldsymbol{x},t) + \frac{1}{\sqrt{L}}\int d\boldsymbol{x}'ds\ A^\ell(\boldsymbol{x},\boldsymbol{x}';t,s)g^{\ell+1}(\boldsymbol{x}',s) \\
&\quad + \frac{\eta_0\gamma_0}{L}\mathbb{E}_{\boldsymbol{x}'}\int_0^t ds\ \Delta(\boldsymbol{x}',s)\Phi^\ell(\boldsymbol{x},\boldsymbol{x}';t,s)\ g^{\ell+1}(\boldsymbol{x}',s)\ ,\ u^\ell\sim\mathcal{GP}(0,\Phi^{\ell-1}) \\
g^\ell(\boldsymbol{x},t) &= g^{\ell+1}(\boldsymbol{x},t) + \frac{1}{\sqrt{L}}\dot{\phi}(h^\ell(\boldsymbol{x},t))z^\ell(\boldsymbol{x},t)\ , \\
z^\ell(\boldsymbol{x},t) &= r^\ell(\boldsymbol{x},t) + \int d\boldsymbol{x}'ds\ B^\ell(\boldsymbol{x},\boldsymbol{x}';t,s)\phi(h^\ell(\boldsymbol{x}',s))\ ,\ r^\ell\sim\mathcal{GP}(0,G^{\ell+1}) \\
&\quad + \frac{\eta_0\gamma_0}{\sqrt{L}}\mathbb{E}_{\boldsymbol{x}'}\int_0^t ds\ \Delta(\boldsymbol{x}',s)G^{\ell+1}(\boldsymbol{x},\boldsymbol{x}';t,s)\phi(h^\ell(\boldsymbol{x}',s))
\end{aligned}} \qquad (32a)$$

where the order parameters satisfy the the following saddle point equations

$$\boxed{\begin{aligned}
&\Phi^\ell(\boldsymbol{x},\boldsymbol{x}';t,s) = \left\langle \phi(h^\ell(\boldsymbol{x},t))\phi(h^\ell(\boldsymbol{x}',s))\right\rangle \ , \ G^\ell(\boldsymbol{x},\boldsymbol{x}';t,s) = \left\langle g^\ell(\boldsymbol{x},t)g^\ell(\boldsymbol{x}',s)\right\rangle \\
&A^\ell(\boldsymbol{x},\boldsymbol{x}';t,s) = \left\langle \frac{\delta\phi(h^\ell(\boldsymbol{x},t))}{\delta r^\ell(\boldsymbol{x}',s)}\right\rangle \ , \ B^\ell(\boldsymbol{x},\boldsymbol{x}';t,s) = \left\langle \frac{\delta g^{\ell+1}(\boldsymbol{x},t)}{\delta u^{\ell+1}(\boldsymbol{x}',s)}\right\rangle \\
&\frac{df(\boldsymbol{x},t)}{dt} = \eta_0 \mathbb{E}_{\boldsymbol{x}'} K(\boldsymbol{x},\boldsymbol{x}',t)\Delta(\boldsymbol{x}',t) \ , \ \Delta(\boldsymbol{x},t) = -\frac{\partial \mathcal{L}}{\partial f(\boldsymbol{x},t)} \\
&K(\boldsymbol{x},\boldsymbol{x}',t) = \frac{1}{L}\sum_{\ell=1}^{L} G^{\ell+1}(\boldsymbol{x},\boldsymbol{x}';t,s)\Phi^\ell(\boldsymbol{x},\boldsymbol{x}';t,s)
\end{aligned}} \tag{33a}$$

Up to the residual skip connections and the factors of $\frac{1}{\sqrt{L}}$, this agrees with the $N \to \infty$ limit DMFT derived in [6]. At fixed L this above equation can be solved with a self-consistent Monte Carlo sampling procedure. For deep linear networks, the equations close since all h, g fields are Gaussian. In the next section, we analyze the behavior of the above equations at large depth L so that we can take a large depth limit.

E The $L \to \infty$ limit from the DMFT Saddle Point Equations

In this section, we consider how the infinite width dynamics behaves at large depth. We will show that the limiting dynamics takes the form of a system of ODEs for the kernels in terms of layer time $\tau = \ell/L$. First, we will show that the response functions need to be rescaled with respect to L. Then we will take a continuum limit over layers to arrive at the final ODEs for the Φ and G correlation functions.

E.1 Unfolding the Layer Recursion

A useful first step which we will use in the following analysis is "unfolding" the recurrence to get explicit formulas for h^ℓ in terms of all of the Gaussian sources u and all of the gradient fields g^ℓ.

$$\begin{aligned}
h^\ell(\boldsymbol{x},t) =& h^1(\boldsymbol{x},t) + \frac{1}{\sqrt{L}}\sum_{k=0}^{\ell} u^k(\boldsymbol{x},t) + \frac{1}{\sqrt{L}}\sum_{k=1}^{\ell-1}\int d\boldsymbol{x}' ds\ A^k(\boldsymbol{x},\boldsymbol{x}';t,s) g^{k+1}(\boldsymbol{x}',s) \\
&+ \frac{\eta_0\gamma_0}{L}\mathbb{E}_{\boldsymbol{x}'}\int_0^t ds\ \Delta(\boldsymbol{x}',s)\sum_{k=0}^{\ell-1}\Phi^k(\boldsymbol{x},\boldsymbol{x}',t,s) g^{k+1}(\boldsymbol{x}',s)
\end{aligned} \tag{34}$$

and similarly for the backward pass, we have

$$\begin{aligned}
g^\ell(\boldsymbol{x},t) =& g^L(\boldsymbol{x},t) + \frac{1}{\sqrt{L}}\sum_{k=\ell}^{L}\dot{\phi}(h^k(\boldsymbol{x},t)) r^k(\boldsymbol{x},t) \\
&+ \frac{1}{\sqrt{L}}\sum_{k=\ell}^{L}\dot{\phi}(h^k(\boldsymbol{x},t))\int d\boldsymbol{x}' ds\ B^k(\boldsymbol{x},\boldsymbol{x}';t,s)\phi(h^k(\boldsymbol{x}',s)) \\
&+ \frac{\eta_0\gamma_0}{L}\sum_{k=\ell}^{L}\dot{\phi}(h^k(\boldsymbol{x},t))\mathbb{E}_{\boldsymbol{x}'}\int_0^t ds\ \Delta(\boldsymbol{x}',s)\sum_{k=1}^{\ell-1} G^k(\boldsymbol{x},\boldsymbol{x}',t,s)\phi(h^k(\boldsymbol{x}',s))
\end{aligned} \tag{35}$$

A very useful fact that can be immediately gleaned from the above equations is that

$$\begin{aligned}
\forall k \le \ell\ ,\ \frac{\delta h^\ell(\boldsymbol{x},t)}{\delta u^k(\boldsymbol{x},t)} \sim \mathcal{O}\left(\frac{1}{\sqrt{L}}\right) \\
\forall k \ge \ell\ ,\ \frac{\delta g^\ell(\boldsymbol{x},t)}{\delta r^k(\boldsymbol{x},t)} \sim \mathcal{O}\left(\frac{1}{\sqrt{L}}\right)
\end{aligned} \tag{36}$$

These facts will be utilized to obtain the scale of the response functions in the large depth limit.

E.2 Characterizing Response Functions at Large Depth

First, we will study the response functions A^ℓ, B^ℓ. We note that from the saddle point equations,

$$A^\ell(\boldsymbol{x}, \boldsymbol{x}'; t, s) = \left\langle \frac{\delta\phi(h^\ell(\boldsymbol{x}, t'))}{\delta r^\ell(\boldsymbol{x}', s)} \right\rangle. \tag{37}$$

Since $\frac{\delta\phi(h)}{\delta r} = \dot{\phi}(h)\frac{\delta h}{\delta r}$, it suffices to consider the scale of the derivative $\frac{\delta h^\ell(\boldsymbol{x},t')}{\delta r(\boldsymbol{x}',s)}$

$$\begin{aligned} A^\ell(\boldsymbol{x}, \boldsymbol{x}'; t, s) = \frac{1}{\sqrt{L}} \sum_{k=1}^{\ell} \int d\boldsymbol{x}'' dt' A^k(\boldsymbol{x}, \boldsymbol{x}''; t, t') \left\langle \dot{\phi}(h^\ell(\boldsymbol{x}, t)) \frac{\delta g^{k+1}(\boldsymbol{x}'', t')}{\delta r^\ell(\boldsymbol{x}', s)} \right\rangle \\ + \frac{\eta_0\gamma_0}{L} \sum_{k=0}^{\ell-1} \mathbb{E}_{\boldsymbol{x}'} \int_0^t dt' \Delta(\boldsymbol{x}', s) \Phi^k(\boldsymbol{x}, \boldsymbol{x}'; t, s) \left\langle \dot{\phi}(h^\ell(\boldsymbol{x}, t)) \frac{\delta g^k(\boldsymbol{x}'', t')}{\delta r^\ell(\boldsymbol{x}', s)} \right\rangle \end{aligned} \tag{38}$$

Next, we use the fact that $\frac{\delta g^k(\boldsymbol{x}'',t')}{\delta r^\ell(\boldsymbol{x}',s)} \sim \mathcal{O}(L^{-1/2})$. Therefore, the second sum (the one which depends on $\eta_0\gamma_0 L^{-1}$) scales as $\mathcal{O}(\eta_0\gamma_0 L^{-1/2})$. Solving the A^ℓ recursion iteratively from the first layer, we arrive at the conclusion that

$$A^\ell(\boldsymbol{x}, \boldsymbol{x}'; t, s) \sim \mathcal{O}(\eta_0\gamma_0 L^{-1/2}). \tag{39}$$

Indeed, as a sanity check, plugging this in gives a consistent solution to the above equation. Repeating an identical argument for the backward pass reveals an identical scaling for B^ℓ with depth L

$$B^\ell(\boldsymbol{x}, \boldsymbol{x}'; t, s) \sim \mathcal{O}(\eta_0\gamma_0 L^{-1/2}) \tag{40}$$

We therefore need to introduce a rescaled version of the response functions

$$A^\ell(\boldsymbol{x}, \boldsymbol{x}', t, s) \leftarrow \frac{\sqrt{L}}{\eta_0\gamma_0} A^\ell(\boldsymbol{x}, \boldsymbol{x}', t, s) \tag{41}$$

$$B^\ell(\boldsymbol{x}, \boldsymbol{x}', t, s) \leftarrow \frac{\sqrt{L}}{\eta_0\gamma_0} B^\ell(\boldsymbol{x}, \boldsymbol{x}', t, s) \tag{42}$$

After this rescaling, we have

$$A^\ell(\boldsymbol{x}, \boldsymbol{x}', t, s) = \frac{\sqrt{L}}{\eta_0\gamma_0} \left\langle \frac{\delta\phi(h^\ell(\boldsymbol{x}, t))}{\delta r^\ell(\boldsymbol{x}', s)} \right\rangle \tag{43}$$

