Dynamics of Finite Width Kernel and Prediction Fluctuations in Mean Field Neural Networks

Blake Bordelon & Cengiz Pehlevan
John Paulson School of Engineering and Applied Sciences,
Center for Brain Science,
Kempner Institute for the Study of Natural & Artificial Intelligence,
Harvard University
Cambridge MA, 02138
blake_bordelon@g.harvard.edu, cpehlevan@g.harvard.edu

Abstract

We analyze the dynamics of finite width effects in wide but finite feature learning neural networks. Starting from a dynamical mean field theory description of infinite width deep neural network kernel and prediction dynamics, we provide a characterization of the $\mathcal{O}(1/\sqrt{\text{width}})$ fluctuations of the DMFT order parameters over random initializations of the network weights. Our results, while perturbative in width, unlike prior analyses, are non-perturbative in the strength of feature learning. In the lazy limit of network training, all kernels are random but static in time and the prediction variance has a universal form. However, in the rich, feature learning regime, the fluctuations of the kernels and predictions are dynamically coupled with a variance that can be computed self-consistently. In two layer networks, we show how feature learning can dynamically reduce the variance of the final tangent kernel and final network predictions. We also show how initialization variance can slow down online learning in wide but finite networks. In deeper networks, kernel variance can dramatically accumulate through subsequent layers at large feature learning strengths, but feature learning continues to improve the signal-to-noise ratio of the feature kernels. In discrete time, we demonstrate that large learning rate phenomena such as edge of stability effects can be well captured by infinite width dynamics and that initialization variance can decrease dynamically. For CNNs trained on CIFAR-10, we empirically find significant corrections to both the bias and variance of network dynamics due to finite width.

1 Introduction

Learning dynamics of deep neural networks are challenging to analyze and understand theoretically, but recent progress has been made by studying the idealization of infinite-width networks. Two types of infinite-width limits have been especially fruitful. First, the kernel or lazy infinite-width limit, which arises in the standard or neural tangent kernel (NTK) parameterization, gives prediction dynamics which correspond to a linear model [1–5]. This limit is theoretically tractable but fails to capture adaptation of internal features in the neural network, which are thought to be crucial to the success of deep learning in practice. Alternatively, the mean field or μ-parameterization allows feature learning at infinite width [6–9].

With a set of well-defined infinite-width limits, prior theoretical works have analyzed finite networks in the NTK parameterization perturbatively, revealing that finite width both enhances the amount of feature evolution (which is still small in this limit) but also introduces variance in the kernels and the predictions over random initializations [10–15]. Because of these competing effects, in some situations wider networks are better, and in others wider networks perform worse [16].

37th Conference on Neural Information Processing Systems (NeurIPS 2023).

In this paper, we analyze finite-width network learning dynamics in the mean field parameterization. In this parameterization, wide networks are empirically observed to outperform narrow networks [7, 17, 18]. Our results and framework provide a methodology for reasoning about detrimental finite-size effects in such feature-learning neural networks. We show that observable averages involving kernels and predictions obey a well-defined power series in inverse width even in rich training regimes. We generally observe that the leading finite-size corrections to both the bias and variance components of the square loss are increased for narrower networks, and diminish performance. Further, we show that richer networks are closer to their corresponding infinite-width mean field limit. For simple tasks and architectures the leading $\mathcal{O}(1/\text{width})$ corrections to the error can be descriptive, while for large sample size or more realistic tasks, higher order corrections appear to become relevant. Concretely, our contributions are listed below:

1. Starting from a dynamical mean field theory (DMFT) description of infinite-width nonlinear deep neural network training dynamics, we provide a complete recipe for computing fluctuation dynamics of DMFT order parameters over random network initializations during training. These include the variance of the training and test predictions and the $\mathcal{O}(1/\text{width})$ variance of feature and gradient kernels throughout training.
2. We first solve these equations for the lazy limit, where no feature learning occurs, recovering a simple differential equation which describes how prediction variance evolves during learning.
3. We solve for variance in the rich feature learning regime in two-layer networks and deep linear networks. We show richer nonlinear dynamics improve the signal-to-noise ratio (SNR) of kernels and predictions, leading to closer agreement with infinite-width mean field behavior.
4. We analyze in a two-layer model why larger training set sizes in the overparameterized regime enhance finite-width effects and how richer training can reduce this effect.
5. We show that large learning rate effects such as edge-of-stability [19–21] dynamics can be well captured by infinite width theory, with finite size variance accurately predicted by our theory.
6. We test our predictions in Convolutional Neural Networks (CNNs) trained on CIFAR-10 [22]. We observe that wider networks and richly trained networks have lower logit variance as predicted. However, the timescale of training dynamics is significantly altered by finite width even after ensembling. We argue that this is due to a detrimental correction to the mean dynamical NTK.

1.1 Related Works

Infinite-width networks at initialization converge to a Gaussian process with a covariance kernel that is computed with a layerwise recursion [23–26, 13]. In the large but finite width limit, these kernels do not concentrate at each layer, but rather propagate finite-size corrections forward through the network [27–30, 14]. During gradient-based training with the NTK parameterization, a hierarchy of differential equations have been utilized to compute small feature learning corrections to the kernel through training [10–13]. However the higher order tensors required to compute the theory are initialization dependent, and the theory breaks down for sufficiently rich feature learning dynamics. Various works on Bayesian deep networks have also considered fluctuations and perturbations in the kernels at finite width during inference [31, 32]. Other relevant work in this domain are [33–39].

An alternative to standard/NTK parameterization is the mean field or μP-limit where features evolve even at infinite width [6–9, 40–42]. Recent studies on two-layer mean field networks trained online with Gaussian data have revealed that finite networks have larger sensitivity to SGD noise [43, 44]. Here, we examine how finite-width neural networks are sensitive to initialization noise. Prior work has studied how the weight space distribution and predictions converge to mean field dynamics with a dynamical error $\mathcal{O}(1/\sqrt{\text{width}})$ [40, 45], however in the deep case this requires a probability distribution over couplings between adjacent layers. Our analysis, by contrast, focuses on a function and kernel space picture which decouples interactions between layers at infinite width. A starting point for our present analysis of finite-width effects was a previous set of studies [9, 46] which identified the DMFT action corresponding to randomly initialized deep NNs which generates the distribution over kernel and network prediction dynamics. These prior works discuss the possibility of using a finite-size perturbation series but crucially failed to recognize the role of the network prediction fluctuations on the kernel fluctuations which are necessary to close the self-consistent equations in the rich regime. Using the mean field action to calculate a perturbation expansion around DMFT is a long celebrated technique to obtain finite size corrections in physics [47–50] and has been utilized for random untrained recurrent networks [51, 52], and more recently to calculate variance of

feature kernels Θ^{ℓ} at initialization $t = 0$ in deep MLPs or RNNs [53]. We extend these prior studies to the dynamics of training and to probe how feature learning alters finite size corrections.

2 Problem Setup

We consider wide neural networks where the number of neurons (or channels for a CNN) N in each layer is large. For a multi-layer perceptron (MLP), the network is defined as a map from input $x \in \mathbb{R}^D$ to hidden preactivations $h^{\ell} \in \mathbb{R}^N$ in layers $\ell \in \{1, ..., L\}$ and finally output f

$$f = \frac{1}{\gamma N} W^L \cdot \phi(h^L) , \quad h^{\ell+1} = \frac{1}{\sqrt{N}} W^{\ell} \phi(h^{\ell}) , \quad h^1 = \frac{1}{\sqrt{D}} W^0 x , \tag{1}$$

where γ is a scale factor that controls feature learning strength, with large γ leading to rich feature learning dynamics and the limit of small $\gamma \to 0$ (or generally if γ scales as $N^{-\alpha}$ for $\alpha > 0$ as $N \to \infty$, NTK parameterization corresponds to $\alpha = \frac{1}{2}$) gives lazy learning where no features are learned [4, 7, 9]. The parameters $\theta = \{W^0, W^1, ..., W^L\}$ are optimized with gradient descent or gradient flow $\frac{d\theta}{dt} = -N\gamma^2 \nabla_{\theta} \mathcal{L}$ where $\mathcal{L} = \mathbb{E}_{x \in \mathcal{D}} \ell(f(x, \theta), y)$ is a loss computed over dataset $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), \dots (x_P, y_P)\}$. This parameterization and learning rate scaling ensures that $\frac{df}{dt} \sim \mathcal{O}_N(1)$ and $\frac{dh^{\ell}}{dt} = \mathcal{O}_N(\gamma)$ at initialization. This is equivalent to maximal update parameterization (μP)[8], which can be easily extended to other architectures including neural networks with trainable bias parameters as well as convolutional, recurrent, and attention layers [8, 9].

3 Review of Dynamical Mean Field Theory

The infinite-width training dynamics of feature learning neural networks was described by a DMFT in [9, 46]. We first review the DMFT's key concepts, before extending it to get insight into finite-widths. To arrive at the DMFT, one first notes that the training dynamics of such networks can be rewritten in terms of a collection of dynamical variables (or *order parameters*) $q = \text{Vec}\{f(t), \Theta^{\ell}(t, s), G^{\ell}(t, s), ...\}$ [9], which include feature and gradient kernels [9, 54]

$$\Theta^{\ell}(t, s) \equiv \frac{1}{N} \phi(h^{\ell}(t)) \cdot \phi(h^{\ell}(s)) , \quad G^{\ell}(t, s) \equiv \frac{1}{N} g^{\ell}(t) \cdot g^{\ell}(s), \tag{2}$$

where $g^{\ell}(t) = \gamma N \frac{\partial f(t)}{\partial h^{\ell}(t)}$ are the back-propagated gradient signals. Further, for width-N networks the distribution of these dynamical variables across weight initializations (from a Gaussian distribution $\sim \mathcal{N}(0, I)$) is given by $p(q) \propto \exp(NS(q))$, where the action $S(q)$ contains interactions between neuron activations and the kernels at each layer [9].

The DMFT introduced in [9] arises in the $N \to \infty$ limit when $p(q)$ is strongly peaked around the saddle point q_1 where $\frac{\partial S}{\partial q}|_{q_1} = 0$. Analysis of the saddle point equations reveal that the training dynamics of the neural network can be alternatively described by a stochastic process. A key feature of this process is that it describes the training time evolution of the distribution of neuron pre-activations in each layer (informally the histogram of the elements of $h^{\ell}(t)$) where each neuron's pre-activation behaves as an i.i.d. draw from this *single-site* stochastic process. We denote these random processes by $h^{\ell}(t)$. Kernels in (2) are now computed as *averages* over these infinite-width single site processes $\Theta^{\ell}(t, s) = \phi(h^{\ell}(t))\phi(h^{\ell}(s))$, $G^{\ell}(t, s) = g^{\ell}(t)g^{\ell}(s)$, where averages arise from the $N \to \infty$ limit of the dot products in (2). DMFT also provides a set of self-consistent equations that describe the complete statistics of these random processes, which depend on the kernels, as well as other quantities. To make our notation and terminology clearer for a machine learning audience, we provide Table 1 for a definition of the physics terminology in machine learning language.

Order params. q	Action $S(q)$	Propagator	Single Site Density
Concentrating variables	q's log-density	Asymptotic Covariance	Neuron Marginals

Table 1: Relationship between the physics and ML terminology for the central objects in this paper. The q which concentrate at infinite width, but fluctuate at finite width N. This paper is primarily interested in , the asymptotic covariance of the order parameters.

4 Dynamical Fluctuations Around Mean Field Theory

We are interested in going beyond the infinite-width limit to study more realistic finite-width networks. In this regime, the order parameters q fluctuate in a $\mathcal{O}(1/\sqrt{N})$ neighborhood of q_1 [55, 51, 53, 46].