To calculate this rescaled response function, we can derive a closed set of equations for all of the cross layer correlators

$$\begin{aligned} \frac{\delta h^\ell(\boldsymbol{x}, t)}{\delta r^{\ell'}(\boldsymbol{x}', s)} &= \frac{\eta_0\gamma_0}{L} \sum_{k=1}^{\ell} \int d\boldsymbol{x}'' ds\, C_h^k(\boldsymbol{x}, \boldsymbol{x}'', t, s) \frac{\delta g^k(\boldsymbol{x}'', s)}{\delta r^{\ell'}(\boldsymbol{x}', s)} \\ \frac{\delta g^\ell(\boldsymbol{x}, t)}{\delta r^{\ell'}(\boldsymbol{x}', s)} &= \frac{1}{\sqrt{L}} \Theta(\ell' - \ell) \delta(\boldsymbol{x} - \boldsymbol{x}') \delta(t - s) \dot{\phi}(h^{\ell'}(\boldsymbol{x}, t)) \\ &\quad + \frac{1}{\sqrt{L}} \sum_{k=\ell}^{L} r^k(\boldsymbol{x}, t) \frac{\delta\dot{\phi}(h^k(\boldsymbol{x}, t))}{\delta r^{\ell'}(\boldsymbol{x}', s)} \\ &\quad + \frac{\eta_0\gamma_0}{L} \sum_{k=\ell}^{L} \dot{\phi}(h^k(\boldsymbol{x}, t)) \int d\boldsymbol{x}'' ds C_g^k(\boldsymbol{x}, \boldsymbol{x}'', t, s) \frac{\delta\phi(h^k(\boldsymbol{x}'', s))}{\delta r^{\ell'}(\boldsymbol{x}', s)} \\ &\quad + \frac{\eta_0\gamma_0}{L} \sum_{k=\ell}^{L} \frac{\delta\dot{\phi}(h^k(\boldsymbol{x}, t))}{\delta r^{\ell'}(\boldsymbol{x}', s)} \int d\boldsymbol{x}'' ds C_g^k(\boldsymbol{x}, \boldsymbol{x}'', t, s) \phi(h^k(\boldsymbol{x}'', s)) \end{aligned} \tag{44}$$

where we introduced the shorthand

$$\begin{aligned} C_h^\ell(\boldsymbol{x}, \boldsymbol{x}', t, s) &= A^{\ell-1}(\boldsymbol{x}, \boldsymbol{x}', t, s) + \Phi^{\ell-1}(\boldsymbol{x}, \boldsymbol{x}', t, s)\Delta(\boldsymbol{x}', s)p(\boldsymbol{x}') \\ C_h^\ell(\boldsymbol{x}, \boldsymbol{x}', t, s) &= B^\ell(\boldsymbol{x}, \boldsymbol{x}', t, s) + G^{\ell+1}(\boldsymbol{x}, \boldsymbol{x}', t, s)\Delta(\boldsymbol{x}', s)p(\boldsymbol{x}') \end{aligned} \tag{45}$$

Similar equations exist for all derivatives with respect to $u^\ell(\boldsymbol{x}, t)$. Once the above equations are solved, then one can compute the response functions A^ℓ, B^ℓ by averaging the necessary field derivatives.

E.3 Continuum Layer-time Limit at Infinite Depth

In this section, we argue the infinite depth limit can be viewed as a continuum limiting process defined by ODEs for the kernels and SDEs for the preactivation and gradient fields for a layertime $\tau = \ell/L$. We start by examining the asymptotic structure of the preactivation h^ℓ dynamics

$$h^\ell(\boldsymbol{x};t) \sim h^1(\boldsymbol{x};t) + \frac{1}{\sqrt{L}}\sum_{k=1}^{\ell} u^k(\boldsymbol{x};t) + \frac{\eta_0\gamma_0}{L}\sum_{k=0}^{\ell-1}\int_0^t ds\, \mathbb{E}_{\boldsymbol{x}'} C_h^k(\boldsymbol{x},\boldsymbol{x}';t,s)\, g^{k+1}(\boldsymbol{x}',s)\Delta(\boldsymbol{x}';s) \tag{46}$$

The Gaussian source variables u^k are all uncorrelated across layers. So the first sum is also a mean zero Gaussian with variance

$$\left\langle \left(\frac{1}{\sqrt{L}}\sum_{k=1}^{\ell} u^k(\boldsymbol{x},t)\right)\left(\frac{1}{\sqrt{L}}\sum_{k=1}^{\ell} u^k(\boldsymbol{x}',s)\right)\right\rangle = \frac{1}{L}\sum_{k=1}^{\ell}\Phi^{k-1}(\boldsymbol{x},\boldsymbol{x}';t,s) \sim \mathcal{O}(1) \tag{47}$$

At large depth L, this term will behave as integrated Brownian motion since each layer's contribution is independent and order $\sqrt{d\tau} \sim \frac{1}{\sqrt{L}}$. Next, we reason about the scale of feature learning update

$$\frac{\eta_0^2\gamma_0^2}{L^2}\sum_{k=0}^{\ell-1}\sum_{k'=0}^{\ell-1}\int_0^t ds'\int_0^t ds\, \mathbb{E}_{\boldsymbol{x}',\boldsymbol{x}''}\, \Delta(\boldsymbol{x}';s)\Delta(\boldsymbol{x}'';s') \times C_h^k(\boldsymbol{x},\boldsymbol{x}';t,s)\, C_h^{k'}(\boldsymbol{x},\boldsymbol{x}';t,s)\, \left\langle g^{k+1}(\boldsymbol{x}',s) g^{k'+1}(\boldsymbol{x}'',s')\right\rangle \sim \mathcal{O}(1) \tag{48}$$

The above double sum is $\mathcal{O}(1)$ because we have $\left\langle g^k g^{k'}\right\rangle \sim \mathcal{O}(1)$ for all k, k'. This fact arises from the skip connections, which generate *long-range correlations* across layers. This behavior is very different from non-residual networks where g^ℓ and $g^{\ell'}$ are independent for all $\ell \neq \ell'$ [6].

To take the continuum limit, we define functions of a continuous layer time variable $\tau \in [0,1]$,

$$h(\tau,\boldsymbol{x},t) = h^\ell(\boldsymbol{x},t)|_{\ell=\tau L}\ ,\ g(\tau,\boldsymbol{x},t) = g^\ell(\boldsymbol{x},t)|_{\ell=\tau L}$$
$$\Phi(\tau;\boldsymbol{x},\boldsymbol{x}';t,s) = \Phi^\ell(\boldsymbol{x},\boldsymbol{x}';t,s)|_{\ell=\tau L}\ ,\ G(\tau;\boldsymbol{x},\boldsymbol{x}';t,s) = G^\ell(\boldsymbol{x},\boldsymbol{x}';t,s)|_{\ell=\tau L}. \tag{49}$$

The key idea when taking the continuum limit is the following relations based on a notion an infinitesimal layer increment $d\tau \sim \frac{1}{L}$. Let F^ℓ be a deterministic function of the layer index and let $F(\tau)$ be its continuous analogue which satisfies $F(\tau) = F^\ell|_{\ell=\tau L}$. Then,

$$\lim_{L\to\infty}\frac{1}{L}\sum_{\ell=1}^{\tau L} F^\ell \to \int_0^\tau d\tau' F(\tau)$$
$$\lim_{L\to\infty} L\left[F^{\tau L+1} - F^{\tau L}\right] \to \frac{\partial}{\partial\tau}F(\tau). \tag{50}$$

We will use these relations when deriving continuum limits of the DMFT process. We note that standard error analysis for Euler discretizations can be used to argue that finite depth averages converge to the τ integral at rate

$$\left|\frac{1}{L}\sum_{\ell=1}^{\tau L} F^\ell - \int_0^\tau d\tau' F(\tau')\right| \sim \mathcal{O}(L^{-1}) \tag{51}$$

This is merely the approximation error of a Riemann sum which is proportional to the step size [47], which in this case is $1/L$. This idea can be used to generate a convergence rate of the NTK and the network output predictions at large depth as we discuss in I.

E.4 Preactivation Dynamics

As discussed in the main text, the large L limit for the dynamics of h^ℓ can be viewed as a stochastic integral over τ and over Brownian motion for the u^k sum

$$\begin{aligned} h(\tau;\boldsymbol{x},t) = \; & h(0;\boldsymbol{x},t) + \int_0^\tau du(\tau',\boldsymbol{x},t) \\ & + \eta_0\gamma_0 \int d\boldsymbol{x}' \int_0^t ds \int_0^\tau d\tau' \left[\Phi(\tau';\boldsymbol{x},\boldsymbol{x}';t,s)\Delta(\boldsymbol{x}';s)p(\boldsymbol{x}') + A(\tau',\boldsymbol{x},\boldsymbol{x}';t,s)\right] g(\tau',\boldsymbol{x}',s) \end{aligned} \tag{52}$$

where du is Brownian motion term with covariance structure

$$\langle du(\tau,\boldsymbol{x},t) du(\tau',\boldsymbol{x}',s)\rangle = \delta(\tau-\tau')\Phi(\tau,\boldsymbol{x},\boldsymbol{x}',t,s)d\tau. \tag{53}$$

Similarly, the gradient fields have an analogous structure

$$\begin{aligned} g(\tau;\boldsymbol{x},t) = g(\tau;\boldsymbol{x},t) + & \int_\tau^1 dr(\tau',\boldsymbol{x},t)\dot\phi(h(\tau',\boldsymbol{x},t)) \\ & + \eta_0\gamma_0 \mathbb{E}_{\boldsymbol{x}'} \int_0^t ds \int_\tau^1 d\tau' \dot\phi(h(\tau',\boldsymbol{x},t)) G(\tau';\boldsymbol{x},\boldsymbol{x}';t,s)\Delta(\boldsymbol{x}';s)\phi(h(\tau';\boldsymbol{x}',s)) \\ & + \eta_0\gamma_0 \int d\boldsymbol{x}' \int_0^t ds \int_\tau^1 d\tau' \dot\phi(h(\tau',\boldsymbol{x},t)) B(\tau',\boldsymbol{x},\boldsymbol{x}';t,s)\phi(h(\tau';\boldsymbol{x}',s)). \end{aligned} \tag{54}$$

where dr is Brownian motion with covariance structure

$$\langle dr(\tau,\boldsymbol{x},t) dr(\tau',\boldsymbol{x}',s)\rangle = \delta(\tau-\tau')G(\tau,\boldsymbol{x},\boldsymbol{x}',t,s)d\tau. \tag{55}$$

We note that, due to the nonlinearities, the h, g fields are *non-Gaussian* at finite γ_0 as in the case of mean field networks [6]. We note that an alternative to the above integral formulation is a stochastic differential equations

$$\begin{aligned} dh(\tau,\boldsymbol{x},t) = \; & du(\tau,\boldsymbol{x},t) \\ & + d\tau\, \eta_0\gamma_0 \int d\boldsymbol{x}' \int_0^t ds \left[\Phi(\tau;\boldsymbol{x},\boldsymbol{x}';t,s)\Delta(\boldsymbol{x}';s)p(\boldsymbol{x}') + A(\tau;\boldsymbol{x},\boldsymbol{x}';t,s)\right] g(\tau,\boldsymbol{x}',s) \end{aligned} \tag{56}$$

The $\gamma_0 \to 0$ limit of the h equation at $t, s = 0$ coincides with the SDE derived at initialization by [22, 23]. However, we now see that feature learning term (the term which depends on γ_0) adds a drift term to the layerwise dynamics compared to the diffusion term from the initial weights.

E.5 Kernel Layerwise ODEs in τ in Feature Learning Regime

We can start from the stochastic difference equation

$$\begin{aligned} dh(\tau,\boldsymbol{x},t) &= du(\tau,\boldsymbol{x}',t) + d\tau\, \eta_0\gamma_0 \int d\boldsymbol{x}' \int_0^t ds\, C_h(\tau,\boldsymbol{x},\boldsymbol{x}',t,s) g(\tau,\boldsymbol{x}',s) \\ C_h(\tau,\boldsymbol{x},\boldsymbol{x}',t,s) &= A(\tau,\boldsymbol{x},\boldsymbol{x}',t,s) + \Phi(\tau,\boldsymbol{x},\boldsymbol{x}',t,s)\Delta(\boldsymbol{x}',s)p(\boldsymbol{x}'). \end{aligned} \tag{57}$$

and derive an ODE for the feature kernel at layer-time τ. Applying Ito's lemma [28], we obtain the following feature kernel dynamics

$$\begin{aligned} & \frac{\partial}{\partial\tau}\Phi(\tau,\boldsymbol{x},\boldsymbol{x}',t,s) = \\ & \frac{1}{2}\int d\tilde{\boldsymbol{x}} d\tilde{\boldsymbol{x}}' dt' ds' \Phi(\tau,\tilde{\boldsymbol{x}},\tilde{\boldsymbol{x}}',t',s') \left\langle \frac{\partial^2}{\partial h(\tau,\tilde{\boldsymbol{x}},t')\partial h(\tau,\tilde{\boldsymbol{x}}',s')}[\phi(h(\tau,\boldsymbol{x},t))\phi(h(\tau,\boldsymbol{x}',s))] \right\rangle \\ & + \eta_0\gamma_0 \int d\boldsymbol{x}'' dt' C_h(\tau,\boldsymbol{x},\boldsymbol{x}'',t') \left\langle g(\tau,\boldsymbol{x}'',t')\dot\phi(h(\tau,\boldsymbol{x},t))\phi(h(\tau,\boldsymbol{x}',s)) \right\rangle \\ & + \eta_0\gamma_0 \int d\boldsymbol{x}'' ds' C_h(\tau,\boldsymbol{x}',\boldsymbol{x}'',s') \left\langle g(\tau,\boldsymbol{x}'',s')\phi(h(\tau,\boldsymbol{x},t))\dot\phi(h(\tau,\boldsymbol{x}',s)) \right\rangle \end{aligned} \tag{58}$$