Statistics of these fluctuations can be calculated from a general cumulant expansion (see App. D) [55, 56, 51]. We will focus on the leading-order corrections to the infinite-width limit in this expansion.

Proposition 1 *The finite-width N average of observable $O(q)$ across initializations, which we denote by $\langle O(q)\rangle_N$, admits an expansion of the form whose leading terms are*

$$\langle O(q)\rangle_N = \frac{\int dq \exp\left(NS[q]\right) O(q)}{\int dq \exp\left(NS[q]\right)} = \langle O(q)\rangle_1 + N\left[\langle V(q)O(q)\rangle_1 - \langle V(q)\rangle_1 \langle O(q)\rangle_1\right] + \ldots, \tag{3}$$

where $\langle\rangle_1$ denotes an average over the Gaussian distribution $q \sim \mathcal{N}\left(q_1, -\frac{1}{N}\left[\nabla_q^2 S[q_1]\right]^{-1}\right)$ and the function $V(q) \equiv S(q) - S(q_1) - \frac{1}{2}(q-q_1)^\top \nabla_q^2 S(q_1)(q-q_1)$ contains cubic and higher terms in the Taylor expansion of S around q_1. The terms shown include all the leading and sub-leading terms in the series in powers of $1/N$. The terms in ellipses are at least $\mathcal{O}(N^{-1})$ suppressed compared to the terms provided.

The proof of this statement is given in App. D. The central object to characterize finite size effects is the unperturbed covariance (the *propagator*): $\Delta = -\left[\nabla^2 S(q_1)\right]^{-1}$. This object can be shown to capture leading order fluctuation statistics $\left\langle (q-q_1)(q-q_1)^\top \right\rangle_N = \frac{1}{N}\Delta + \mathcal{O}(N^{-2})$ (App. D.1), which can be used to reason about, for example, expected square error over random initializations. Correction terms at finite width may give a possible explanation of the superior performance of wide networks at fixed γ [7, 17, 18]. To calculate such corrections, in App. E, we provide a complete description of Hessian $\nabla_q^2 S(q)$ and its inverse (the propagator) for a depth-L network. This description constitutes one of our main results. The resulting expressions are lengthy and are left to App. E. Here, we discuss them at a high level. Conceptually there are two primary ingredients for obtaining the full propagator:

- Hessian sub-blocks κ which describe the *uncoupled variances* of the kernels, such as

$$\kappa^{\ell}(t,s,t^0,s^0) \equiv \phi(h^{\ell}(t))\phi(h^{\ell}(s))\phi(h^{\ell}(t^0))\phi(h^{\ell}(s^0)) - \Phi^{\ell}(t,s)\Phi^{\ell}(t^0,s^0) \tag{4}$$

  Similar terms also appear in other studies on finite width Bayesian inference [13, 31, 32] and in studies on kernel variance at initialization [27, 14, 29, 53].

- Blocks which capture the *sensitivity* of field averages to pertubations of order parameters, such as

$$D^{\ell,\ell'}(t,s,t^0,s^0) \equiv \frac{\partial\, \phi(h^{\ell}(t))\phi(h^{\ell}(s))}{\partial \Phi^{\ell'}(t^0,s^0)}, \quad D^{G^{\ell}}(t,s,t^0) \equiv \frac{\partial\, g^{\ell}(t) g^{\ell}(s)}{\partial \Delta^{\ell}(t^0)}, \tag{5}$$

  where $\Delta^{\ell}(t) = -\frac{\partial^{\ell}(f,y)}{\partial f}\big|_{f^{\ell}(t)}$ are error signal for each data point.

Abstractly, we can consider the uncoupled variances κ as "sources" of finite-width noise for each order parameter and the D blocks as summarizing a directed causal graph which captures how this noise propagates in the network (through layers and network predictions). In Figure 1, we illustrate this graph showing directed lines that represent causal influences of order parameters on fields and vice versa. For instance, if Φ^{ℓ} were perturbed, $D^{\ell+1,\ell}$ would quantify the resulting perturbation to $\Phi^{\ell+1}$ through the fields $h^{\ell+1}$.

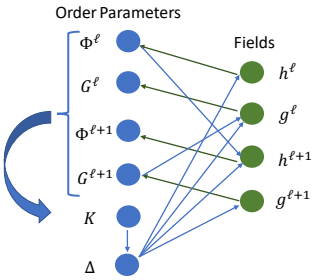


Figure 1: The directed causal graph between DMFT order parameters (blue) and fields (green) defines the D tensors of our theory. Each arrow represents a causal dependence. K denotes the NTK.

In App. E, we calculate κ and D tensors, and show how to use them to calculate the propagator. As an example of our results:

Proposition 2 *Partition q into primal $q_1 = \mathrm{Vec}\{f^{\ell}(t), \Phi^{\ell}(t,s)\ldots\}$ and conjugate variables $q_2 = \mathrm{Vec}\{\hat{f}^{\ell}(t), \hat{\Phi}^{\ell}(t,s)\ldots\}$. Let $\Sigma = \frac{\partial^2}{\partial q_2 \partial q_2^\top} S[q_1, q_2]$ and $D = \frac{\partial^2}{\partial q_2 \partial q_1^\top} S[q_1, q_2]$, then the propagator for q_1 has the form $\Delta_{q_1} = D^{-1}\left(D^{-1}\right)^\top$ (App E). The variables q_1 are related to network observables, while conjugates q_2 arise as Lagrange multipliers in the DMFT calculation. From the propagator Δ_{q_1} we can read off the variance of network observables such as $N\mathrm{Var}(f) \sim \Sigma_f$.*

The necessary order parameters for calculating the fluctuations are obtained by solving the DMFT using numerical methods introduced in [9]. We provide a pseudocode for this procedure in App. F. We proceed to solve the equations defining in special cases which are illuminating and numerically feasible including lazy training, two layer networks and deep linear NNs.

5 Lazy Training Limit

To gain some initial intuition about why kernel fluctuations alter learning dynamics, we first analyze the static kernel limit $\gamma \to 0$ where features are frozen. To prevent divergence of the network in this limit, we use a background subtracted function $\tilde{f}(x, \ ) = f(x, \ ) - f(x, \ _0)$ which is identically zero at initialization [4]. For mean square error, the $N \to \infty$ and $\gamma \to 0$ limit is governed by $\frac{@\tilde{f}(x)}{@t} = \mathsf{E}_{x^0 \ D} \Delta(x^0) K(x, x^0)$ with $\Delta(x) = y(x) - \tilde{f}(x)$ (for MSE) and K is the static (finite width and random) NTK. The finite-N initial covariance of the NTK has been analyzed in prior works [27, 13, 14], which reveal a dependence on depth and nonlinearity. Since the NTK is static in the $\gamma \to 0$ limit, it has constant initialization variance through training. Further, all sensitivity blocks of the Hessian involving the kernels and the prediction errors (such as the D) vanish. We represent the covariance of the NTK as $\kappa(x_1, x_2, x_2, x_3) = N\mathrm{Cov}(K(x_1, x_2), K(x_3, x_4))$. To identify the dynamics of the error covariance, we relate K, the finite width NTK, to K_1 which is the deterministic infinite width NTK K_1. We consider the eigendecomposition of the infinite-width NTK $K_1(x, x^0) = \sum_k \lambda_k \psi_k(x) \psi_k(x^0)$ with respect to the training distribution $\mathcal{D}$, and decompose κ in this basis.

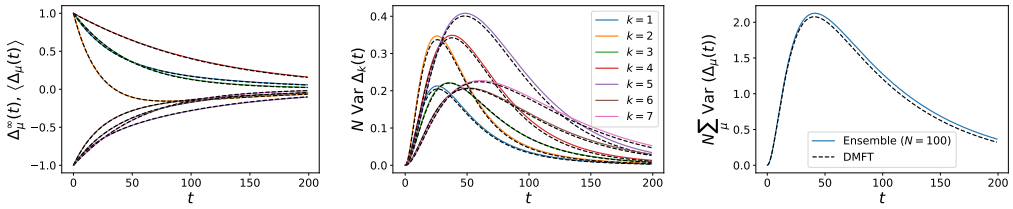
$$\kappa_{k`mn} = \langle \kappa(x_1, x_2, x_3, x_4) \psi_k(x_1) \psi_{\cdot}(x_2) \psi_n(x_3) \psi_m(x_4) \rangle_{x_1; x_2; x_3; x_4 \ D}, \tag{6}$$

where averages are computed over the training distribution $\mathcal{D}$.

Proposition 3 *For MSE loss, the prediction error covariance* $(t, s) = N\mathrm{Cov}_0(\ (t), \ (s))$ *satisfies a differential equation (App. H)*

$$\frac{\partial}{\partial t} + \lambda_k \quad \frac{\partial}{\partial s} + \lambda_{\cdot} \quad \Sigma_{k\cdot}(t, s) = \sum_{nm} \kappa_{km`n} \Delta_m^1(t) \Delta_n^1(s), \tag{7}$$

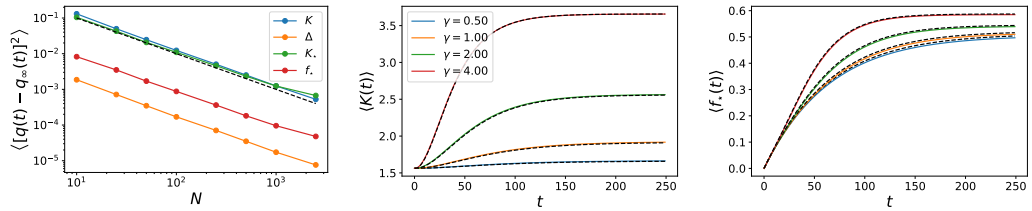
where $\Delta_k^1(t) \equiv \exp(-\lambda_k t) \langle \psi_k(x) y(x) \rangle_x$ *are the errors at infinite width for eigenmode* k.



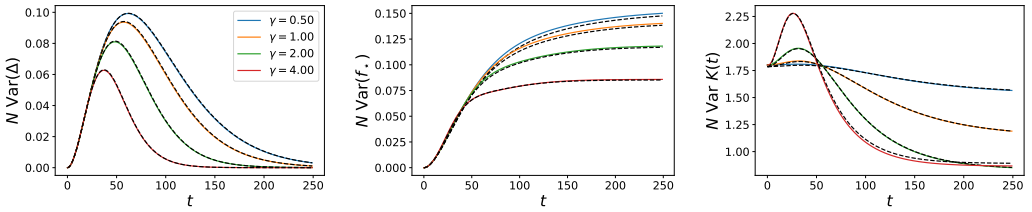
(a) Average Prediction Errors (b) Top Eigenmodes Variance (c) Total Train Variance

Figure 2: We show the accuracy of the lazy-limit ODE in equation (where) comapared to a two-layer finite width $N = 100$ ReLU network trained with $\gamma = 0.05$ on $P = 10$ random training data points. (a) The average dynamics over an ensemble of $E = 500$ networks (solid colors) compared to the infinite width predictions (dashed black). (b) The predicted finite size variance for each eigenmode of the error $\Delta_k(t) = \ (t) \cdot \ _k$. These are not ordered simply by magnitude of eigenvalues or the target projections $y_k = y \cdot \ _k$, but rather depend on all eigenvalue gaps $\lambda_k - \lambda_\ell$ for $k \neq \ell$ and also the $\kappa_{k`nm}$ tensor. (c) The total variance for all training points N $\mathrm{Var}\Delta \ (t) = N \ _k \mathrm{Var}\Delta_k(t)$ is also well predicted by the DMFT propagator equations.