The first term on the right hand side comes from signal propagation, which persists even in the lazy limit. The second set of terms come from feature learning corrections. This equation should be integrated from $\tau = 0$ to $\tau = 1$. Similarly, there is a backward pass ODE for the G kernel

$$\begin{aligned}
&\frac{\partial}{\partial \tau} G(\tau, \boldsymbol{x}, \boldsymbol{x}', t, s) = -\left\langle \dot{\phi}(h(\tau, \boldsymbol{x}, t)) \dot{\phi}(h(\tau, \boldsymbol{x}', s)) \right\rangle G(\tau, \boldsymbol{x}, \boldsymbol{x}', t, s) \\
&- \eta_0 \gamma_0 \int d\boldsymbol{x}'' \int_0^{t'} dt' C_g(\tau, \boldsymbol{x}, \boldsymbol{x}'', t, t') \left\langle \dot{\phi}(h(\tau, \boldsymbol{x}, t)) \phi(h(\tau, \boldsymbol{x}', s)) g(\tau, \boldsymbol{x}'', t') \right\rangle \\
&- \eta_0 \gamma_0 \int d\boldsymbol{x}'' \int_0^{s'} ds' C_g(\tau, \boldsymbol{x}, \boldsymbol{x}'', t, t') \left\langle \phi(h(\tau, \boldsymbol{x}, t)) \dot{\phi}(h(\tau, \boldsymbol{x}', s)) g(\tau, \boldsymbol{x}'', s') \right\rangle \\
&C_g(\tau, \boldsymbol{x}, \boldsymbol{x}', t, s) = B(\tau, \boldsymbol{x}, \boldsymbol{x}', t, s) + G(\tau, \boldsymbol{x}, \boldsymbol{x}', t, s) \Delta(\boldsymbol{x}', s) p(\boldsymbol{x}')
\end{aligned} \tag{59}$$

which should be integrated from $\tau = 1$ to $\tau = 0$.

We note again, that convergence to the ODE will occur at rate $\mathcal{O}(L^{-1})$ in absolute error.

E.6 Response Function Equations in the Continuum Limit

The response functions can also be determined in the $L \to \infty$ limit using the following relations

$$A(\tau, \boldsymbol{x}, \boldsymbol{x}', t, s) = \frac{1}{\eta_0 \gamma_0} \left\langle \frac{\delta \phi(h(\tau, \boldsymbol{x}, t))}{\delta \, dr(\tau, \boldsymbol{x}', s)} \right\rangle \ , \ B(\tau, \boldsymbol{x}, \boldsymbol{x}', t, s) = \frac{1}{\eta_0 \gamma_0} \left\langle \frac{\delta g(\tau, \boldsymbol{x}, t)}{\delta \, du(\tau, \boldsymbol{x}', s)} \right\rangle \tag{60}$$

The reason that the variation is computed with respect to the differential Brownian motion du, dr is to match the scale of the increment $\frac{1}{\sqrt{L}}$, which appears in the finite L response functions $A^\ell = \frac{\sqrt{L}}{\eta_0 \gamma_0} \left\langle \frac{\delta \phi(h^\ell)}{\delta r^\ell} \right\rangle$. Using our continuum limit equations, we obtain the following field derivatives

$$\begin{aligned}
\frac{\delta h(\tau, \boldsymbol{x}, t)}{\delta du(\tau', \boldsymbol{x}', s)} &= \Theta(\tau - \tau') \delta(\boldsymbol{x} - \boldsymbol{x}') \delta(t - s) \\
&\quad + \eta_0 \gamma_0 \int d\tau'' \int d\boldsymbol{x}'' \int dt' C_h(\tau'', \boldsymbol{x}, \boldsymbol{x}'', t, t') \frac{\delta g(\tau'', \boldsymbol{x}'', t')}{\delta du(\tau', \boldsymbol{x}', s)} \\
\frac{\delta g(\tau, \boldsymbol{x}, t)}{\delta du(\tau', \boldsymbol{x}', s)} &= \eta_0 \gamma_0 \int d\tau'' \dot{\phi}(h(\tau'', \boldsymbol{x}, t)) \int d\boldsymbol{x}'' \int dt' C_g(\tau'', \boldsymbol{x}, \boldsymbol{x}'', t, t') \frac{\delta \phi(h(\tau'', \boldsymbol{x}'', t'))}{\delta du(\tau', \boldsymbol{x}', s)} \\
\frac{\delta h(\tau, \boldsymbol{x}, t)}{\delta dr(\tau', \boldsymbol{x}', s)} &= \eta_0 \gamma_0 \int d\tau'' \int d\boldsymbol{x}'' \int dt' C_h(\tau'', \boldsymbol{x}, \boldsymbol{x}'', t, t') \frac{\delta g(\tau'', \boldsymbol{x}'', t')}{\delta dr(\tau', \boldsymbol{x}', s)} \\
\frac{\delta g(\tau, \boldsymbol{x}, t)}{\delta dr(\tau', \boldsymbol{x}', s)} &= \Theta(\tau' - \tau) \delta(\boldsymbol{x} - \boldsymbol{x}') \delta(t - s) \ \dot{\phi}(h(\tau, \boldsymbol{x}, t)) \\
&\quad + \eta_0 \gamma_0 \int d\tau'' \dot{\phi}(h(\tau'', \boldsymbol{x}, t)) \int d\boldsymbol{x}'' \int dt' C_h(\tau'', \boldsymbol{x}, \boldsymbol{x}'', t, t') \frac{\delta \phi(h(\tau'', \boldsymbol{x}'', t'))}{\delta dr(\tau', \boldsymbol{x}', s)}
\end{aligned} \tag{61}$$

Once these recursions are solved, the response functions can be computed by averaging the necessary field derivatives at each layer time τ

$$A(\tau, \boldsymbol{x}, \boldsymbol{x}', t, s) = \left\langle \frac{\delta \phi(h(\tau, \boldsymbol{x}, t))}{\delta dr(\tau, \boldsymbol{x}', s)} \right\rangle \ , \ B(\tau, \boldsymbol{x}, \boldsymbol{x}', t, s) = \left\langle \frac{\delta g(\tau, \boldsymbol{x}, t)}{\delta du(\tau, \boldsymbol{x}', s)} \right\rangle . \tag{62}$$

E.7 Final Result: Infinite Width and Depth Limit

In this section we combine the results obtained in the preceeding sections to state our main theoretical result which is the infinite width and depth limit of the learning dynamics in our ResNet. Below we state the result for gradient flow, but minor modifications can also characterize discrete time SGD, momentum, weight decay etc as we discuss in Appendices K.

Proposition 2 *Consider the $N, L \to \infty$ limit of the ResNet μP-$\frac{1}{\sqrt{L}}$ model presented in the main text trained with gradient flow on data distribution $p(\boldsymbol{x})$. The preactivation $h(\tau, \boldsymbol{x}, t)$ and gradient*

$g(\tau, \boldsymbol{x}, t)$ stochastic fields obey the equations

$$h(\tau, \boldsymbol{x}, t) = h(0, \boldsymbol{x}, t) + \int_0^\tau du(\tau', \boldsymbol{x}, t) + \eta_0 \gamma_0 \int_0^\tau d\tau' \int d\boldsymbol{x}' \int_0^t ds\, C_h(\tau', \boldsymbol{x}, \boldsymbol{x}', t, s) g(\tau', \boldsymbol{x}', s)$$
$$g(\tau, \boldsymbol{x}, t) = g(1, \boldsymbol{x}, t) + \int_\tau^1 \dot{\phi}(h(\tau', \boldsymbol{x}, t)) dr(\tau', \boldsymbol{x}, t)$$
$$+ \eta_0 \gamma_0 \int_0^\tau d\tau' \dot{\phi}(h(\tau', \boldsymbol{x}, t)) \int d\boldsymbol{x}' \int_0^t ds\, C_g(\tau', \boldsymbol{x}, \boldsymbol{x}', t, s) \phi(h(\tau', \boldsymbol{x}', s)) \tag{63}$$

where the zero mean Brownian motions $du(\tau, \boldsymbol{x}, t), dr(\tau, \boldsymbol{x}, t)$ have the following covariance structure

$$\langle du(\tau, \boldsymbol{x}, t) du(\tau', \boldsymbol{x}', s) \rangle = \Phi(\tau, \boldsymbol{x}, \boldsymbol{x}', t, s) \delta(\tau - \tau') d\tau$$
$$\langle dr(\tau, \boldsymbol{x}, t) dr(\tau', \boldsymbol{x}', s) \rangle = G(\tau, \boldsymbol{x}, \boldsymbol{x}', t, s) \delta(\tau - \tau') d\tau, \tag{64}$$

the deterministic kernels C_h and C_g have the form

$$C_h(\tau, \boldsymbol{x}, \boldsymbol{x}', t, s) = \Phi(\tau, \boldsymbol{x}, \boldsymbol{x}', t, s) \Delta(\boldsymbol{x}', s) p(\boldsymbol{x}') + A(\tau, \boldsymbol{x}, \boldsymbol{x}', t, s)$$
$$C_g(\tau, \boldsymbol{x}, \boldsymbol{x}', t, s) = G(\tau, \boldsymbol{x}, \boldsymbol{x}', t, s) \Delta(\boldsymbol{x}', s) p(\boldsymbol{x}') + B(\tau, \boldsymbol{x}, \boldsymbol{x}', t, s) \tag{65}$$

where the kernels Φ, G and response functions A, B are computed as averages over the h, g stochastic process

$$\Phi(\tau, \boldsymbol{x}, \boldsymbol{x}', t, s) = \langle \phi(h(\tau, \boldsymbol{x}, t)) \phi(h(\tau, \boldsymbol{x}', s)) \rangle \ , \ G(\tau, \boldsymbol{x}, \boldsymbol{x}', t, s) = \langle g(\tau, \boldsymbol{x}, t) g(\tau, \boldsymbol{x}', s) \rangle$$
$$A(\tau, \boldsymbol{x}, \boldsymbol{x}', t, s) = \frac{1}{\eta_0 \gamma_0} \left\langle \frac{\delta \phi(h(\tau, \boldsymbol{x}, t))}{\delta dr(\tau, \boldsymbol{x}', s)} \right\rangle \ , \ B(\tau, \boldsymbol{x}, \boldsymbol{x}', t, s) = \frac{1}{\eta_0 \gamma_0} \left\langle \frac{\delta g(\tau, \boldsymbol{x}, t)}{\delta du(\tau, \boldsymbol{x}', s)} \right\rangle . \tag{66}$$

From the feature and gradient kernels, we can compute the dynamical NTK and the dynamics of network predictions

$$K(\boldsymbol{x}, \boldsymbol{x}', t) = \int_0^1 d\tau\, G(\tau, \boldsymbol{x}, \boldsymbol{x}', t, t) \Phi(\tau, \boldsymbol{x}, \boldsymbol{x}', t, t)$$
$$\frac{d}{dt} f(\boldsymbol{x}, t) = \eta_0 \int d\boldsymbol{x}' p(\boldsymbol{x}') K(\boldsymbol{x}, \boldsymbol{x}', t) \Delta(\boldsymbol{x}', t) \ , \ \Delta(\boldsymbol{x}, t) = -\frac{\partial \mathcal{L}}{\partial f(\boldsymbol{x}, t)}. \tag{67}$$

This is a closed system of equations relating preactiation and gradient marginals, the kernel dynamics and the network prediction dynamics. In this limit, the kernels, response functions and network outputs are all deterministic.