An example verifying these dynamics is provided in Figure 2. In the case where the target is an eigenfunction $y = \psi_k$, the covariance has the form $\Sigma_{k`}(t, s) = \kappa_{k`k \ k} \frac{\exp(\ _k \ (t+s))}{(\ _k \ _k)(\ _{\cdot} \ _k)}$. If the kernel is rank one with eigenvalue λ, then the dynamics have the simple form $\Sigma \ (t, s) = \kappa y^2 \ t \ s \ e^{\ (t+s)}$. We note that similar terms appear in the prediction dynamics obtained by truncating the Neural Tangent Hierarchy [10, 11], however those dynamics concerned small feature learning



(a) Deviation from DMFT vs N (b) Average NTK vs DMFT (c) Average test prediction



(d) Train error variance (e) Test Prediction Variance (f) Kernel Variance Dynamics

Figure 3: An ensemble of $E = 1000$ two layer $N = 256$ tanh networks trained on a single training point. Dashed black lines are DMFT predictions. (a) The square deviation from the infinite width DMFT scales as $\mathcal{O}(1/N)$ for all order parameters. (b) The ensemble average NTK $\langle K(t) \rangle$ (solid colors) and (c) ensemble average test point predictions $f_?(t)$ for a point with $\frac{x \cdot x_?}{D} = 0.5$ closely follow the infinite width predictions (dashed black). (d) The variance (estimated over the ensemble) of the train error $\Delta(t) = y - f(t)$ initially increases and then decreases as the training point is fit. (e) The variance of $f_?$ increases with time but decreases with γ. (f) The variance of the NTK during feature learning experiences a transient increase before decreasing to a lower value.

corrections rather than from initialization variance (App. H.1). Corrections to the mean $\langle \Delta \rangle$ are analyzed in App. H.2. We find that the variance and mean correction dynamics involves non-trivial coupling across eigendirections with a mixture of exponentials with timescales $\{\lambda_k{}^1\}$.

6 Rich Regime in Two-Layer Networks

In this section, we analyze how feature learning alters the variance through training. We show a denoising effect where the signal to noise ratios of the order parameters improve with feature learning.

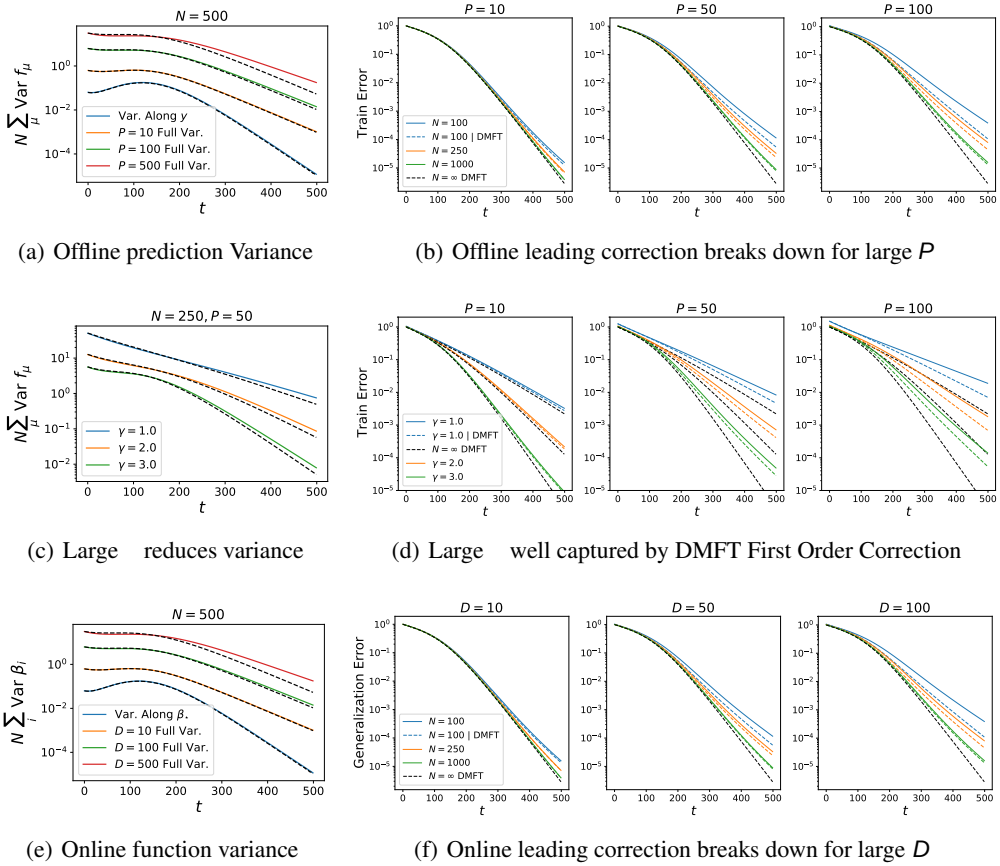
6.1 Kernel and Error Coupled Fluctuations on Single Training Example

In the rich regime, the kernel evolves over time but inherits fluctuations from the training errors . To gain insight, we first study a simplified setting where the data distribution is a single training example x and single test point $x_?$ in a two layer network. We will track $\Delta(t) = y - f(x, t)$ and the test prediction $f_?(t) = f(x_?, t)$. To identify the dynamics of these predictions we need the NTK $K(t)$ on the train point, as well as the train-test NTK $K_?(t)$. In this case, all order parameters can be viewed as scalar functions of a single time index (unlike the deep network case, see App. E).

Proposition 4 *Computing the Hessian of the DMFT action and inverting (App. I), we obtain the following covariance for* $q_1 = Vec\{\Delta(t), f_?(t), K(t), K_?(t)\}_{t \in \mathbb{R}_+}$

$$q_1 = \begin{bmatrix} I + K & 0 & 0 & 0 \\ -K_? & I & 0 & 0 \\ -D & 0 & I & 0 \\ -D_? & 0 & 0 & I \end{bmatrix}^{-1} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} I + K & 0 & 0 & 0 \\ -K_? & I & 0 & 0 \\ -D & 0 & I & 0 \\ -D_? & 0 & 0 & I \end{bmatrix}^{-1>}, \tag{8}$$

where $[K](t,s) = \Theta(t-s)K(s)$, $[E](t,s) = \Theta(t-s)\Delta(s)$ *are Heaviside step functions and* $D(t,s) = \frac{\partial}{\partial(s)}(\phi(h(t))^2 + g(t)^2)$ *and* $D_?(t,s) = \frac{\partial}{\partial(s)}(\phi(h(t))\phi(h_?(t)) + g(t)g_?(t))$



(a) Offline prediction Variance (b) Offline leading correction breaks down for large P

(c) Large reduces variance (d) Large well captured by DMFT First Order Correction

(e) Online function variance (f) Online leading correction breaks down for large D

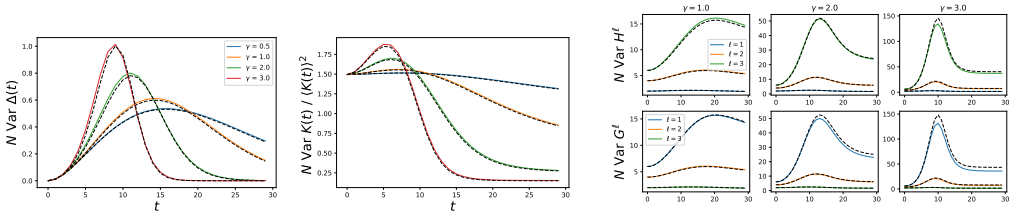
Figure 4: Large input dimension or multiple samples amplify finite size effects in a simple two layer model with unstructured data. Black dashed lines are theory. (a) The variance of offline learning with P training examples in a two layer linear network. (b) The leading perturbative approximation to the train error breaks down when samples P becomes comparable to N. (c)-(d) Richer training reduces variance. (e)-(f) Online learning in a depth 2 linear network has identical dynamics and finite width fluctuations, but with predictor variance $\sim D/N$ for input dimension D (Appendix K).

quantify sensitivity of the kernel to perturbations in the error signal $\Delta(s)$. Lastly κ and $\kappa_?$ are the uncoupled variances of $K(t)$ and $K_{??}(t)$ and $\kappa_?$ is the uncoupled covariance of $K(t), K_?(t)$.

In Fig. 3, we plot the resulting theory (diagonal blocks of q_1 from Equation 8) for two layer neural networks. As predicted by theory, all average squared deviations from the infinite width DMFT scale as $\mathcal{O}(N^{-1})$. Similarly, the average kernels $\langle K \rangle$ and test predictions $\langle f_? \rangle$ change by a larger amount for larger γ (equation (79)). The experimental variances also match the theory quite accurately. The variance of the train error $\Delta(t)$ peaks earlier and at a lower value for richer training, but all variances go to zero at late time as the model approaches the interpolation condition $\Delta = 0$. As $\gamma \to 0$ the curve approaches $N \text{ Var}(\Delta(t)) \sim \kappa\, y^2\, t^2\, e^{-2t}$, where κ is the initial NTK variance (see Section 5). While the train prediction variance goes to zero, the test point prediction does not, with richer networks reaching a lower asymptotic variance. We suspect this dynamical effect could explain lower variance observed in feature learning networks compared to lazy networks [7, 18]. In Fig. A.1, we show that the reduction in variance is not due to a reduction in the uncoupled variance $\kappa(t, s)$, which increases in γ. Rather the reduction in variance is driven by the coupling of perturbations across time.

6.2 Offline Training with Multiple Samples or Online Training in High Dimension

In this section we go beyond the single sample equations of the prior section and explore training with multiple P examples. In this case, we have training errors $\{\Delta^{\mu}(t)\}_{\mu=1}^{P}$ and multiple kernel entries $K^{\mu\nu}(t)$ (App. E). Each of the errors $\Delta^{\mu}(t)$ receives a $\mathcal{O}(N^{-1/2})$ fluctuation, the training error $\sum_{\mu}^{P} \Delta^2$ has an additional variance on the order of $\mathcal{O}(\frac{P}{N})$. In the case of two-layer linear



(a) Prediction Variance (b) NTK variance (c) Variance by Layer

Figure 5: Depth 4 linear network with single training point. Black dashed lines are theory. (a) The variance of the training error along the task relevant subspace. We see that unlike the two layer model, more feature learning can lead to larger peaks in the finite size variance. (b) The variance of the NTK in the task relevant subspace. When properly normalized against the square of the mean $\langle K(t) \rangle^2$, the final NTK variance decreases with feature learning. (c) The gap in feature kernel variance across different layers of the network is amplified by feature learning strength γ.

networks trained on whitened data ($\frac{1}{D} \boldsymbol{X} \cdot \boldsymbol{X} = \delta$), the equations for the propagator simplify and one can separately solve for the variance of $(t) \in \mathbb{R}^P$ along signal direction $\boldsymbol{y} \in \mathbb{R}^P$ and along each of the $P-1$ orthogonal directions (App. J). At infinite width, the task-orthogonal component $_?$ vanishes and only the signal dimension $\Delta_y(t)$ evolves in time with differential equation [9, 46]

$$\frac{d}{dt}\Delta_y(t) = 2 \sqrt{1 + \gamma^2 (y - \Delta_y(t))^2}\, \Delta_y(t) \,, \qquad _?(t) = 0. \tag{9}$$

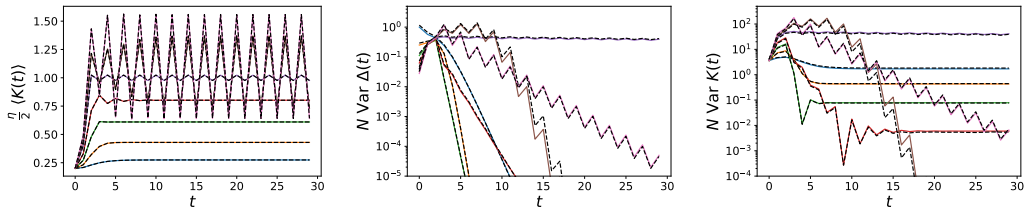
However, at finite width, both the $\Delta_y(t)$ and the $P-1$ orthogonal variables $_?$ inherit initialization variance, which we represent as $\Sigma_y(t,s)$ and $\Sigma_?(t,s)$. In Fig. 4 (a)-(b) we show this approximate solution $|(t)|^2 \sim \Delta_y(t)^2 + \frac{2}{N}\Delta_y^1(t)\Delta_y(t) + \frac{1}{N}\Sigma_y(t,t) + \frac{(P-1)}{N}\Sigma_?(t,t) + \mathcal{O}(N^{-2})$ across varying γ and varying P (see Appendix J for Σ_y and $\Sigma_?$ formulas). We see that variance of train point predictions $f(t)$ increases with the total number of points despite the signal of the target vector y^2 being fixed. In this model, the bias correction $\frac{2}{N}\Delta_y^1(t)\Delta_y(t)$ is always $\mathcal{O}(1/N)$ but the variance correction is $\mathcal{O}(P/N)$. The fluctuations along the $P-1$ orthogonal directions begin to dominate the variance at large P. Fig. 4 (b) shows that as P increases, the leading order approximation breaks down as higher order terms become relevant. Analysis for online training reveals identical fluctuation statistics, but with variance that scales as $\sim D/N$ (Appendix K) as we verify in Figure 4 (e)-(f).