F Infinite Depth then Infinite Width Limit

In this section, we compute the limiting process in the other direction, by directly working with the distribution of order parameters $\boldsymbol{q}$ at finite N. We show that at large depth L the DMFT action converges to a limiting object $\mathcal{S}_\infty(\boldsymbol{q})$. We the show that the large N limit of the distribution recovers the same saddle point derived in Appendix E for the large N then large L limit. Thus, this section constitutes a theoretical argument that the width and depth limits commute for the whole course of training dynamics. The derivations are performed at the level of rigor of physics, and we leave open for future work a rigorous result about commutativity of limits throughout training.

F.1 Large L Limit at Fixed N

We derived the finite N, L distribution of order parameters in Appendix Section D. The result was that the log-density of the order parameters $\boldsymbol{q}$ over random initialization of network weights was given by the DMFT action $\ln p(\boldsymbol{q}) \propto N \mathcal{S}_L(\boldsymbol{q})$. Suppressing unnecessary (for the purposes of this section) sample and time indices, the action at depth $\mathcal{S}_L$ had the form

$$\mathcal{S}_L = \sum_\ell \left[\Phi^\ell \hat{\Phi}^\ell + G^\ell \hat{G}^\ell - A^\ell B^\ell \right] + \mathcal{Z}[\boldsymbol{q}] \tag{68}$$

where the single site density $\mathcal{Z}$ has the form

$$\mathcal{Z} = \int \prod_\ell dh^\ell d\hat{h}^\ell dg^\ell d\hat{g}^\ell$$
$$\mathbb{E}_{\{u^\ell, r^\ell\}} \exp\left(i \sum_\ell \hat{h}^\ell \left(h^{\ell+1} - h^\ell - \frac{1}{\sqrt{L}} u^\ell - \frac{1}{\sqrt{L}} A^\ell g^{\ell+1} - \frac{\eta\gamma}{L} \Phi^\ell \Delta g^{\ell+1}\right)\right)$$
$$\exp\left(i \sum_\ell \hat{g}^\ell \left(g^\ell - g^{\ell+1} - \frac{1}{\sqrt{L}} r^\ell \dot{\phi}(h^\ell) - \frac{1}{\sqrt{L}} \dot{\phi}(h^\ell) B^\ell \phi(h^\ell) - \frac{\eta_0\gamma_0}{L} \dot{\phi}(h^\ell) G^{\ell+1} \Delta\phi(h^\ell)\right)\right)$$
$$\exp\left(-\sum_\ell \phi(h^\ell) \hat{\Phi}^\ell \phi(h^\ell) - \sum_\ell g^\ell \hat{G}^\ell g^\ell\right) \tag{69}$$

where the averages are over u^ℓ, r^ℓ which are Gaussians with covariances $\Phi^{\ell-1}$ and $G^{\ell+1}$ respectively. In the above equation we work with h, g directly rather than working with the intermediate fields χ, ξ which only depend on the initialization.

To obtain a proper large depth limit, we rescale the conjugate order parameters as

$$\hat{\Phi}^\ell \to \frac{1}{L} \hat{\Phi}^\ell \ , \ \hat{G}^\ell \to \frac{1}{L} \hat{G}^\ell \ , \ A^\ell \to \frac{\eta_0\gamma_0}{\sqrt{L}} A^\ell \ , \ B^\ell \to \frac{\eta_0\gamma_0}{\sqrt{L}} B^\ell \tag{70}$$

We note that such a rescaling will not alter the distribution of real network observables $\{\Phi, G, K, f, ...\}$ [6, 7]. After this rescaling, we take $L \to \infty$ which gives a continuum limit of the resulting layer sums, we obtain the following simplified infinite depth action in terms of the kernels and features at layer time $\tau \in [0, 1]$

$$\mathcal{S}_\infty = \int d\tau [\Phi(\tau)\hat{\Phi}(\tau) + G(\tau)\hat{G}(\tau) - \eta_0^2\gamma_0^2 A(\tau)B(\tau)] + \ln \mathcal{Z}$$
$$\mathcal{Z} = \int \mathcal{D}\hat{h}\mathcal{D}\hat{g}\mathcal{D}h\mathcal{D}g \, \mathbb{E}_{u(\tau), r(\tau)} \exp\left(i \int \hat{h}(\tau)[dh(\tau) - du(\tau) - \eta_0\gamma_0 C_h(\tau) g(\tau) d\tau]\right)$$
$$\exp\left(i \int \hat{g}(\tau)[-dg(\tau) - dr(\tau)\dot{\phi}(h(\tau)) - \eta_0\gamma_0 \dot{\phi}(h(\tau)) C_g(\tau) \phi(h(\tau)) d\tau]\right)$$
$$\exp\left(-\int d\tau [\phi(h(\tau))\hat{\Phi}(\tau)\phi(h(\tau)) + g(\tau)\hat{G}(\tau)g(\tau)]\right). \tag{71}$$

where C_h, C_g are given in the main text and the $du(\tau), dr(\tau)$ are Brownian motion processes over layer time τ. We therefore find that the distribution of order parameters at finite width N and infinite depth $L \to \infty$ is

$$\lim_{L\to\infty} p_L(\boldsymbol{q}) = p_\infty(\boldsymbol{q}) \propto \exp\left(N\mathcal{S}_\infty(\boldsymbol{q})\right) \tag{72}$$

This constitutes taking the infinite depth limit $L \to \infty$ at fixed N. At this stage, the order parameters do not obey a deterministic set of equations, but are samples from the above density. In the next section, we study infinite width of this density and find that we recover the same result that we derived previously in the other direction.

F.2 Saddle Point Equations on Infinite Depth Action

This section considers the large N limit of the infinite depth stochastic process. In particular, we evaluate the saddle point of $\mathcal{S}_\infty$ and show that it recovers the same limit as we obtained by first taking

a saddle point and then taking the large L limit. This shows that the limits commute.

$$\begin{aligned}
\frac{\delta \mathcal{S}_\infty}{\delta \hat{\Phi}(\tau)} &= \Phi(\tau) - \langle \phi(h(\tau))\phi(h(\tau))\rangle = 0 \\
\frac{\delta \mathcal{S}_\infty}{\delta \hat{G}(\tau)} &= G(\tau) - \langle g(\tau) g(\tau)\rangle = 0 \\
\frac{\delta \mathcal{S}_\infty}{\delta A(\tau)} &= -\eta_0^2 \gamma_0^2 B(\tau) - i\eta_0\gamma_0 \left\langle \hat{h}(\tau) g(\tau) \right\rangle = 0 \\
\frac{\delta \mathcal{S}_\infty}{\delta B(\tau)} &= -\eta_0^2 \gamma_0^2 A(\tau) - i\eta_0\gamma_0 \left\langle \hat{g}(\tau)\dot{\phi}(h(\tau))\phi(h(\tau)) \right\rangle = 0
\end{aligned} \tag{73}$$

For the last two expressions, we recognizing that

$$\begin{aligned}
&\frac{\delta}{\delta du(\tau')} \exp\left(-i\int \hat{h}(\tau) du(\tau)\right) = -i\hat{h}(\tau)\exp\left(-i\int \hat{h}(\tau) du(\tau)\right) \\
&\frac{\delta}{\delta dr(\tau')} \exp\left(-i\int \hat{g}(\tau)\dot{\phi}(h(\tau)) dr(\tau)\right) = -i\hat{g}(\tau)\dot{\phi}(h(\tau))\exp\left(-i\int \hat{g}(\tau)\dot{\phi}(h(\tau)) dr(\tau)\right)
\end{aligned} \tag{74}$$

We can thus express the saddle point equations in terms of the left hand side derivatives with respect to Gaussian sources. Performing integration by parts, we arrive at the usual response function formulas

$$A(\tau) = \eta_0^{-1}\gamma_0^{-1}\left\langle \frac{\delta\phi(h(\tau))}{\delta dr(\tau))} \right\rangle \ , \ B(\tau) = \eta_0^{-1}\gamma_0^{-1}\left\langle \frac{\delta g(\tau)}{\delta du(\tau)} \right\rangle \tag{75}$$

Lastly, we can integrate over $\hat{h}(\tau)$ and $\hat{g}(\tau)$ which give Dirac-Delta functions that define the h and g stochastic processes

$$dh(\tau) = du(\tau) + \eta_0\gamma_0 C_h(\tau) g(\tau) \ , \ dg(\tau) = -dr(\tau)\dot{\phi}(h(\tau)) - \eta_0\gamma_0\dot{\phi}(h(\tau)) C_g(\tau)\phi(h(\tau)) d\tau \tag{76}$$

The rest of the indices (over samples and training times) can be incorporated into the above argument and the result is easily verified to match that derived in Appendix E from the large width then large depth limit. This exercise thus verifies that the large width and large depth limit continue to commute throughout training, extending the commutativity at initialization proven by [23].

G Exactly Solveable Cases

In this section, we give exact solutions to the dynamics which do not require any non-Gaussian integrals. The two cases where this can be achieved are

1. the lazy limit $\gamma_0 \to 0$ of network training where the features are frozen
2. networks with linear activations, which preserve Gaussianity of h, g at any γ_0.

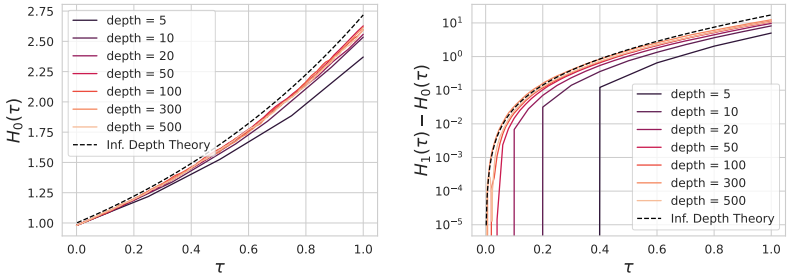
We discuss these two limits in great detail below.

G.1 Lazy Limit

In the lazy limit $\gamma_0 \to 0$, the flow for the preactivations $h(\tau, \boldsymbol{x}, t) = h(\tau, \boldsymbol{x})$ and the gradient fields $g^\ell(\tau, \boldsymbol{x}, t) = g^\ell(\tau, \boldsymbol{x})$ are constant over training time. Similarly, the kernels Φ^ℓ, G^ℓ are also constant throughout the dynamics. We thus have the following forward and backward dynamics

$$h(\tau, \boldsymbol{x}) = h(0, \boldsymbol{x}) + \int_0^\tau du(\tau', \boldsymbol{x}) \tag{77}$$

$$g(\tau, \boldsymbol{x}) = g(1, \boldsymbol{x}) + \int_\tau^1 \dot{\phi}(h(\tau', \boldsymbol{x})) dr(\tau', \boldsymbol{x}) \tag{78}$$



(a) Kernels at initialization (b) Kernel Updates after 1 Step

Figure G.1: We plot the feature kernels $H(\tau) = H^\ell|_{\ell=\tau L}$ in width $N = 1000$ residual linear networks before and after taking a gradient step in our parameterization. The kernel profiles at initialization $H_0(\tau)$ and the kernel changes $H_1(\tau) - H_0(\tau)$ converge for large L. We compare to the theoretical result expected from the $L \to \infty$ limit of the DMFT equations (dashed black).