7 Deep Networks

In networks deeper than two layers, the DMFT propagator has complicated dependence on non-diagonal (in time) entries of the feature kernels (see App. E). This leads to Hessian blocks with four time and four sample indices such as $D\hat{}(t,s,t^0,s^0) = \frac{@}{@\,{}^{1}(t^0;s^0)}\, \phi(h\hat{}(t))\phi(h\hat{}(s))$, rendering any numerical calculation challenging. However, in deep linear networks trained on whitened data, we can exploit the symmetry in sample space and the Gaussianity of preactivation features to exactly compute derivatives without Monte Carlo sampling as we discuss in App. L. An example set of results for a depth 4 network is provided in Fig. 5. The variance for the feature kernels $H\hat{}$ accumulate finite size variance by layer along the forward pass and the gradient kernels $G\hat{}$ accumulate variance on the backward pass. The SNR of the kernels $\frac{hH\,i^2}{N\,\mathrm{Var}(H)}$ improves with feature learning, suggesting that richer networks will be better modeled by their mean field limits. Examples of the off-diagonal correlations obtained from the propagator are provided in App. Fig. A.3.

8 Variance can be Small Near Edge of Stability

In this section, we move beyond the gradient flow formalism and ask what large step sizes do to finite size effects. Recent studies have identified that networks trained at large learning rates can be qualitatively different than networks in the gradient flow regime, including the catapult [57] and edge of stability (EOS) phenomena [19–21]. In these settings, the kernel undergoes an initial scale



(a) Mean Kernel Dynamics (b) Error Variance (c) Kernel Variance

Figure 6: Edge of stability effects do not imply deviations from infinite width behavior. Black dashed lines are theory. (a) The average kernel over an ensemble of several $N = 500$ NNs (solid color). For small γ, the kernel reaches its asymptote before hitting the edge of stability. For large γ, the kernel increases and then oscillates around $2/\eta$. (b)-(c) Remarkably variance due to finite size can *reduce* during training (for γ smaller and larger than the critical value $\sim 1/\eta$), showing that infinite width DMFT can be predictive of finite NNs trained with large learning rate.

growth before exhibiting either a recovery or a clipping effect. In this section, we explore whether these dynamics are highly sensitive to initialization variance or if finite networks are well captured by mean field theory. Following [57], we consider two layer networks trained on a single example $|\boldsymbol{x}|^2 = D$ and $y = 1$. We use learning rate η and feature learning strength γ. The infinite width mean field equations for the prediction f_t and the kernel K_t are (App. M)

$$f_{t+1} = f_t + \eta K_t \Delta_t + \eta^2 \gamma^2 f_t \Delta_t^2 \ , \ K_{t+1} = K_t + 4\eta\gamma^2 f_t \Delta_t + \eta^2 \gamma^2 \Delta_t^2 K_t. \tag{10}$$

For small η, the equations are well approximated by the gradient flow limit and for small γ corresponds to a discrete time linear model. For large $\eta\gamma > 1$, the kernel K progressively sharpens (increases in scale) until it reaches $2/\eta$ and then oscillates around this value. It may be expected that near the EOS, the large oscillations in the kernels and predictions could lead to amplified finite size effects, however, we show in Fig. 6 that the leading order propagator elements decrease even after reaching the EOS threshold, indicating *reduced* disagreement between finite and infinite width dynamics.

9 Finite Width Alters Bias, Training Rate, and Variance in Realistic Tasks

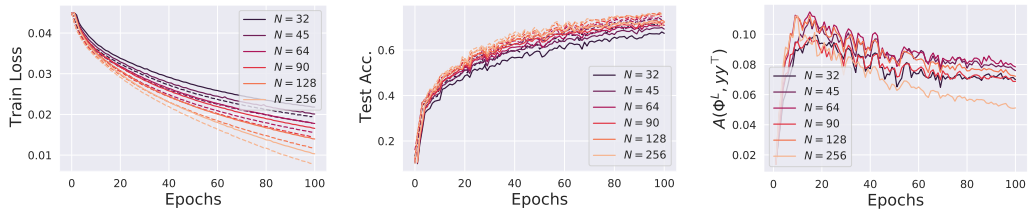
To analyze the effect of finite width on neural network dynamics during realistic learning tasks, we studied a vanilla depth-6 ReLU CNN trained on CIFAR-10 (experimental details in App. B, G.2) In Fig. 7, we train an ensemble of $E = 8$ independently initialized CNNs of each width N. Wider networks not only have better performance for a single model (solid), but also have lower bias (dashed), measured with ensemble averaging of the logits. Because of faster convergence of wide networks, we observe wider networks have higher variance, but if we plot variance at fixed ensembled training accuracy, wider networks have consistently lower variance (Fig. 7(d)).

We next seek an explanation for why wider networks after ensembling trains at a faster *rate*. Theoretically, this can be rationalized by a finite-width alteration to the ensemble averaged NTK, which governs the convergence timescale of the ensembled predictions (App. G.1). Our analysis in App. G.1 suggests that the rate of convergence receives a finite size correction with leading correction $\mathcal{O}(N^{-1})$ G.2. To test this hypothesis, we fit the ensemble training loss curve to exponential function $\mathcal{L} \approx C \exp(-R_N t)$ where C is a constant. We plot the fit R_N as a function of N^{-1} result in Fig. 7(e). For large N, we see the leading behavior is linear in N^{-1}, but begins to deviate at small N as a quadratic function of N^{-1}, suggesting that second order effects become relevant around $N \lesssim 100$.

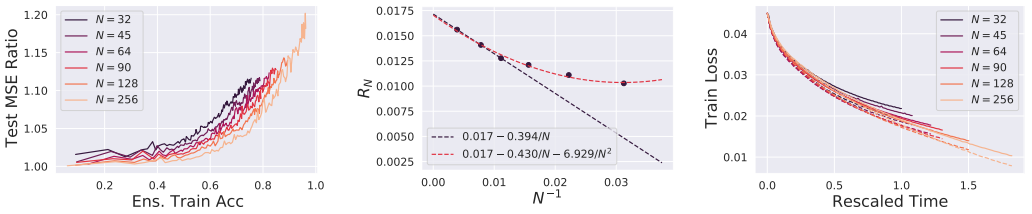
In App. Fig. A.4, we train a smaller subset of CIFAR-10 where we find that R_N is well approximated by a $\mathcal{O}(N^{-1})$ correction, consistent with the idea that higher sample size drives the dynamics out of the leading order picture. We also analyze the effect of γ on variance in this task. In App. Fig. A.5, we train $N = 64$ models with varying γ. Increased γ reduces variance of the logits and alters the representation (measured with kernel-task alignment), the training and test accuracy are roughly insensitive to the richness γ in the range we considered.

10 Discussion

We studied the leading order fluctuations of kernels and predictions in mean field neural networks. Feature learning dynamics can reduce undesirable finite size variance, making finite networks order



(a) Train Loss Across Widths (b) Test Acc. Across Widths (c) Alignment Dynamics



(d) Test MSE Ensembling Ratio (e) Training Rates vs N (f) Train Acc. Rescaled Time

Figure 7: Depth 6 CNN trained on CIFAR-10 for different widths N with richness $\gamma = 0.2$, $E = 8$ ensembles. (a)-(b) For this range of widths, we find that smaller networks perform worse in train and test error, not only in terms of the single models (solid) but also in terms of bias (dashed). The delayed training of ensembled finite width models indicates that the correction to the mean order parameters (App. G) is non-negligible. (c) Alignment of the average kernel to test labels is also not conserved across width. (d) The ratio of the test MSE for a single model to the ensembled logit MSE. (e) The fitted rate R_N of training width N models as a function of N^{-1}. We rescale the time axis by R_N to allow for a fair comparison of prediction variance for networks at comparable performance levels. (f) In rescaled time, ensembled network training losses (dashed) are coincident.

parameters closer to the infinite width limit. In several toy models, we revealed some interesting connections between the influence of feature learning, depth, sample size, and large learning rate and the variance of various DMFT order parameters. Lastly, in realistic tasks, we illustrated that bias corrections can be significant as rates of learning can be modified by width. Though our full set of equations for the leading finite size fluctuations are quite general in terms of network architecture and data structure, they are only derived at the level of rigor of physics rather than a formally rigorous proof which would need several additional assumptions to make the perturbation expansion properly defined. Further, the leading terms in our perturbation series involving only does not capture the complete finite size distribution defined in Eq. (3), especially as the sample size becomes comparable to the width. It would be interesting to see if proportional limits of the rich training regime where samples and width scale linearly can be examined dynamically [58]. Future work could explore in greater detail the higher order contributions from averages involving powers of $V(\boldsymbol{q})$ by examining cubic and higher derivatives of S in Eq. (3). It could also be worth examining in future work how finite size impacts other biologically plausible learning rules, where the effective NTK can have asymmetric (over sample index) fluctuations [46]. Also of interest would be computing the finite width effects in other types of architectures, including residual networks with various branch scalings [59, 60]. Further, even though we expect our perturbative expressions to give a precise asymptotic description of finite networks in mean field/μP, the resulting expressions are not realistically computable in deep networks trained on large dataset size P for long times T since the number of Hessian entries scales as $\mathcal{O}(T^4P^4)$ and a matrix of this size must be stored in memory and inverted in the general case. Future work could explore solveable special cases such as high dimensional limits.

Code Availability

Code to reproduce the experiments in this paper is provided at `https://github.com/Pehlevan-Group/dmft_fluctuations`. Details about numerical methods and computational implementation can be found in Appendices F and N.

Acknowledgements

CP is supported by NSF Award DMS-2134157, NSF CAREER Award IIS-2239780, and a Sloan Research Fellowship. BB is supported by a Google PhD research fellowship and NSF Award DMS-2134157. This work has been made possible in part by a gift from the Chan Zuckerberg Initiative Foundation to establish the Kempner Institute for the Study of Natural and Artificial Intelligence. The computations in this paper were run on the FASRC cluster supported by the FAS Division of Science Research Computing Group at Harvard University. BB thanks Alex Atanasov, Jacob Zavatone-Veth for their comments on this manuscript and Boris Hanin, Greg Yang, Mufan Bill Li and Jeremy Cohen for helpful discussions.