We note that the above equations indicate that $h(\tau, \boldsymbol{x})$ are all jointly Gaussian with zero mean. Since h is Gaussian, it makes sense to characterize its covariance $H(\tau, \boldsymbol{x}, \boldsymbol{x}') = \langle h(\tau, \boldsymbol{x}) h(\tau, \boldsymbol{x}') \rangle$ at layer-time τ. This object satisfies the differential equation

$$\begin{aligned} \frac{\partial}{\partial \tau} H(\tau, \boldsymbol{x}, \boldsymbol{x}') &= \Phi(\tau, \boldsymbol{x}, \boldsymbol{x}') \\ \Phi(\tau, \boldsymbol{x}, \boldsymbol{x}') &= \langle \phi(h(\tau, \boldsymbol{x})) \phi(h(\tau, \boldsymbol{x}')) \rangle \\ h(\tau, \boldsymbol{x}, \boldsymbol{x}') &\sim \mathcal{N}(0, \boldsymbol{H}(\tau)) \end{aligned} \tag{79}$$

Since $h(\tau, \boldsymbol{x})$ are zero-mean jointly Gaussian random variables, the average to compute Φ can be computed in closed form for polynomials and several practically used nonlinearities (ReLU, ERF, etc) [11]. Now, to compute the backward pass, we must

$$\frac{\partial}{\partial \tau} G(\tau, \boldsymbol{x}, \boldsymbol{x}') = - \left\langle \dot{\phi}(h(\tau, \boldsymbol{x})) \dot{\phi}(h(\tau', \boldsymbol{x}')) \right\rangle G(\tau, \boldsymbol{x}, \boldsymbol{x}') \tag{80}$$

Again, the bivariate Gaussian integral $\left\langle \dot{\phi}(h(\tau, \boldsymbol{x})) \dot{\phi}(h(\tau', \boldsymbol{x}')) \right\rangle$ has a closed form expression for many regularly used activation functions ϕ. Given that we have solved for $\Phi(\tau)$ and $G(\tau)$, the dynamics of the predictor is

$$\begin{aligned} &\frac{d}{dt} f(\boldsymbol{x}, t) = \eta_0 \mathbb{E}_{\boldsymbol{x}'} K(\boldsymbol{x}, \boldsymbol{x}') \Delta(\boldsymbol{x}', t) \ , \ \Delta(\boldsymbol{x}, t) = - \frac{\partial \mathcal{L}}{\partial f(\boldsymbol{x}, t)} \\ &K(\boldsymbol{x}, \boldsymbol{x}') = \int_0^1 d\tau \ G(\tau, \boldsymbol{x}, \boldsymbol{x}') \Phi(\tau, \boldsymbol{x}, \boldsymbol{x}'). \end{aligned} \tag{81}$$

This is an exact description of the network predictor dynamics in the $N \to \infty, L \to \infty$ and $\gamma_0 \to 0$ limit. However, the $\gamma_0 \to 0$ limit lacks a fundamental phenomenon of deep neural networks, namely feature learning. To gain analytical insight into the feature learning dynamics, we will study linear networks in the next section.

G.2 Infinite Depth ResNets with Linear Activations

In the case where the activations are linear, we can also close the equations at the level of the kernels (since activations remain Gaussian even after feature learning). To provide an exactly solveable model of the dynamics, we consider computing the kernels after one step of feature learning.

G.2.1 Full DMFT equations in Deep Linear Networks

The DMFT equations for the preactivation processes have the form

$$h(\tau, \boldsymbol{x}, t) = h(0, \boldsymbol{x}, t) + \int_0^\tau du(\tau', \boldsymbol{x}, t) + \eta_0\gamma_0 \int_0^\tau d\tau' \int d\boldsymbol{x}' \int ds\, C_h(\tau', \boldsymbol{x}, \boldsymbol{x}', t, s) g(\tau', \boldsymbol{x}', s)$$
$$g(\tau, \boldsymbol{x}, t) = g(1, \boldsymbol{x}, t) + \int_\tau^1 dr(\tau', \boldsymbol{x}, t) + \eta_0\gamma_0 \int_\tau^1 d\tau' \int d\boldsymbol{x}' \int ds\, C_g(\tau', \boldsymbol{x}, \boldsymbol{x}', t, s) h(\tau', \boldsymbol{x}', s) \tag{82}$$

where C_h, C_g have the usual expressions. The field derivatives (to get the response functions) have the form

$$\begin{aligned}
\frac{\delta h(\tau, \boldsymbol{x}, t)}{\delta du(\tau', \boldsymbol{x}', s)} &= \Theta(\tau - \tau')\delta(\boldsymbol{x} - \boldsymbol{x}')\delta(t - s) \\
&\quad + \eta_0\gamma_0 \int_0^\tau d\tau'' \int dx'' \int dt' C_h(\tau'', \boldsymbol{x}, \boldsymbol{x}'', t, t') \frac{\delta g(\tau'', \boldsymbol{x}'', t')}{\delta du(\tau', \boldsymbol{x}', s)} \\
\frac{\delta g(\tau, \boldsymbol{x}, t)}{\delta du(\tau', \boldsymbol{x}', s)} &= \eta_0\gamma_0 \int_\tau^1 d\tau'' \int dx'' \int dt' C_g(\tau'', \boldsymbol{x}, \boldsymbol{x}', t, t') \frac{\delta h(\tau'', \boldsymbol{x}'', t')}{\delta du(\tau', \boldsymbol{x}', s)} \\
\frac{\delta h(\tau, \boldsymbol{x}, t)}{\delta dr(\tau', \boldsymbol{x}', s)} &= \eta_0\gamma_0 \int_0^\tau d\tau'' \int dx'' \int dt' C_h(\tau'', \boldsymbol{x}, \boldsymbol{x}', t, t') \frac{\delta g(\tau'', \boldsymbol{x}'', t')}{\delta dr(\tau', \boldsymbol{x}', s)} \\
\frac{\delta g(\tau, \boldsymbol{x}, t)}{\delta dr(\tau', \boldsymbol{x}', s)} &= \Theta(\tau' - \tau)\delta(\boldsymbol{x} - \boldsymbol{x}')\delta(t - s) \\
&\quad + \eta_0\gamma_0 \int_\tau^1 d\tau'' \int dx'' \int dt' C_g(\tau'', \boldsymbol{x}, \boldsymbol{x}', t, t') \frac{\delta h(\tau'', \boldsymbol{x}'', t')}{\delta dr(\tau', \boldsymbol{x}', s)}
\end{aligned} \tag{83}$$

Since the activations are linear, these derivatives are no longer stochastic and we have the equal layer time $\tau = \tau'$ derivatives as the response functions

$$A(\tau, \boldsymbol{x}, \boldsymbol{x}', t, s) = \frac{\delta h(\tau, \boldsymbol{x}, t)}{\delta dr(\tau, \boldsymbol{x}', s)}\ ,\ B(\tau, \boldsymbol{x}, \boldsymbol{x}', t, s) = \frac{\delta g(\tau, \boldsymbol{x}, t)}{\delta du(\tau, \boldsymbol{x}', s)} \tag{84}$$

Next, we compute the kernels as simple Gaussian averages

$$H(\tau, \boldsymbol{x}, \boldsymbol{x}', t, s) = \langle h(\tau, \boldsymbol{x}, t) h(\tau, \boldsymbol{x}, t) \rangle\ ,\ G(\tau, \boldsymbol{x}, \boldsymbol{x}', t, s) = \langle g(\tau, \boldsymbol{x}, t) g(\tau, \boldsymbol{x}, t) \rangle \tag{85}$$

In the next section, we show a simple example where these equations become simple linear differential equations.

G.2.2 One step One Sample Feature Kernel Equations

Before training, the initial conditions for the kernels $H_0(\tau)$ and $G_0(\tau)$ are

$$\begin{aligned}
&\partial_\tau H_0(\tau) = H_0(\tau)\ ,\ \partial_\tau G_0(\tau) = -G_0(\tau) \\
&\implies H_0(\tau) = e^\tau\ ,\ G_0(\tau) = e^{-\tau} + 1 - e^{-1}
\end{aligned} \tag{86}$$

After a step of gradient descent, the preactivation at layer time τ has the form

$$h(\tau) = h(0) + \int_0^\tau du(\tau') + \eta_0\gamma_0 \int_0^\tau d\tau' [A(\tau') + H_0(\tau')y] g_0(\tau') \tag{87}$$

We let $H(\tau), G(\tau)$ represent the kernels after one step. The response function after one step has satisfies the differential equation

$$\partial_\tau A(\tau) = A(\tau) + H_0(\tau)y\ ,\ A(0) = 0 \implies A(\tau) = \tau e^\tau\, y \tag{88}$$

The kernel $H(\tau)$ after one step therefore satisfies

$$\begin{aligned}
H(\tau) &= H(0) + \int_0^\tau d\tau' H(\tau') \\
&\quad + \eta_0^2\gamma_0^2 \int_0^\tau \int_0^\tau d\tau' d\tau'' G_0(\tau', \tau'')[A(\tau') + H_0(\tau')y][A(\tau'') + H_0(\tau'')y] \\
&= H(0) + \int_0^\tau d\tau' H(\tau') + \eta_0^2\gamma_0^2 y^2 \int_0^\tau \int_0^\tau d\tau' d\tau'' G_0(\tau', \tau'') e^{\tau' + \tau''} [\tau' + 1][\tau'' + 1].
\end{aligned} \tag{89}$$

where $G_0(\tau,\tau') = \langle g_0(\tau) g_0(\tau')\rangle$. Differentiating both sides with respect to τ gives the integro-differential equation

$$\partial_\tau H(\tau) = H(\tau) + 2\eta_0^2\gamma_0^2 y^2 e^\tau[\tau+1]\int_0^\tau d\tau' G_0(\tau,\tau') e^{\tau'}[\tau'+1] \tag{90}$$

Note that $G_0(\tau,\tau') = G_0(\tau) = e^{-\tau} + 1 - e^{-1}$ for $\tau' < \tau$. So we can pull this out of the integral and we are left with

$$\int_0^\tau d\tau' e^{\tau'}[\tau'+1] = \tau e^\tau \tag{91}$$

We are finally left with the ODE

$$\partial_\tau H(\tau) = H(\tau) + 2\eta_0^2\gamma_0^2 y^2\, e^{2\tau}[e^{-\tau} + 1 - e^{-1}][\tau^2+\tau] \tag{92}$$

Integrating gives

$$H(\tau) = e^\tau H(0) + 2\eta_0^2\gamma_0^2 e^\tau\left[\frac{1}{3}\tau^3 + \frac{1}{2}\tau^2 + (1+e^{-1})(\tau^2 e^\tau - \tau e^\tau + e^\tau - 1)\right] \tag{93}$$

This solution gives the profile of the feature kernels after a single step of gradient descent. This is verified numerically in Figure G.1. At small τ this goes as $H(\tau) - H_0(\tau) \sim \eta_0^2\gamma_0^2[1+(1+e^{-1})]\tau^2$.

H Finite Width Convergence: Fluctuations do not scale with L

At fixed L, one can compute systematic asymptotic corrections in $\mathcal{O}_N(N^{-1})$ to the statistical properties of $\boldsymbol{q}$. This follows closely from the finite width analysis around mean field dynamics performed by [7]. The idea of their expansion is to extract from $\mathcal{S}$ information about finite size deviations from the infinite width DMFT process. The key fact which appears in the next to leading order terms is that at finite width N, the order parameters $\boldsymbol{q}$ become random with covariance

$$\text{Cov}(\boldsymbol{q}) \sim \frac{1}{N}\boldsymbol{\Sigma}^L + \mathcal{O}_N(N^{-2})\ ,\ \boldsymbol{\Sigma}^L = \left[-\nabla^2\mathcal{S}(\boldsymbol{q}_\infty)\right]^{-1} \tag{94}$$

The matrix $\frac{1}{N}\boldsymbol{\Sigma}^L$ gives the leading order (in powers of $1/N$) covariance of the order parameters over random initialization at width N depth L. The goal of this section is to show that the entries of this covariance matrix $\boldsymbol{\Sigma}^L$ that correspond to the NTK and (consequently the predictor f) do not diverge with L for the architecture of this paper (residual networks with $1/\sqrt{L}$ branch scaling), but rather approach a well defined limit at rate $\mathcal{O}(L^{-1})$. We again stress that this is unique to this architecture. Non-residual deep networks networks have propagator entries which scale linearly with L since variances accumulate rather than average out over layers [5, 7]. We note that the predictor f can be included in the set of order parameters $\boldsymbol{q}$ at depth L, so that the finite width deviation scales as

$$\left\langle (f_{N,L} - f_{\infty,L})^2\right\rangle \sim \frac{1}{N}\Sigma_f^L \tag{95}$$

where averages are taken over random initialization.

H.1 Components of the Hessian

Since we are primarily interested in the scaling of the finite width effects with respect to N and L, we will suppress the sample and time indices of the kernels and instead focus on the dependence of the fluctuations on layer index ℓ. This will not change the analysis of the scaling but will save us from tracking a tremendous amount of indices. We will let Φ^ℓ to represent the feature kernel at layer ℓ and G^ℓ the gradient kernel at layer ℓ with no time or sample indices. Under this simplified notation, the Hessian has entries of the form

$$\begin{aligned}
\frac{\partial^2\mathcal{S}}{\partial\hat{\Phi}^\ell\partial\hat{\Phi}^{\ell'}} &= \left\langle \phi(h^\ell)\phi(h^\ell)\phi(h^{\ell'})\phi(h^{\ell'})\right\rangle - \Phi^\ell\Phi^{\ell'} \equiv \kappa_\Phi^{\ell,\ell'} \\
\frac{\partial^2\mathcal{S}}{\partial\hat{\Phi}^\ell\partial\Phi^{\ell'}} &= \delta_{\ell,\ell'} - \frac{\partial}{\partial\Phi^{\ell'}}\left\langle\phi(h^\ell)\phi(h^\ell)\right\rangle \equiv \delta_{\ell,\ell'} - \frac{1}{L}D_\Phi^{\ell,\ell'} \\
\frac{\partial^2\mathcal{S}}{\partial\Phi^\ell\partial\Phi^{\ell'}} &= 0
\end{aligned} \tag{96}$$

First, we note that since the h^ℓ fields have *long-range correlations*, the $\kappa^{\ell,\ell'}$ is non-vanishing for all ℓ, ℓ'. We next note that $D_\Phi^{\ell,\ell'} \sim \mathcal{O}(1)$ since $\frac{\partial h^\ell}{\partial \Phi^{\ell'}} \sim \mathcal{O}(L^{-1})$ and by Price's theorem,

$$D_\Phi^{\ell,\ell'} = \frac{1}{2}\left\langle \frac{\partial^2}{\partial u^{\ell'}\partial u^{\ell'}}[\phi(h^\ell)\phi(h^\ell)]\right\rangle + 2L\left\langle \dot{\phi}(h^\ell)\phi(h^\ell)\frac{\partial h^\ell}{\partial \Phi^{\ell'}}\right\rangle \sim \mathcal{O}(1) \tag{97}$$

Similarly, we also need to calculate the terms involving G^ℓ

$$\begin{aligned}
\frac{\partial^2 \mathcal{S}}{\partial \hat{G}^\ell \partial \hat{G}^{\ell'}} &= \left\langle g^\ell g^\ell g^{\ell'} g^{\ell'}\right\rangle - G^\ell G^{\ell'} \equiv \kappa_G^{\ell,\ell'} \\
\frac{\partial^2 \mathcal{S}}{\partial \hat{G}^\ell \partial G^{\ell'}} &= \delta_{\ell,\ell'} - \frac{\partial}{\partial G^{\ell'}}\left\langle g^\ell g^\ell\right\rangle \equiv \delta_{\ell,\ell'} - \frac{1}{L}D_G^{\ell,\ell'} \\
\frac{\partial^2 \mathcal{S}}{\partial G^\ell \partial G^{\ell'}} &= 0
\end{aligned} \tag{98}$$

Lastly, we must consider the cross terms involving both $\{\Phi^\ell\}$ and $\{G^\ell\}$

$$\begin{aligned}
\frac{\partial^2 \mathcal{S}}{\partial \hat{\Phi}^\ell \partial \hat{G}^{\ell'}} &= \left\langle \phi(h^\ell)\phi(h^\ell) g^{\ell'} g^{\ell'}\right\rangle - \Phi^\ell G^{\ell'} \equiv \kappa_{\Phi,G}^{\ell,\ell'} \\
\frac{\partial^2 \mathcal{S}}{\partial \hat{\Phi}^\ell \partial G^{\ell'}} &= -\frac{\partial}{\partial G^{\ell'}}\left\langle \phi(h^\ell)\phi(h^\ell)\right\rangle \equiv -\frac{1}{L}D_{\Phi,G}^{\ell,\ell'} \\
\frac{\partial^2 \mathcal{S}}{\partial \hat{G}^\ell \partial \Phi^{\ell'}} &= -\frac{\partial}{\partial \Phi^{\ell'}}\left\langle g^\ell g^\ell\right\rangle \equiv -\frac{1}{L}D_{G,\Phi}^{\ell,\ell'} \\
\frac{\partial^2 \mathcal{S}}{\partial \Phi^\ell \partial G^{\ell'}} &= 0
\end{aligned} \tag{99}$$

The κ tensor represents an instantaneous source of variance for the kernels, while the D tensor represents the sensitivity of the kernel at layer ℓ to a perturbation in the kernel at layer ℓ'. All entries of each D matrix are $\mathcal{O}(1)$. Now, to compute the propagator, we construct block matrices

$$\boldsymbol{\kappa} = \begin{bmatrix} \boldsymbol{\kappa}_\Phi & \boldsymbol{\kappa}_{\Phi,G} \\ \boldsymbol{\kappa}_{G,\Phi} & \boldsymbol{\kappa}_G \end{bmatrix}, \ \boldsymbol{D} = \begin{bmatrix} \boldsymbol{D}_\Phi & \boldsymbol{D}_{\Phi,G} \\ \boldsymbol{D}_{G,\Phi} & \boldsymbol{D}_G \end{bmatrix} \tag{100}$$

Now, to obtain the propagator $\boldsymbol{\Sigma}$, we must compute the inverse of the Hessian $\nabla^2\mathcal{S}$

$$\boldsymbol{\Sigma} = -\begin{bmatrix} \boldsymbol{0} & (\boldsymbol{I} - \frac{1}{L}\boldsymbol{D})^\top \\ \boldsymbol{I} - \frac{1}{L}\boldsymbol{D} & \boldsymbol{\kappa} \end{bmatrix}^{-1} = \begin{bmatrix} \left(\boldsymbol{I} - \frac{1}{L}\boldsymbol{D}\right)^{-1}\boldsymbol{\kappa}\left[\left(\boldsymbol{I} - \frac{1}{L}\boldsymbol{D}\right)^{-1}\right]^\top & -\left(\boldsymbol{I} - \frac{1}{L}\boldsymbol{D}\right)^{-1} \\ -\left[\left(\boldsymbol{I} - \frac{1}{L}\boldsymbol{D}\right)^{-1}\right]^\top & \boldsymbol{0} \end{bmatrix} \tag{101}$$

The observables of interest are the top-left block which gives the joint covariance over all $\{\Phi^\ell, G^\ell\}$ variables. We introduce the following 2×2 block matrices

$$\boldsymbol{\kappa}^{\ell,\ell'} = \begin{bmatrix} \kappa_\Phi^{\ell,\ell'} & \kappa_{\Phi G}^{\ell,\ell'} \\ \kappa_{G\Phi}^{\ell,\ell'} & \kappa_G^{\ell,\ell'} \end{bmatrix}, \ \boldsymbol{\Sigma}^{\ell,\ell'} = \begin{bmatrix} \Sigma_\Phi^{\ell,\ell'} & \Sigma_{\Phi G}^{\ell,\ell'} \\ \Sigma_{G\Phi}^{\ell,\ell'} & \Sigma_G^{\ell,\ell'} \end{bmatrix}, \ \boldsymbol{D}^{\ell,\ell'} = \begin{bmatrix} D_\Phi^{\ell,\ell'} & D_{\Phi G}^{\ell,\ell'} \\ D_{G\Phi}^{\ell,\ell'} & D_G^{\ell,\ell'} \end{bmatrix} \tag{102}$$

The above equation gives us we get the following 2×2 matrix equations

$$\boldsymbol{\Sigma}^{\ell,\ell'} - \frac{1}{L}\sum_k \boldsymbol{D}^{\ell,k}\boldsymbol{\Sigma}^{k,\ell'} - \frac{1}{L}\sum_k \boldsymbol{\Sigma}^{\ell,k}[\boldsymbol{D}^{\ell',k}]^\top + \frac{1}{L^2}\sum_{kk'} \boldsymbol{D}^{\ell,k}\boldsymbol{\Sigma}^{k,k'}[\boldsymbol{D}^{k',\ell'}]^\top = \boldsymbol{\kappa}^{\ell,\ell'} \tag{103}$$

We see that the solutions $\boldsymbol{\Sigma}^{\ell,\ell'}$ to the above equation will be $\mathcal{O}_L(1)$. This is precisely due to the factors of $1/L$ which appear in the above sums. One way to see this is to consider a large $L \gg 1$ limit where the above layer sums converge to integrals over τ

$$\begin{aligned}
&\boldsymbol{\Sigma}(\tau,\tau') - \int d\tau'' \boldsymbol{D}(\tau,\tau'')\boldsymbol{\Sigma}(\tau'',\tau') - \int d\tau'' \boldsymbol{\Sigma}(\tau'',\tau')\boldsymbol{D}(\tau,\tau'')^\top \\
&+ \int d\tau'' d\tau''' \boldsymbol{D}(\tau,\tau'')\boldsymbol{\Sigma}(\tau'',\tau''')\boldsymbol{D}(\tau',\tau''')^\top = \boldsymbol{\kappa}(\tau,\tau')
\end{aligned} \tag{104}$$

So we find that the covariance of kernels at each pair of layers $\boldsymbol{\Sigma}(\tau,\tau')$ is $\mathcal{O}_L(1)$.

H.2 Variance of the NTK and Predictor

Using the above fact that $\mathbf{\Sigma}^{\ell,\ell'} \sim \mathcal{O}_L(1)$, we can reason about the variance of the neural tangent kernel K

$$\begin{aligned}\text{Var}(K) = &\frac{1}{NL^2}\sum_{\ell\ell'}\mathbf{\Sigma}_{\Phi}^{\ell,\ell'}G^{\ell+1}G^{\ell'+1} + \frac{1}{NL^2}\sum_{\ell\ell'}\mathbf{\Sigma}_{G}^{\ell,\ell'}\Phi^{\ell-1}\Phi^{\ell'-1} \\ &+ \frac{2}{NL^2}\sum_{\ell\ell'}\mathbf{\Sigma}_{\Phi G}^{\ell,\ell'}G^{\ell+1}\Phi^{\ell'-1} \sim \mathcal{O}_{L,N}(N^{-1}). \end{aligned} \tag{105}$$

This demonstrates that the NTK will have initialization variance that scales only with the width N but not the depth L. Using the fact that $\frac{\partial f}{\partial t} = K\Delta$, we see that f will inherit fluctuations on the same scale of $\mathcal{O}_{N,L}(N^{-1/2})$. The depth L NTK and predictor covariance converge to the infinite depth propagator with rate $\mathcal{O}(L^{-1})$.

I Finite Width And Depth Error Analysis

In this section, we combine the results of the previous sections. We consider the following error decomposition where we introduce the norm $||z|| = \sqrt{\langle z^2\rangle}$ and $\langle\rangle$ denotes an average over random initializations of the network. Applying the triangle inequality on this norm, we have

$$\begin{aligned}||f_{N,L} - f_{\infty,\infty}|| \leq &||f_{N,L} - \langle f_{N,L}\rangle|| \quad \text{(Finite Width Variance)} \\ &+ ||\langle f_{N,L}\rangle - f_{\infty,L}|| \quad \text{(Mean Predictor Finite Width Error)} \\ &+ ||f_{\infty,L} - f_{\infty,\infty}|| \quad \text{(Finite Depth Error at Infinite Width)}\end{aligned} \tag{106}$$

From the preeceeding section H, the first term has asymptotic behavior

$$\left\langle (f_{N,L} - \langle f_{N,L}\rangle)^2 \right\rangle \sim \frac{1}{N}\Sigma_f^L + \mathcal{O}(N^{-2}) = \frac{1}{N}\Sigma_f^\infty + \mathcal{O}\left(\frac{1}{NL} + \frac{1}{N^2}\right) \tag{107}$$

where the last term comes from the fact that Σ^L approximates the infinite depth propagator Σ^∞ with error $1/L$. Next, we can apply results developed from [7] which show that the mean predictor satisfies $\langle f_{N,L}\rangle = f_{\infty,L} + \mathcal{O}(N^{-1})$ to find that mean finite width error scales as

$$|\langle f_{N,L}\rangle - f_{\infty,L}|^2 \sim \mathcal{O}(N^{-2}) \tag{108}$$

Lastly, the finite depth error in the infinite width DMFT arises from the $1/L$ discretization effect. Thus the square error goes as

$$|f_{\infty,L} - f_{\infty,\infty}|^2 \sim \mathcal{O}(L^{-2}). \tag{109}$$

which follows from the discretization error of the limiting large depth process for the NTK and thus the predictor. Altogether, we have

$$||f_{N,L} - f_{\infty,\infty}|| \sim \sqrt{\frac{1}{N}\Sigma_f^\infty + \mathcal{O}\left(\frac{1}{NL}\right)} + \mathcal{O}\left(\frac{1}{N}\right) + \mathcal{O}\left(\frac{1}{L}\right) \tag{110}$$

We see that at large width and depth this is dominated by the first term which comes from the finite width fluctuations. However, if the model is ensembled, then we expect a faster rate of convergence from the last two sources of error. Indeed, we provide an experiment verifying that ensembling improves the rate of convergence in Figure C.8.