References

[1] Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 8571–8580. Curran Associates, Inc., 2018.

[2] Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *Advances in Neural Information Processing Systems*, 32, 2019.

[3] Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *Advances in neural information processing systems*, 32, 2019.

[4] Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. *Advances in Neural Information Processing Systems*, 32, 2019.

[5] Greg Yang and Etai Littwin. Tensor programs iib: Architectural universality of neural tangent kernel training dynamics. In *International Conference on Machine Learning*, pages 11762–11772. PMLR, 2021.

[6] 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*, pages 2388–2464. PMLR, 2019.

[7] Mario Geiger, Stefano Spigler, Arthur Jacot, and Matthieu Wyart. Disentangling feature and lazy training in deep neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(11):113301, 2020.

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

[9] Blake Bordelon and Cengiz Pehlevan. Self-consistent dynamical field theory of kernel evolution in wide neural networks. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[10] Jiaoyang Huang and Horng-Tzer Yau. Dynamics of deep neural networks and neural tangent hierarchy. In *International conference on machine learning*, pages 4542–4551. PMLR, 2020.

[11] Ethan Dyer and Guy Gur-Ari. Asymptotics of wide networks from feynman diagrams. In *International Conference on Learning Representations*, 2020.

[12] Anders Andreassen and Ethan Dyer. Asymptotics of wide convolutional neural networks. *arXiv preprint arXiv:2008.08675*, 2020.

[13] Daniel A Roberts, Sho Yaida, and Boris Hanin. *The principles of deep learning theory*. Cambridge University Press Cambridge, MA, USA, 2022.

[14] Boris Hanin. Random fully connected neural networks as perturbatively solvable hierarchies. *arXiv preprint arXiv:2204.01058*, 2022.

[15] Sho Yaida. Meta-principled family of hyperparameter scaling strategies. *arXiv preprint arXiv:2210.04909*, 2022.

[16] Jaehoon Lee, Samuel Schoenholz, Jeffrey Pennington, Ben Adlam, Lechao Xiao, Roman Novak, and Jascha Sohl-Dickstein. Finite versus infinite neural networks: an empirical study. *Advances in Neural Information Processing Systems*, 33:15156–15172, 2020.

[17] Greg 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, 2021.

[18] Alexander Atanasov, Blake Bordelon, Sabarish Sainathan, and Cengiz Pehlevan. The onset of variance-limited behavior for networks in the lazy and rich regimes. In *The Eleventh International Conference on Learning Representations*, 2023.

[19] Jeremy Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. In *International Conference on Learning Representations*, 2021.

[20] Alex Damian, Eshaan Nichani, and Jason D. Lee. Self-stabilization: The implicit bias of gradient descent at the edge of stability. In *The Eleventh International Conference on Learning Representations*, 2023.

[21] Atish Agarwala, Fabian Pedregosa, and Jeffrey Pennington. Second-order regression models exhibit progressive sharpening to the edge of stability. *arXiv preprint arXiv:2210.04860*, 2022.

[22] Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research).

[23] Radford M Neal. Priors for infinite networks. In *Bayesian Learning for Neural Networks*, pages 29–53. Springer, 1996.

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

[25] Jaehoon Lee, Jascha Sohl-dickstein, Jeffrey Pennington, Roman Novak, Sam Schoenholz, and Yasaman Bahri. Deep neural networks as gaussian processes. In *International Conference on Learning Representations*, 2018.

[26] Alexander G. de G. Matthews, Jiri Hron, Mark Rowland, Richard E. Turner, and Zoubin Ghahramani. Gaussian process behaviour in wide deep neural networks. In *International Conference on Learning Representations*, 2018.

[27] Boris Hanin and Mihai Nica. Finite depth and width corrections to the neural tangent kernel. In *International Conference on Learning Representations*, 2020.

[28] 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.

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

[30] Jacob Zavatone-Veth and Cengiz Pehlevan. Exact marginal prior distributions of finite bayesian neural networks. *Advances in Neural Information Processing Systems*, 34:3364–3375, 2021.

[31] 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, 2021.

[32] Gadi Naveh, Oded Ben David, Haim Sompolinsky, and Zohar Ringel. Predicting the outputs of finite deep neural networks trained with noisy gradients. *Physical Review E*, 104(6):064301, 2021.

[33] Adam X. Yang, Maxime Robeyns, Edward Milsom, Ben Anson, Nandi Schoots, and Laurence Aitchison. A theory of representation learning gives a deep generalisation of kernel methods. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 39380–39415. PMLR, 23–29 Jul 2023.

[34] Inbar Seroussi, Gadi Naveh, and Zohar Ringel. Separation of scales and a thermodynamic description of feature learning in some cnns. *Nature Communications*, 14(1):908, 2023.

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

[36] Jacob A Zavatone-Veth and Cengiz Pehlevan. Depth induces scale-averaging in overparameterized linear bayesian neural networks. *55th Asilomar Conference on Signals, Systems, and Computers*, 2021.

[37] Jacob A Zavatone-Veth, William L Tong, and Cengiz Pehlevan. Contrasting random and learned features in deep bayesian linear regression. *Physical Review E*, 105(6):064118, 2022.

[38] Gadi Naveh and Zohar Ringel. A self consistent theory of gaussian processes captures feature learning effects in finite cnns. *Advances in Neural Information Processing Systems*, 34, 2021.

[39] Jacob A Zavatone-Veth, Abdulkadir Canatar, Benjamin S Ruben, and Cengiz Pehlevan. Asymptotics of representation learning in finite bayesian neural networks*. *Journal of Statistical Mechanics: Theory and Experiment*, 2022(11):114008, nov 2022.

[40] Grant Rotskoff and Eric Vanden-Eijnden. Parameters as interacting particles: long time convergence and asymptotic error scaling of neural networks. *Advances in neural information processing systems*, 31, 2018.

[41] 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.

[42] Dyego Araújo, Roberto I Oliveira, and Daniel Yukimura. A mean-field limit for certain deep neural networks. *arXiv preprint arXiv:1906.00193*, 2019.

[43] Rodrigo Veiga, Ludovic Stephan, Bruno Loureiro, Florent Krzakala, and Lenka Zdeborova. Phase diagram of stochastic gradient descent in high-dimensional two-layer neural networks. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[44] Luca Arnaboldi, Ludovic Stephan, Florent Krzakala, and Bruno Loureiro. From high-dimensional amp; mean-field dynamics to dimensionless odes: A unifying approach to sgd in two-layers networks. In Gergely Neu and Lorenzo Rosasco, editors, *Proceedings of Thirty Sixth Conference on Learning Theory*, volume 195 of *Proceedings of Machine Learning Research*, pages 1199–1227. PMLR, 12–15 Jul 2023.

[45] Huy Tuan Pham and Phan-Minh Nguyen. Limiting fluctuation and trajectorial stability of multilayer neural networks with mean field training. *Advances in Neural Information Processing Systems*, 34:4843–4855, 2021.

[46] Blake Bordelon and Cengiz Pehlevan. The influence of learning rule on representation dynamics in wide neural networks. In *The Eleventh International Conference on Learning Representations*, 2023.

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

[48] Moshe Moshe and Jean Zinn-Justin. Quantum field theory in the large n limit: A review. *Physics Reports*, 385(3-6):69–228, 2003.

[49] Jean Zinn-Justin. *Quantum field theory and critical phenomena*, volume 171. Oxford university press, 2021.

[50] Carson C Chow and Michael A Buice. Path integral methods for stochastic differential equations. *The Journal of Mathematical Neuroscience (JMN)*, 5:1–35, 2015.

[51] Moritz Helias and David Dahmen. *Statistical Field Theory for Neural Networks*. Springer International Publishing, 2020.

[52] A Crisanti and H Sompolinsky. Path integral approach to random neural networks. *Physical Review E*, 98(6):062120, 2018.

[53] 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.

[54] Yizhang Lou, Chris E Mingard, and Soufiane Hayou. Feature learning and signal propagation in deep neural networks. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 14248–14282. PMLR, 17–23 Jul 2022.

[55] Carl M Bender and Steven Orszag. *Advanced mathematical methods for scientists and engineers I: Asymptotic methods and perturbation theory*, volume 1. Springer Science & Business Media, 1999.

[56] Mehran Kardar. *Statistical physics of fields*. Cambridge University Press, 2007.

[57] Aitor Lewkowycz, Yasaman Bahri, Ethan Dyer, Jascha Sohl-Dickstein, and Guy Gur-Ari. The large learning rate phase of deep learning: the catapult mechanism. *arXiv preprint arXiv:2003.02218*, 2020.

[58] Ben Adlam and Jeffrey Pennington. The neural tangent kernel in high dimensions: Triple descent and a multi-scale theory of generalization. In *International Conference on Machine Learning*, pages 74–84. PMLR, 2020.

[59] Soufiane Hayou. On the infinite-depth limit of finite-width neural networks. *Transactions on Machine Learning Research*, 2023.

[60] Blake Bordelon, Lorenzo Noci, Mufan Bill Li, Boris Hanin, and Cengiz Pehlevan. Depthwise hyperparameter transfer in residual networks: Dynamics and scaling limit. *arXiv preprint arXiv:2309.16620*, 2023.

[61] Jonathan Heek, Anselm Levskaya, Avital Oliver, Marvin Ritter, Bertrand Rondepierre, Andreas Steiner, and Marc van Zee. Flax: A neural network library and ecosystem for JAX, 2023.

[62] Michael Baake and Ulrike Schlaegel. The peano-baker series. *Proceedings of the Steklov Institute of Mathematics*, 275(1):155–159, 2011.

[63] Roger W Brockett. *Finite dimensional linear systems*. SIAM, 2015.

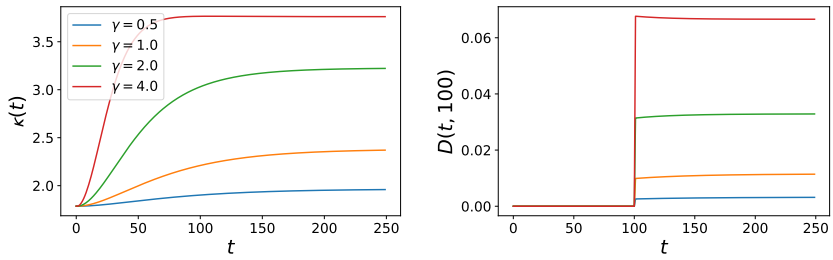
[64] Alexander Atanasov, Blake Bordelon, and Cengiz Pehlevan. Neural networks as kernel learners: The silent alignment effect. In *International Conference on Learning Representations*, 2022.

[65] B. Bordelon, A. Canatar, and C. Pehlevan. Spectrum dependent learning curves in kernel regression and wide neural networks. *International Conference of Machine Learning*, 2020.

[66] Phan-Minh Nguyen. Mean field limit of the learning dynamics of multilayer neural networks. *arXiv preprint arXiv:1902.02880*, 2019.

Appendix

A Additional Figures



(a) Fourth Moment $(t; t)$ (b) Kernel sensitivity $D(t; 100)$

Figure A.1: The κ and D functions for varying γ in Figure 3. (a) The uncoupled kernel variance $\kappa(t, t)$ increases monotonically with γ. This reveals that the dynamical filtering of κ is what is responsible for the late time decrease in variance during feature learning. (b) The tensor $D(t, s)$ measures sensitivity of kernel at time t to perturbation in Δ at time s. The $D(t, s)$ entries also increase with γ. This suggests that the reduction in variance of the training error and the kernel are not due to reduction in κ, but rather a dynamical filtering effect due to scale growth in K_1 and rapid reduction in Δ_1.