J Why Parameterization Influences Hyperparameter Transfer at Finite Widths and Depths

One may wonder why other parameterizations which also admit a scaling limit (such as Neural Tangent Parameterization) do not transfer as well as networks in μP or why our proposal transfers over depth but others do not. To achieve successful transfer across finite widths, it is insufficient to reason about the infinite width/depth scaling limit, rather one needs to identify a parameterization that

minimizes the approximation error between finite networks and the scaling limit (of course subject to the obvious constraints of training in finite time, learning features at the desired rate etc).

Alternative parameterizations that do not keep feature updates constant, can be thought of as choosing a function $\gamma_0(N, L)$ which describes how much the features evolve with N, L. Suppose further that this has a limiting value $\lim_{N,L\to\infty} \gamma_0(N, L) = \gamma_0^\star$. For example, neural tangent parameterization in our $\frac{1}{\sqrt{L}}$ ResNet corresponds to $\gamma_0 = \frac{1}{\sqrt{N}}$ with $\gamma_0^\star = 0$. We let the logits for a finite width/depth model in this parameterization be $f_{N,L}^{\gamma_0(N,L)}$ represent a finite width N and depth L network predictions with feature learning scale $\gamma_0(N, L)$. The approximation error this (N, L) network and another (N', L') network has two potential sources of error

$$||f_{N,L}^{\gamma_0(N,L)} - f_{N',L'}^{\gamma_0(N',L')}|| \leq ||f_{N,L}^{\gamma_0(N,L)} - f_{\infty,\infty}^{\gamma_0(N,L)}|| + ||f_{N',L'}^{\gamma_0(N,L)} - f_{\infty,\infty}^{\gamma_0(N',L')}|| + ||f_{\infty,\infty}^{\gamma_0(N,L)} - f_{\infty,\infty}^{\gamma_0(N',L')}|| \tag{111}$$

Heuristically, one can eliminate the final approximation error by choosing a parameterization where $\gamma_0(N, L) = \gamma_0$ is a width/depth-independent constant. Then one is left with finite approximation errors to an infinite model with the same rate of feature learning.

K Architectural + Optimization Extensions

In this section, we explore various extensions of the model written in the main text. Most of these extensions do not modify the scaling rule or the procedural ideas which we used to characterize the limit. We will explore convolutions, multiple layers per block, training read-in and read-out matrices, discrete time training dynamics (finite step size SGD), momentum, and LayerNorm.

K.1 Convolutional Models

A DMFT for convolutional ResNets can be easily obtained by introducing spatial coordinates $\mathfrak{a}, \mathfrak{b}$ of each layer's representation [6]. The preactivation vectors $\boldsymbol{h}_{\mathfrak{a}}^{\ell}(\boldsymbol{x}, t) \in \mathbb{R}^N$ represent the activity of all N channels in layer ℓ at spatial position $\mathfrak{a}$. The trainable weights take the form $\boldsymbol{W}_{\mathfrak{a}}^{\ell} \in \mathbb{R}^{N\times N}$ describe the filter at a spatial displacement $\mathfrak{a}$ from its center. The recursion of the ResNet is

$$\boldsymbol{h}_{\mathfrak{a}}^{\ell+1}(\boldsymbol{x}, t) = \boldsymbol{h}_{\mathfrak{a}}^{\ell}(\boldsymbol{x}, t) + \frac{1}{\sqrt{NL}} \sum_{\mathfrak{b}} \boldsymbol{W}_{\mathfrak{b}}^{\ell} \phi(\boldsymbol{h}_{\mathfrak{a}+\mathfrak{b}}^{\ell}(\boldsymbol{x}, t)) \tag{112}$$

where the sum over $\mathfrak{b}$ runs over all valid spatial positions in the filter. The relevant order parameters are still feature-feature correlations, but now with additional spatial indices

$$\Phi_{\mathfrak{a},\mathfrak{b}}^{\ell}(\boldsymbol{x}, \boldsymbol{x}', t, s) = \frac{1}{N} \phi(\boldsymbol{h}_{\mathfrak{a}}^{\ell}(\boldsymbol{x}, t)) \cdot \phi(\boldsymbol{h}_{\mathfrak{b}}^{\ell}(\boldsymbol{x}', s)) \tag{113}$$

A similar gradient-gradient kernel is also crucial $G_{\mathfrak{a},\mathfrak{b}}^{\ell}(\boldsymbol{x}, \boldsymbol{x}', t, s)$. From these objects, one can construct a limiting DMFT at $N \to \infty$

K.2 Multiple Layers Per Residual Block

Many residual neural networks, including the popular ResNet-18 and ResNet-50 models, there are more than a single convolution layer on a residual branch. In this section, we show that this does not alter the scaling rule or the convergence to a well defined large width and depth limit provided the number of branches L goes to infinity with number of layers per block K fixed. We therefore work with the case of $K \sim \mathcal{O}(1)$ hidden layers on each branch and still take the branch number L to

infinity. The recurrence of interest is

$$
\begin{aligned}
f &= \frac{1}{\gamma_0 N} \boldsymbol{w}^L \cdot \phi(\boldsymbol{h}^L) \\
\boldsymbol{h}^{\ell+1} &= \boldsymbol{h}^\ell + \frac{1}{\sqrt{L}} \tilde{\boldsymbol{h}}^{\ell,K} \ , \ \ell \in \{1, ..., L-1\} \\
\tilde{\boldsymbol{h}}^{\ell,k+1} &= \frac{1}{\sqrt{N}} \boldsymbol{W}^{\ell,k} \phi(\tilde{\boldsymbol{h}}^{\ell,k}) \ , \ k \in \{1, ..., K-1\} \\
\tilde{\boldsymbol{h}}^{\ell,1} &= \frac{1}{\sqrt{N}} \boldsymbol{W}^{\ell,0} \phi(\boldsymbol{h}^\ell) \\
\boldsymbol{h}^1 &= \frac{1}{\sqrt{D}} \boldsymbol{W}^0 \boldsymbol{x}
\end{aligned} \tag{114}
$$

The goal is to verify that gradient based learning on $\{\boldsymbol{W}^{\ell,k}\}_{\ell \in [L], k \in [K]}$ gives a well defined infinite width and depth limit $N \to \infty$ and $L \to \infty$. The weight dynamics are

$$
\begin{aligned}
\frac{d}{dt} \boldsymbol{W}^{\ell,k} &= \frac{\eta_0 \gamma_0}{\sqrt{NL}} \, \mathbb{E}_{\boldsymbol{x}} \, \Delta(\boldsymbol{x}, t) \, \tilde{\boldsymbol{g}}^{\ell,k+1}(\boldsymbol{x}, t) \phi\left(\tilde{\boldsymbol{h}}^{\ell,k}(\boldsymbol{x}, t)\right)^\top \ , \ k \in \{1, ..., K-1\} \\
\frac{d}{dt} \boldsymbol{W}^{\ell,K} &= \frac{\eta_0 \gamma_0}{\sqrt{NL}} \, \mathbb{E}_{\boldsymbol{x}} \, \Delta(\boldsymbol{x}, t) \, \boldsymbol{g}^{\ell+1}(\boldsymbol{x}, t) \phi\left(\tilde{\boldsymbol{h}}^{\ell,k}(\boldsymbol{x}, t)\right)^\top
\end{aligned} \tag{115}
$$

where we have defined the gradient fields

$$
\begin{aligned}
\tilde{\boldsymbol{g}}^{\ell,k}(\boldsymbol{x}, t) &= N \gamma_0 \sqrt{L} \frac{\partial f(\boldsymbol{x}, t)}{\partial \tilde{\boldsymbol{h}}^{\ell,k}(\boldsymbol{x}, t)} \sim \mathcal{O}_{N,L}(1) \\
\boldsymbol{g}^\ell(\boldsymbol{x}, t) &= N \gamma_0 \frac{\partial f(\boldsymbol{x}, t)}{\partial \boldsymbol{h}^\ell(\boldsymbol{x}, t)} \sim \mathcal{O}_{N,L}(1)
\end{aligned} \tag{116}
$$

We next examine the scale of the updates to preactivation features $\tilde{\boldsymbol{h}}^{\ell,k}$ and $\boldsymbol{h}^\ell$

$$
\begin{aligned}
\tilde{\boldsymbol{h}}^{\ell,k+1}(\boldsymbol{x}, t) &= \frac{1}{\sqrt{N}} \boldsymbol{W}^{\ell,k}(0) \phi(\tilde{\boldsymbol{h}}^{\ell,k}) + \frac{\eta_0 \gamma_0}{\sqrt{L}} \mathbb{E}_{\boldsymbol{x}'} \int_0^t ds \, \Delta(\boldsymbol{x}', s) \Phi^{\ell,k}(\boldsymbol{x}, \boldsymbol{x}'; t, s) \tilde{\boldsymbol{g}}^{\ell,k+1}(\boldsymbol{x}', s) \\
\boldsymbol{h}^{\ell+1}(\boldsymbol{x}, t) &= \boldsymbol{h}^\ell(\boldsymbol{x}, t) + \frac{1}{\sqrt{LN}} \boldsymbol{W}^{\ell,K}(0) \phi(\tilde{\boldsymbol{h}}^{\ell,K-1}) \\
&\quad + \frac{\eta_0 \gamma_0}{L} \mathbb{E}_{\boldsymbol{x}'} \int_0^t ds \, \Delta(\boldsymbol{x}', s) \Phi^{\ell,K}(\boldsymbol{x}, \boldsymbol{x}', t, s) \boldsymbol{g}^{\ell+1}(\boldsymbol{x}', s)
\end{aligned} \tag{117}
$$

We see that the $\tilde{\boldsymbol{h}}^{\ell,k}$ features will be uncorrelated across k since there are no skip connections within a block. However, the $\boldsymbol{h}^\ell$ exhibit long range correlations as before. To understand the cumulative impact of the feature learning corrections to the preactivations, we must reason about how correlated the gradient signals $\boldsymbol{g}^\ell$ and $\tilde{\boldsymbol{g}}^{\ell,k}$ are across different layers. We

$$
\begin{aligned}
\tilde{\boldsymbol{g}}^{\ell,k}(\boldsymbol{x}, t) &= \frac{1}{\sqrt{N}} \dot{\phi}(\tilde{\boldsymbol{h}}^{\ell,k}(\boldsymbol{x}, t)) \odot \boldsymbol{W}^{\ell,k}(t)^\top \tilde{\boldsymbol{g}}^{\ell,k+1}(\boldsymbol{x}, t) \\
\boldsymbol{g}^\ell(\boldsymbol{x}, t) &= \boldsymbol{g}^{\ell+1}(\boldsymbol{x}, t) + \frac{1}{\sqrt{NL}} \dot{\phi}(\boldsymbol{h}^\ell(\boldsymbol{x}, t)) \odot \left[\boldsymbol{W}^{\ell,0}(t)^\top \tilde{\boldsymbol{g}}^{\ell,1}(\boldsymbol{x}, t)\right]
\end{aligned} \tag{118}
$$

We see that the skip connections keep long range correlations in the $\boldsymbol{g}^\ell$ signals, however, the $\tilde{\boldsymbol{g}}^{\ell,k}$ are uncorrelated with one another, as in a standard feedforward network.