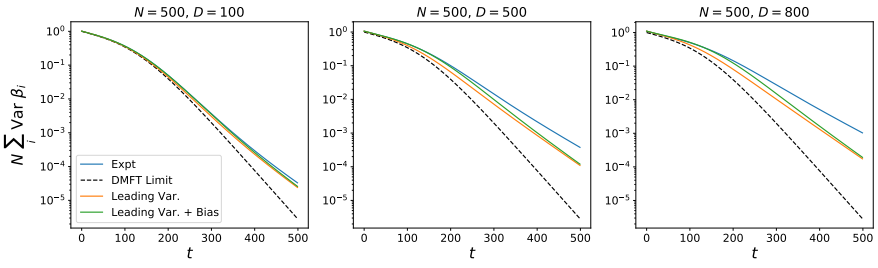


Figure A.2: A comparison of the bias and variance corrections in the toy model of Figure 4. At small D/N (or P/N for offline training) the leading variance and the leading variance and leading bias both track the experiment. Both the bias and the variance contribute positively towards the total generalization error since the variance correction alone (orange) exceeds the DMFT limiting error (dashed) and the variance and bias correction together (green) exceed variance alone (orange). However, for large D/N (or P/N) the leading order picture fails to describe the finite width experiment, indicating significant variance possibly at higher order scales (like $D^2/N^2, D^3/N^3, ...$).

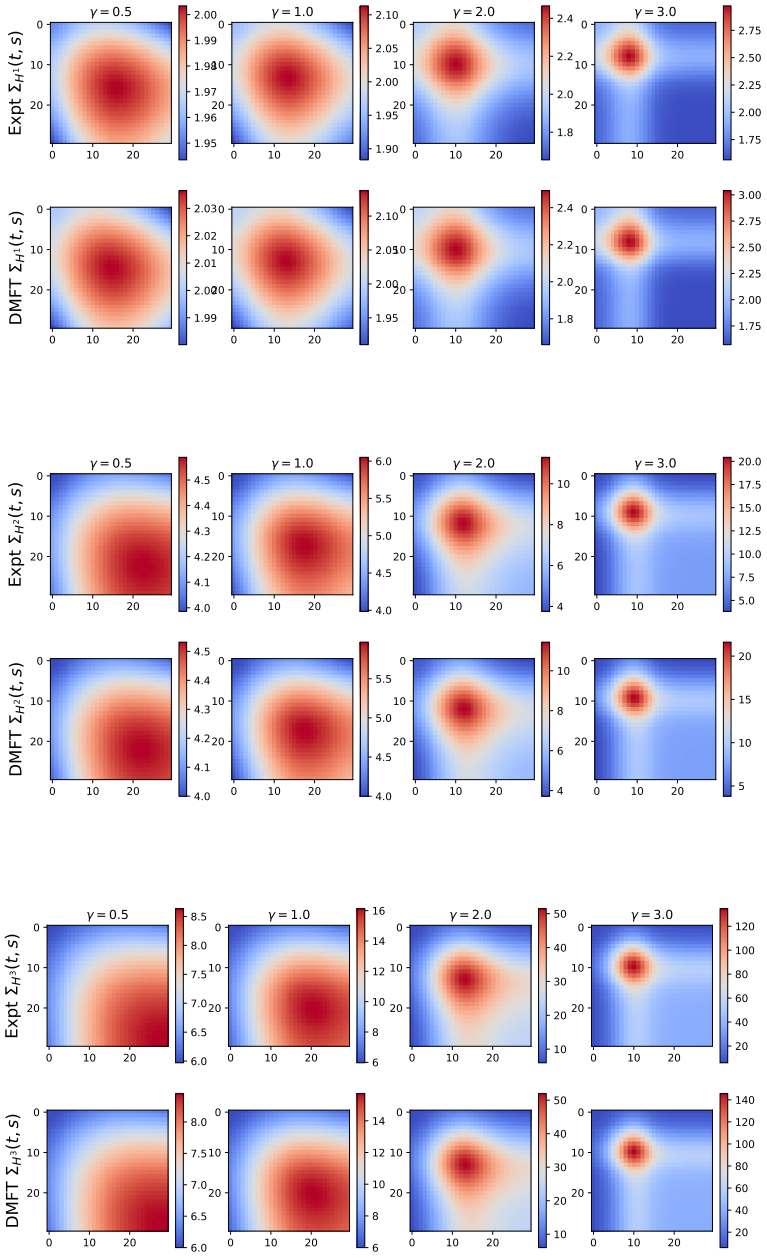
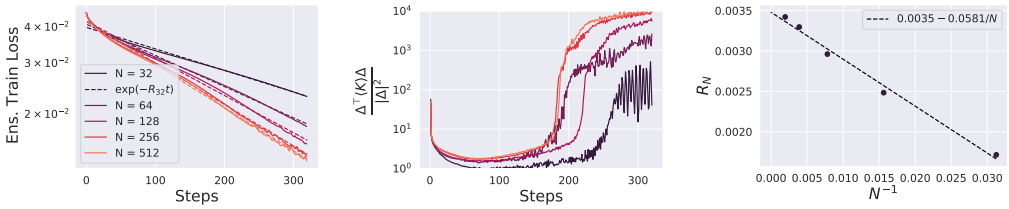
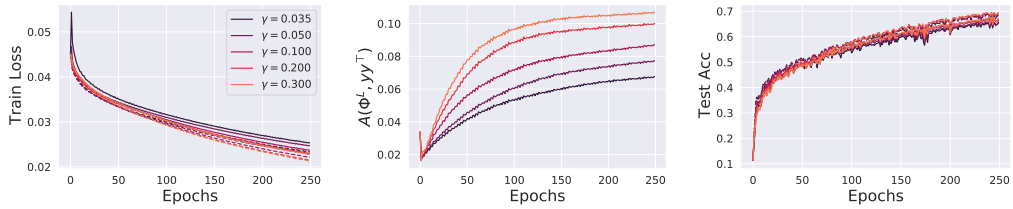


Figure A.3: The covariance of kernel entries across pairs of time points $\Sigma_{H^\ell}(t,s) = N\ \mathrm{Cov}(H^\ell(t,t), H^\ell(s,s))$ for depth 4 linear network trained on whitened data. The variance becomes increasingly localized in time as feature learning γ increases.

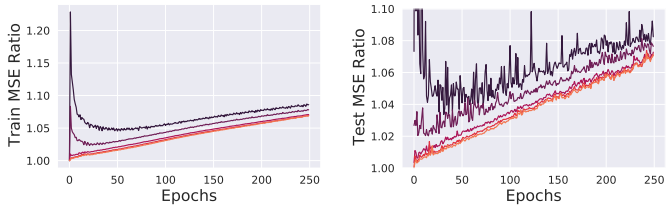


(a) Ens. Averaged Train Loss (b) Raleigh Quotient $\frac{\Delta^\top K \Delta}{|\Delta|^2}$ (c) Rates as Function of N

Figure A.4: The ensemble averaged train loss for the same depth 6 CNN trained on a random subsample of $P = 64$ CIFAR-10 points. Training is full batch gradient descent with $\gamma = 0.05$. (a) The ensemble train accuracy for this subset of CIFAR-10 is well modeled as an exponential in time $\mathcal{L}(t) \propto \exp(-R_N t)$ with a rate R_N that depends on width. (b) The projection of the errors Δ on the average NTK $\langle K \rangle$ (which is related to the rate of decay of the training loss, see Appendix G) reveals that wider networks are more aligned with their instantaneous error signals. (c) The rates R_N are indeed a linear functions of N^{-1}, with $R_N = R_1 + \frac{R^1}{N}$, consistent with the average NTK $\langle K \rangle$ receiving a N^{-1} correction. Using ensembling to find a scaling law like that above can thus allow one extrapolate the training rate of infinite width mean field models.



(a) Training MSE (b) Kernel-Task Alignment (c) Test Acc.



(d) Single vs Ensembled Ratio (e) Single vs Ensembled Ratio

Figure A.5: Width $N = 64$ depth 6 CNNs trained on the full CIFAR-10 with MSE. An ensemble of size $E = 10$ randomly initialized networks are trained. (a) Training MSE for varying γ. (b) Final layer kernel-task alignment does strongly depend on γ, despite similar train dynamics. (c) Top-1 classification test accuracy is only slightly different across γ. A small benefit from ensembling is visible late in training. (c) Initialization variance (measured by the ratio of single model to ensembled MSE) for training and test losses. Richer networks have lower variance throughout training. (b) Networks have distinct kernel dynamics when trained with different γ as evidenced by the alignment (cosine similarity) between the final layer feature kernel Φ^L and the target test labels y.

B CIFAR-10 Experimental Details

We trained the following depth 6 CNN architecture in the mean field parameterization using FLAX [61] on a single GPU. The bias parameters were zero in each hidden Conv layer, but were used for the readout weights. The networks were trained with MSE loss on centered 10 dimensional targets $y \in \mathsf{R}^{10}$ for $\mu \in [P]$. Each convolution was followed by an average pooling operation. To obtain mean field behavior, NTK parameterization with a modified final layer is used [7, 9].

```
from flax import linen as nn
import jax.numpy as jnp

class CNN(nn.Module):

  width: int

  def setup(self):
    kif = nn.initializers.normal(stddev = 1.0) # O_N(1) entries
    self.conv1 = nn.Conv(features = self.width, kernel_init = kif,
    use_bias = False, kernel_size = (3,3))
    self.conv2 = nn.Conv(features = self.width, kernel_init = kif,
    use_bias = False, kernel_size = (3,3))
    self.conv3 = nn.Conv(features = self.width, kernel_init = kif,
    use_bias = False, kernel_size = (3,3))
    self.conv4 = nn.Conv(features = self.width, kernel_init  = kif,
    use_bias = False, kernel_size = (3,3))
    self.conv5 = nn.Conv(features = self.width, kernel_init  = kif,
    use_bias = False, kernel_size = (3,3))
    self.readout = nn.Dense(features = 10, use_bias = True,
    kernel_init = kif)
    return

  def __call__(self, x, train = True):
    N = self.width
    D = 3
    x = self.conv1(x) / jnp.sqrt(D * 9)
    x = jnp.sqrt(2.0) * nn.relu(x)
    x = nn.avg_pool(x, window_shape=(2,2), strides = (2,2)) # 32 x 32
    -> 16 x 16
    x = self.conv2(x) / jnp.sqrt(N*9) # explicit N^{-1/2}
    x = jnp.sqrt(2.0) * nn.relu(x)
    x = nn.avg_pool(x, window_shape=(2,2), strides = (2,2)) # 16 x 16
    -> 8 x 8
    x = self.conv3(x)/jnp.sqrt(N*9)
    x = jnp.sqrt(2.0) * nn.relu(x)
    x = nn.avg_pool(x, window_shape=(2,2), strides = (2,2)) # 8 x 8 ->
     4 x 4
    x = self.conv4(x) / jnp.sqrt(N*9)
    x = jnp.sqrt(2.0) * nn.relu(x)
    x = nn.avg_pool(x, window_shape=(2,2), strides = (2,2)) #  4 x 4
    -> 2 x 2
    x = self.conv5(x) / jnp.sqrt(N*9)
    x = jnp.sqrt(2.0) * nn.relu(x)
    x = nn.avg_pool(x, window_shape=(2,2), strides = (2,2)) #  2 x 2
    -> 1 x 1
    x = x.reshape((x.shape[0], -1))  # flatten
    x = self.readout(x) / N # for mean field scaling
    return x
```

All models were trained with standard SGD with a batch size of 256. Each element in the ensemble of E networks is trained on identical batches presented in identical order. For the Figure 7 experiments, the raw learning rate is scaled as $\eta = 10N\sqrt{\gamma}$ with $\gamma = 0.2$ (note that mean field theory requires scaling the raw learning rate linearly with N since the raw NTK is $\mathcal{O}(N^{\ 1})$ [9]). For Figure A.5, the learning rate is $\eta = 5N\sqrt{\gamma}$. We find that choosing $\eta \propto \sqrt{\gamma}$ gives approximately conserved training

times across γ (though distinct representation dynamics). The Figure A.4 shows the dynamics of fitting $P = 64$ training points with full batch gradient descent and $\gamma = 0.1$.

C Review of DMFT: Deriving the Action

In this section we derive the DMFT action which contains all of the necessary statistical information about randomly initialized finite width N networks. From the action S the DMFT saddle point and the propagator can be computed. This derivation follows closely the original derivation by Bordelon & Pehlevan [9]. We start by writing the gradient flow dynamics on weight matrices

$$\frac{d}{dt}\boldsymbol{W}^{\ell}(t) = \frac{\gamma}{\sqrt{N}} \sum_{\alpha=1}^{N} \Delta^{\alpha}(t)\boldsymbol{g}^{\ell+1}(t)\phi(\boldsymbol{h}^{\ell}(t))^{\top} \tag{11}$$

where $\Delta^{\alpha}(t) = -\frac{\partial L}{\partial f^{\alpha}(t)}$ are the error signals and $\boldsymbol{g}^{\ell}(t) = N\gamma \frac{\partial f^{\alpha}(t)}{\partial h^{\ell}_{\alpha}(t)}$ are the back-propagation signals. The prediction dynamics satisfy

$$\frac{d}{dt} f^{\alpha}(t) = \sum_{\mu} K^{\alpha\mu}(t)\Delta^{\mu}(t) \tag{12}$$

where $K^{\alpha\mu}(t)$ is the instantaneous neural tangent kernel (NTK). At finite width N all of the above quantities depend on the precise initialization of the network. We transform the weight dynamics into an integral equation and use the recurrence for $\boldsymbol{h}^{\ell}$ to obtain the following

$$\begin{aligned}
\boldsymbol{h}^{\ell+1}(t) &= \frac{1}{\sqrt{N}}\boldsymbol{W}^{\ell}(0)\phi(\boldsymbol{h}^{\ell}(t)) + \gamma \int_0^t ds \sum_{\alpha} \Phi^{\ell}(t,s)\boldsymbol{g}^{\ell+1}(s) \\
\boldsymbol{g}^{\ell}(t) &= \dot{\phi}(\boldsymbol{h}^{\ell}(t)) \odot \boldsymbol{z}^{\ell}(t) \\
\boldsymbol{z}^{\ell}(t) &= \frac{1}{\sqrt{N}}\boldsymbol{W}^{\ell}(0)^{\top}\boldsymbol{g}^{\ell+1}(t) + \gamma \int_0^t ds \sum_{\alpha} G^{\ell+1}(t,s)\phi(\boldsymbol{h}^{\ell}(s))
\end{aligned} \tag{13}$$

where we introduced the feature and gradient kernels

$$\Phi^{\ell}(t,s) = \frac{1}{N}\phi(\boldsymbol{h}^{\ell}(t)) \cdot \phi(\boldsymbol{h}^{\ell}(s)) \ , \ G^{\ell}(t,s) = \frac{1}{N}\boldsymbol{g}^{\ell}(t) \cdot \boldsymbol{g}^{\ell}(s). \tag{14}$$

Written this way, we see that the source of the disorder which depends on the initial random weights $\boldsymbol{W}^{\ell}(0)$ comes through the fields

$$\chi^{\ell+1}(t) = \frac{1}{\sqrt{N}}\boldsymbol{W}^{\ell}(0)\phi(\boldsymbol{h}^{\ell}(t)) \ , \quad \xi^{\ell+1}(t) = \frac{1}{\sqrt{N}}\boldsymbol{W}^{\ell}(0)^{\top}\boldsymbol{g}^{\ell}(t). \tag{15}$$

If we can characterize the distribution of the fields $\chi^{\ell}(t)$ and $\xi^{\ell}(t)$, then we can consequently characterize the distribution of $\boldsymbol{h}^{\ell}(t), \boldsymbol{g}^{\ell}(t)$. We therefore choose to study the moment generating functional

$$Z[\{\boldsymbol{j}^{\ell}, \boldsymbol{v}^{\ell}\}] = \left\langle \exp\left(\sum_{\ell} \int dt \ \boldsymbol{j}^{\ell}(t) \cdot \chi^{\ell}(t) + \boldsymbol{v}^{\ell}(t) \cdot \xi^{\ell}(t) \right) \right\rangle_0 \tag{16}$$

Moments of these fields can be computed through differentiation with respect to the sources $\boldsymbol{j}, \boldsymbol{v}$ near zero-source ($\boldsymbol{j} = \boldsymbol{v} = 0$)

$$\begin{aligned}
&\chi^{\ell_1}_{\alpha_1}(t_1)...\chi^{\ell_n}_{\alpha_n}(t_n)\xi^{\ell_1}_{\alpha_1}(t_1)...\xi^{\ell_m}_{\alpha_m}(t_m) \\
&= \frac{\delta}{\delta j^{\ell_1}_{\alpha_1}(t_1)}...\frac{\delta}{\delta j^{\ell_n}_{\alpha_n}(t_n)}\frac{\delta}{\delta v^{\ell_1}_{\alpha_1}(t_1)}...\frac{\delta}{\delta v^{\ell_m}_{\alpha_m}(t_m)} Z[\{\boldsymbol{j}^{\ell}, \boldsymbol{v}^{\ell}\}]|_{j=v=0}.
\end{aligned} \tag{17}$$

To average over the initial weights, we introduce a Fourier representation of the Dirac-Delta function $1 = \int dz\delta(z) = \int \frac{dzd\hat{z}}{2}\exp(i\hat{z}z)$. We perform this transformation for each of the fields to enforce

their definition

$$\begin{aligned}\delta\left(h^{\ell}(t)-\frac{1}{\sqrt{N}}W^{\ell}(0)\phi(h^{\ell-1}(t))\right) &= \int\frac{d\hat{h}^{\ell}(t)}{(2\pi)^N}\exp\left(\hat{h}^{\ell}(t)\cdot\left(h^{\ell}(t)-\frac{1}{\sqrt{N}}W^{\ell}(0)\phi(h^{\ell-1}(t))\right)\right)\\ \delta\left(h^{\ell}(t)-\frac{1}{\sqrt{N}}W^{\ell}(0)^{\top}g^{\ell}(t)\right) &= \int\frac{d\hat{h}^{\ell}(t)}{(2\pi)^N}\exp\left(\hat{h}^{\ell}(t)\cdot\left(h^{\ell}(t)-\frac{1}{\sqrt{N}}W^{\ell}(0)^{\top}g^{\ell+1}(t)\right)\right)\end{aligned} \tag{18}$$

We insert these Dirac delta functions so that we can directly average over the weights

$$\begin{aligned}&\ln\mathbb{E}_{W^{\ell}(0)}\exp\left(-\frac{i}{\sqrt{N}}\mathrm{Tr}W^{\ell}(0)^{\top}\int dt\left[\hat{h}^{\ell+1}(t)\phi(h^{\ell}(t))^{\top}+g^{\ell+1}(t)\hat{h}^{\ell}(t)^{\top}\right]\right)\\ &=-\frac{1}{2}\sum_{\ell}\int dtds\left[\hat{h}^{\ell+1}(t)\cdot\hat{h}^{\ell+1}(s)\Phi^{\ell}(t,s)+\hat{h}^{\ell}(t)\cdot\hat{h}^{\ell}(s)G^{\ell+1}(t,s)\right]\\ &-\frac{1}{N}\sum_{\ell}\int dtds(\hat{h}^{\ell+1}(t)\cdot g^{\ell+1}(s))(\phi(h^{\ell}(t))\cdot\hat{h}^{\ell}(s))\end{aligned} \tag{19}$$

where we introduced the kernels Φ^{ℓ}, G^{ℓ}. We next introduce the order parameter

$$A^{\ell}(t,s)=-\frac{i}{N}\phi(h^{\ell}(t))\cdot\hat{h}^{\ell}(s) \tag{20}$$

To enforce the definitions of the new order parameters $\{\Phi, G, A\}$ we again introduce Dirac-delta functions

$$\begin{aligned}&\delta\left(N\Phi^{\ell}(t,s)-\phi(h^{\ell}(t))\cdot\phi(h^{\ell}(s))\right)\\ &=\int\frac{d\hat{\Phi}^{\ell}(t,s)}{2\pi i}\exp\left(\hat{\Phi}^{\ell}(t,s)\left(N\Phi^{\ell}(t,s)-\phi(h^{\ell}(t))\cdot\phi(h^{\ell}(s))\right)\right)\end{aligned} \tag{21}$$

Analogous constraints for G and A are enforced with conjugate variables $\hat{G}, B$. After introducing these variables, we find that the moment generating functional has the form

$$\begin{aligned}Z=\int\prod_{\ell ts}&\frac{d\hat{\Phi}^{\ell}(t,s)d\Phi^{\ell}(t,s)}{2\pi i}\frac{d\hat{G}^{\ell}(t,s)dG^{\ell}(t,s)}{2\pi i}\frac{d\hat{G}^{\ell}(t,s)dG^{\ell}(t,s)}{2\pi i}\frac{dA^{\ell}(t,s)dB^{\ell}(t,s)}{2\pi i}\\ &\exp\left(NS[\{\Phi^{\ell},\hat{\Phi}^{\ell},G^{\ell},\hat{G}^{\ell},A^{\ell},B^{\ell}\}]\right)\end{aligned} \tag{22}$$

where S is the $\mathcal{O}(1)$ DMFT *action* which defines the statistical distribution over the dynamics. The action takes the form

$$\begin{aligned}S=&\sum_{\ell}\int dt\,ds\left[\hat{\Phi}^{\ell}(t,s)\Phi^{\ell}(t,s)+\hat{G}^{\ell}(t,s)-A^{\ell}(t,s)B^{\ell}(s,t)\right]\\ &+\frac{1}{N}\sum_{\ell=1}^{\infty}\sum_{i=1}^{\infty}\ln\mathcal{Z}^{\ell}[\{j_i^{\ell},v_i^{\ell}\}]\end{aligned} \tag{23}$$

where $\mathcal{Z}^{\ell}$ is the single site stochastic process for layer ℓ which defines the marginal distribution of χ, ξ, with the following form

$$\begin{aligned}\mathcal{Z}^{\ell}[\{j^{\ell}(t),v^{\ell}(t)\}]=\int\prod_{t}&\frac{d\hat{\chi}^{\ell}(t)d\chi^{\ell}(t)}{2\pi}\frac{d\hat{\xi}^{\ell}(t)d\xi^{\ell}(t)}{2\pi}\exp\left(\int dt\sum_{\ell}[j^{\ell}(t)\chi^{\ell}(t)+v^{\ell}(t)\xi^{\ell}(t)]\right)\\ &\exp\left(-\frac{1}{2}\sum_{\ell}\int dtds\left[\hat{\Phi}^{\ell}(t,s)\hat{\chi}^{\ell}(t)\hat{\chi}^{\ell}(s)+\hat{G}^{\ell}(t,s)\hat{\xi}^{\ell}(t)\hat{\xi}^{\ell}(s)\right]\right)\\ &\exp\left(-i\sum_{\ell}\int dtds\left[B^{\ell}(t,s)\hat{\xi}^{\ell}(t)\phi(h^{\ell}(s))+A^{\ell-1}(t,s)\hat{\chi}^{\ell}(t)g^{\ell}(s)\right]\right)\\ &\exp\left(i\sum_{\ell}\int dt[\hat{\chi}^{\ell}(t)\chi^{\ell}(t)+\hat{\xi}^{\ell}(t)\xi^{\ell}(t)]\right)\end{aligned} \tag{24}$$

where in the above, the $\{h, g\}$ fields should be regarded as functionals of $\{\chi, \xi\}$. At zero source $j, v \to 0$ this function S can be regarded as the log density for the complete collection of order parameters $q = \{\hat{\Phi}, \Phi, \hat{G}, G, A, B\}$ which collectively control the dynamics. Concretely, we have that $p(q) \propto \exp(NS(q))$. In the next section we explore an approximation scheme for averages over this distribution at large N.