K.2.1 The Limiting Behavior of Multi-layer per Block Model

Based on the correlation structures, of the $\boldsymbol{g}^\ell$ and $\tilde{\boldsymbol{g}}^{\ell,k}$ gradient featuers, we have the following DMFT equations at infinite width and large depth

$$
\begin{aligned}
h^{\ell+1}(\boldsymbol{x}, t) &= h^1(\boldsymbol{x}, t) + \frac{1}{\sqrt{L}} \sum_{\ell'=1}^{\ell+1} u^{\ell'}(\boldsymbol{x}, t) \\
&\quad + \frac{\eta_0 \gamma_0}{L} \sum_{\ell'=1}^{\ell+1} \mathbb{E}_{\boldsymbol{x}'} \int_0^t ds \, \Delta(\boldsymbol{x}', s) \Phi^{\ell'-1,K}(\boldsymbol{x}, \boldsymbol{x}', t, s) g^{\ell'}(\boldsymbol{x}', s) + \mathcal{O}(L^{-1/2})
\end{aligned} \tag{119}
$$

where the Gaussian source has the form

$$u^{\ell+1}(\boldsymbol{x}, t) \sim \mathcal{GP}(0, \boldsymbol{\Phi}^{\ell,K}) \tag{120}$$

where $\Phi^{\ell,K}$ is determined through the *lazy within-block recursion*

$$\Phi^{k,\ell}(\boldsymbol{x}, \boldsymbol{x}', t, s) = \left\langle \phi(u^{\ell,k}(\boldsymbol{x}, t))\phi(u^{\ell,k}(\boldsymbol{x}, t)) \right\rangle_{u^{\ell,k} \sim \mathcal{N}(0, \boldsymbol{\Phi}^{\ell,k-1})} \ , \ k \in \{1, ..., K-1\}. \tag{121}$$

Though all of the kernels are dynamical, it is as if the weights $\boldsymbol{W}^{\ell,k}$ are static for $k \in \{1, ..., K-1\}$. This is a consequence of the decorrelation of $\tilde{g}^{\ell,k}$ across k. We leave as a future problem whether it is possible to reparameterize the intermediate weights so that each of the internal block weight's dynamics move by $\mathcal{O}_L(1)$ and contribute additional feature learning terms in the dynamics.

K.3 Effect of Training Read-in Weights

If in addition to training the weights of the ResNet "body" we also train the readout and readin weights, they will also contribute to the total NTK

$$K(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{L} \sum_{\ell=1}^{L} G^{\ell+1}(\boldsymbol{x}, \boldsymbol{x}')\Phi^{\ell}(\boldsymbol{x}, \boldsymbol{x}') + G^1(\boldsymbol{x}, \boldsymbol{x}')K^x(\boldsymbol{x}, \boldsymbol{x}'). \tag{122}$$

where $K^x(\boldsymbol{x}, \boldsymbol{x}') = \frac{1}{D}\boldsymbol{x} \cdot \boldsymbol{x}'$ is the input kernel. We see that the addition of these two layers does not change the fact that the NTK is $\mathcal{O}_L(1)$ in this parameterization. Now, we need to verify that the first layer weights move by the appropriate scale

$$\frac{d}{dt} W^0_{ij}(t) = \frac{\eta_0 \gamma_0}{\sqrt{D}} \mathbb{E}_{\boldsymbol{x}} \Delta(\boldsymbol{x}, t) g^1_i(\boldsymbol{x}, t) x_j \sim \mathcal{O}_{L,N}(1) \tag{123}$$

These equations lead to the following base case equations for the fields $g^L(\boldsymbol{x}, t)$ and $h^1(\boldsymbol{x}, t)$

$$\begin{aligned} h^1(\boldsymbol{x}, t) &= u^1(\boldsymbol{x}, t) + \eta_0 \gamma_0 \mathbb{E}_{\boldsymbol{x}'} \int_0^t ds \Delta(\boldsymbol{x}', s) g^1(\boldsymbol{x}', s) K^x(\boldsymbol{x}, \boldsymbol{x}') \\ g^L(\boldsymbol{x}, t) &= \dot{\phi}(h^L(\boldsymbol{x}, t)) r^L(\boldsymbol{x}, t) + \eta_0 \gamma_0 \dot{\phi}(h^L(\boldsymbol{x}, t)) \mathbb{E}_{\boldsymbol{x}'} \int_0^t ds \ \Phi^L(\boldsymbol{x}, \boldsymbol{x}', t, s) \Delta(\boldsymbol{x}', s) \end{aligned} \tag{124}$$

These fields will therefore evolve by $\mathcal{O}_{N,L}(1)$ in this parameterization.

K.4 Discrete Time Updates

DMFT for feedforward networks can also be modified for discrete time [6, 7]. A straightforward extension of these results will give discrete time dynamics for the field dynamics h^ℓ, g^ℓ for our ResNet. The one subtlety about discrete time is that the neural tangent kernel no longer governs the evolution of the predictor f. Instead,

$$f(\boldsymbol{x}, t) = \eta_0 \sum_{s<t} \mathbb{E}_{\boldsymbol{x}' \sim \mathfrak{B}_s} \Delta(\boldsymbol{x}', s) \Phi^L(\boldsymbol{x}, \boldsymbol{x}', t, s) + \eta_0 \int d\boldsymbol{x}' \sum_{s<t} A^L(\boldsymbol{x}, \boldsymbol{x}', t, s), \tag{125}$$

where $\mathfrak{B}_s$ is the minibatch at timestep s and $A^L(\boldsymbol{x}, \boldsymbol{x}', t, s) = \frac{\sqrt{L}}{\eta_0 \gamma_0} \left\langle \frac{\delta \phi(h^L(\boldsymbol{x}, t))}{\delta r^L(\boldsymbol{x}', s)} \right\rangle$ is the final layer response function. These are now evaluated for t, s integer rather than real numbers.

K.5 Momentum

Momentum can be easily handled within our framework. Let $\boldsymbol{\theta} = \text{Vec}\{\boldsymbol{W}^0, \boldsymbol{W}^1, ..., \boldsymbol{W}^{L-1}, \boldsymbol{w}^L\}$ represent the concatenation of all trainable network parameters. The momentum update is controlled by a momentum parameter α which controls the exponential moving average of gradients

$$\begin{aligned} \boldsymbol{\theta}(t+1) &= \boldsymbol{\theta}(t) + \eta_0 \gamma_0 \boldsymbol{v}(t) \\ \boldsymbol{v}(t) &= \alpha \ \boldsymbol{v}(t-1) - (1-\alpha) N \gamma_0 \nabla_{\boldsymbol{\theta}} \mathbb{E}_{\boldsymbol{x}} \mathcal{L}[f(\boldsymbol{x}, \boldsymbol{\theta}(t))]. \end{aligned} \tag{126}$$

Writing this out for a particular hidden layer yields

$$\boldsymbol{W}^\ell(t+1) = \boldsymbol{W}^\ell(t) + \eta_0\gamma_0 \boldsymbol{V}^\ell(t)$$
$$\boldsymbol{V}^\ell(t) = \alpha \boldsymbol{V}^\ell(t-1) + (1-\alpha)\frac{1}{\sqrt{LN}}\mathbb{E}_{\boldsymbol{x}}\Delta(\boldsymbol{x},t)\boldsymbol{g}^{\ell+1}(\boldsymbol{x},t)\phi(h^\ell(\boldsymbol{x},t))^\top \tag{127}$$

We can unfold the recurrence for $\boldsymbol{V}^\ell(t)$ to get the expression

$$\boldsymbol{V}^\ell(t) = \frac{(1-\alpha)}{\sqrt{LN}}\sum_{s=0}^{t}\alpha^{t-s}\,\mathbb{E}_{\boldsymbol{x}}\Delta(\boldsymbol{x},s)\boldsymbol{g}^{\ell+1}(\boldsymbol{x},s)\phi(\boldsymbol{h}^\ell(\boldsymbol{x},s))^\top \tag{128}$$

Plugging this into the update equations for $\boldsymbol{W}^\ell(t)$ we get

$$\boldsymbol{W}^\ell(t+1) = \boldsymbol{W}^\ell(t) + \frac{\eta_0\gamma_0(1-\alpha)}{\sqrt{LN}}\sum_{s=0}^{t}\alpha^{t-s}\,\mathbb{E}_{\boldsymbol{x}}\Delta(\boldsymbol{x},s)\boldsymbol{g}^{\ell+1}(\boldsymbol{x},s)\phi(\boldsymbol{h}^\ell(\boldsymbol{x},s))^\top \tag{129}$$

From this equation, we can again unfold the recurrence for $\boldsymbol{W}^\ell(t)$ and write the recurrence equations for $\boldsymbol{h}^\ell, \boldsymbol{g}^\ell$, verifying that we get feature learning updates of the correct scale

$$\boldsymbol{h}^{\ell+1}(\boldsymbol{x},t) = \boldsymbol{h}^\ell(\boldsymbol{x},t) + \frac{1}{\sqrt{L}}\boldsymbol{\chi}^{\ell+1}(\boldsymbol{x},t)$$
$$+ \frac{\eta_0\gamma_0(1-\alpha)}{L}\sum_{t'<t}\sum_{s<t'}\alpha^{t'-s}\mathbb{E}_{\boldsymbol{x}'}\Phi^\ell(\boldsymbol{x},\boldsymbol{x}',t,s)\Delta(\boldsymbol{x}',s)\boldsymbol{g}^{\ell+1}(\boldsymbol{x}',s) \tag{130}$$

where $\boldsymbol{\chi}^{\ell+1}(\boldsymbol{x},t) = \frac{1}{\sqrt{N}}\boldsymbol{W}^\ell(0)\phi(\boldsymbol{h}^\ell(\boldsymbol{x},t))$. We see that this update has the correct scaling structure, the random (uncorrelated across layers) field $\boldsymbol{\chi}$ has a scale of $\frac{1}{\sqrt{L}}$ and will behave as a Brownian motion and the feature learning update with the gradient field (strongly correlated across layers) has the correct $\frac{1}{L}$ scaling. The normal DMFT procedure can now be carried out on the $\boldsymbol{\chi}$ fields and the corresponding $\boldsymbol{\xi}$ fields for the backward pass.

L Weight Decay

Weight decay dynamics can also be analyzed with DMFT. The gradient flow dynamics with weight decay with parameter λ have the form

$$\frac{d}{dt}\boldsymbol{W}^\ell(t) = \frac{\eta_0\gamma_0}{\sqrt{LN}}\,\mathbb{E}_{\boldsymbol{x}}\,\Delta(\boldsymbol{x},t)\boldsymbol{g}^{\ell+1}(\boldsymbol{x},t)\phi(\boldsymbol{h}^\ell(\boldsymbol{x},t))^\top - \eta_0\lambda\boldsymbol{W}^\ell(t) \tag{131}$$

The dependence of $\boldsymbol{W}^\ell(t)$ on the initial condition can be isolated with a simple integrating factor

$$\boldsymbol{W}^\ell(t) = e^{-\eta_0\lambda t}\boldsymbol{W}^\ell(0) + \frac{\eta_0\gamma_0}{\sqrt{NL}}\int ds\, e^{-\eta_0\lambda(t-s)}\mathbb{E}_{\boldsymbol{x}}\,\Delta(\boldsymbol{x},s)\boldsymbol{g}^{\ell+1}(\boldsymbol{x},s)\phi(\boldsymbol{h}^\ell(\boldsymbol{x},s))^\top \tag{132}$$

Computing the forward pass recursion we find

$$\boldsymbol{h}^{\ell+1}(\boldsymbol{x},t) = \boldsymbol{h}^\ell(\boldsymbol{x},t) + \frac{1}{\sqrt{L}}e^{-\eta_0\lambda t}\boldsymbol{\chi}^\ell(\boldsymbol{x},t)$$
$$+ \frac{\eta_0\gamma_0}{L}\int ds\, e^{-\eta_0\lambda(t-s)}\mathbb{E}_{\boldsymbol{x}'}\,\Delta(\boldsymbol{x}',s)\boldsymbol{g}^{\ell+1}(\boldsymbol{x}',s)\Phi^\ell(\boldsymbol{x},\boldsymbol{x}',t,s) \tag{133}$$

This clearly will admit an infinite width and depth limit following the arguments in Appendices D & E. The difference is the presence of factors of $e^{-\eta_0\lambda t}$, which suppress the influence of the initial condition at late time.