D Cumulant Expansion of Observables

We are interested in a principled power series expansion (in $1/N$) of any observable average $\langle O(q) \rangle$ that depends on DMFT order parameters q. At any width N the observable average takes the form

$$\langle O(q) \rangle_N = \frac{\int dq \exp(NS(q)) O(q)}{\int dq \exp(NS(q))} \tag{25}$$

As discussed in the main text, the $N \to \infty$ limit gives $\langle O(q) \rangle_N \sim O(q_1)$ where $\frac{\partial S}{\partial q}|_{q_1} = 0$ by a steepest descent argument [55]. We assume that S's Hessian is negative semidefinite so that $\equiv -\nabla^2 S(q)|_{q_1}{}^{-1} \succeq 0$ and Taylor expand $S(q)$ around the saddle point q_1 giving $S(q) = S(q_1) + \frac{1}{2}(q - q_1)^\top \nabla^2 S(q)(q - q_1) + V(q - q_1)$. We note that the remainder function V contains only cubic and higher powers of $q - q_1 \equiv /\sqrt{N}$. The variable will be order $\mathcal{O}(1)$. This will allow us to verify that additional terms are suppressed in powers of $1/N$. Expanding both the numerator and denominator's integrands in powers of V, we find

$$\begin{aligned}
\langle O(q) \rangle_N &= \frac{\int dq \exp\left(-\frac{N}{2}(q - q_1)^\top {}^{-1}(q - q_1) + NV(q - q_1)\right) O(q)}{\int dq \exp\left(-\frac{N}{2}(q - q_1)^\top {}^{-1}(q - q_1) + NV(q - q_1)\right)} \\
&= \frac{\int d \exp\left(-\frac{1}{2}{}^\top {}^{-1}\right)(1 + NV + \frac{N^2}{2}V^2 + ...) O(q_1 + N^{-1=2})}{\int d \exp\left(-\frac{1}{2}{}^\top {}^{-1}\right)(1 + NV + \frac{N^2}{2}V^2 + ...)} \\
&= \frac{\langle O \rangle_1 + N \langle VO \rangle_1 + \frac{N^2}{2!} V^2 O_1 + \frac{N^3}{3!} V^3 O_1 + ...}{1 + N \langle V \rangle_1 + \frac{N^2}{2!} \langle V^2 \rangle_1 + \frac{N^3}{3!} \langle V^3 \rangle_1 + ...} \\
&= \langle O \rangle_1 \frac{1 + N \langle VO \rangle_1 / \langle O \rangle_1 + \frac{N^2}{2!} V^2 O_1 / \langle O \rangle_1 + \frac{N^3}{3!} V^3 O_1 / \langle O \rangle_1 + ...}{1 + N \langle V \rangle_1 + \frac{N^2}{2!} \langle V^2 \rangle_1 + \frac{N^3}{3!} \langle V^3 \rangle_1 + ...}
\end{aligned} \tag{26}$$

where $\langle \rangle_1$ represents an average over the Gaussian fluctuation $\mathcal{N}\left(q_1, -\frac{1}{N}\nabla_q^2 S(q_1)^{-1}\right)$. We see that the series in the denominator contains terms of the form $\frac{N^k}{k!} V^k{}_1$ while the numerator depends on terms of the form $\frac{N^k}{k!} V^k O_1 / \langle O \rangle_1$. In either of these power series, the k-th term can contribute at most

$$\frac{N^k V^k O_1}{\langle O \rangle_1}, N^k V^k{}_1 \sim \begin{cases} \mathcal{O}(N^{(k+1)=2}) & k \text{ odd} \\ \mathcal{O}(N^{k=2}) & k \text{ even} \end{cases} \tag{27}$$

since V contributes only cubic and higher terms. Thus each term in the numerator and denominator's series contains increasing powers of $1/N$. Concretely, each of the two series have terms of order $\{N^0, N^{-1}, N^{-1}, N^{-2}, N^{-2}, ...\}$. Thus any quantity of the form $\frac{\langle O i}{\langle O i_1}$ admits a ratio of power series in powers of $1/N$. One could truncate each of the series in the numerator and denominator to a desired order in N. Alternatively, the denominator could be expanded giving a single series (the cumulant expansion [56]). The first few terms in the cumulant expansion have the form

$$\begin{aligned}
\langle O \rangle_N &= \langle O \rangle_1 + N \left[\langle OV \rangle_1 - \langle O \rangle_1 \langle V \rangle_1 \right] \\
&+ \frac{N^2}{2} \Big[V^2 O_1 - 2 \langle VO \rangle_1 \langle V \rangle_1 + 2 \langle V \rangle_1^2 \langle O \rangle_1 - V^2{}_1 \langle O \rangle_1 \Big] + ...
\end{aligned} \tag{28}$$

In this work, we mainly are interested in the leading order correction to $\langle O \rangle$ which can always be obtained with the truncation after the terms linear in V for any observable O.

D.1 Square Deviation from DMFT

We will now analyze the fluctuation statistics of our order parameters around the saddle point $(q - q_1)(q - q_1)^\top_N$ which has the form

$$(q - q_1)(q - q_1)^\top_N = \frac{(q - q_1)(q - q_1)^\top_1 + N\ V\,(q - q_1)(q - q_1)^\top_1 + \ldots}{1 + N\langle V\rangle_1 + \ldots}$$
$$= \frac{\frac{1}{N} + \mathcal{O}(N^{-2})}{1 + \mathcal{O}(N^{-1})} \sim \frac{1}{N} + \mathcal{O}(N^{-2}), \tag{29}$$

as stated in the main text and verified empirically in Figure 3 (a). The reason that the terms in the numerator involving V can be no larger than $\mathcal{O}(N^{-2})$ comes from vanishing of odd moments for $q - q_1$ in the unperturbed distribution. Thus the leading expression for $(q - q_1)(q - q_1)^\top$ only depends on and not on V.

D.2 Mean Deviation from DMFT

Although the square displacement from DMFT only depended on Σ and not on V, we note that the *average order parameter displacement* $\langle q - q_1\rangle_N$ does receive a $\mathcal{O}(1/N)$ correction that depends on the perturbed potential V

$$\langle q - q_1\rangle_N = \frac{\langle q - q_1\rangle_1 + N\langle (q - q_1)V\rangle_1 + \frac{N^2}{2}\ (q - q_1)V^2\ _1 + \ldots}{1 + N\langle V\rangle_1 + \frac{N^2}{2}\langle V^2\rangle_1 + \ldots}$$
$$\sim \frac{\left\langle \frac{\partial V}{\partial q}\right\rangle_1 + \mathcal{O}(N^{-2})}{1 + \mathcal{O}(N^{-1})} \sim \left\langle \frac{\partial V}{\partial q}\right\rangle_1 + \mathcal{O}(N^{-2}). \tag{30}$$

where in the last line we used Stein's lemma (Gaussian integration by parts) for the Gaussian distribution over q. Note that $\left\langle \frac{\partial V}{\partial q}\right\rangle_1 \sim \mathcal{O}\left(\frac{1}{N}\right)$ since the derivative of the cubic term in V gives a quadratic function of $q - q_1$, whose average must be $\mathcal{O}(N^{-1})$. In this work, we focus primarily on the structure of the propagator, but outline a general recipe for getting the leading mean correction in Appendix G and H.2.

D.3 Covariance of Order Parameters

Lastly, we combine the previous two observations to reason about the scaling of the order parameter covariance over initializations. We note that the leading covariance of the order parameters over random initializations is also given by the propagator: $\mathrm{Cov}(q) \sim \frac{1}{N} + \mathcal{O}(N^{-2})$, since

$$\mathrm{Cov}(q) = \left\langle (q - \langle q\rangle_N)(q - \langle q\rangle_N)^\top \right\rangle_N$$
$$= \left\langle (q - q_1)(q - q_1)^\top \right\rangle_N - \left\langle (q_1 - \langle q\rangle_N)(q_1 - \langle q\rangle_N)^\top \right\rangle_N$$
$$\sim \frac{1}{N} + \mathcal{O}(N^{-2}) \tag{31}$$

due to the arguments above which showed that $(q - q_1)(q - q_1)^\top \sim \frac{1}{N} + \mathcal{O}(N^{-2})$ and that $q_1 - \langle q\rangle_N \sim \mathcal{O}(N^{-1})$. Therefore, in the leading order picture, it is safe to associate with the covariance of order parameters over random initializations of the network weights.

E Propagator Structure for the full DMFT Action

In this section, we examine the propagator structure for the full DMFT action. This action is modified from other prior works [9, 46] to include the evolution of the network prediction errors $\Delta(t)$. Those prior works noted that Δ and the NTK K are deterministic functions of deterministic order parameters $\{\Phi, G\}$ in the $N \to \infty$ limit so those authors did not explicitly include Δ or K in the action. At finite width N, including Δ, K in the action is crucial as the fluctuation in prediction errors Δ has significant consequences for dynamical fluctuations of kernels through the preactivation and

pre-gradient fields. In this section, we will mainly focus on gradient flow, but we describe large step size in Appendix M.

$$
\begin{aligned}
S = & \sum_{\ell} \int dt ds \left[\hat{\Phi}^{\ell}(t,s)\Phi^{\ell}(t,s) + \hat{G}^{\ell}(t,s)G^{\ell}(t,s) - \gamma^2 A^{\ell}(s,t)B^{\ell}(t,s) \right] \\
& + \sum_{\ell} \int dt \hat{\Delta}^{\ell}(t) \left[\Delta^{\ell}(t) - y^{\ell} + \sum_{\ell} \int ds \Theta(t-s) K^{\ell}(s) \Delta^{\ell}(s) \right] \\
& + \sum_{\ell} \int dt \hat{K}^{\ell}(t) \left[K^{\ell}(t) - \sum_{\ell} G^{\ell+1}(t) \Phi^{\ell}(t) \right] \\
& + \sum_{\ell} \ln \mathcal{Z}^{\ell}[\ , \hat{\ }^{\ell}, \hat{\boldsymbol{G}}^{\ell}, \ ^{\ell-1}, \boldsymbol{G}^{\ell+1}, \boldsymbol{A}^{\ell-1}, \boldsymbol{B}^{\ell}]
\end{aligned} \tag{32}
$$

where the single site moment generating functionals $\mathcal{Z}^{\ell}$ have the form

$$
\begin{aligned}
\mathcal{Z}^{\ell} = & \mathbb{E}_{\{h^{\ell}(t); z^{\ell}(t)\}} \exp\left(- \sum_{\ell} \int dt ds \left[\phi(h^{\ell}(t))\phi(h^{\ell}(s))\hat{\Phi}^{\ell}(t,s) + g^{\ell}(t)g^{\ell}(s)\hat{G}^{\ell}(t,s) \right] \right) \\
h^{\ell}(t) = & u^{\ell}(t) + \gamma \int_0^t ds \sum_{\ell} \Phi^{\ell-1}(t,s)\Delta^{\ell}(s) + A^{\ell-1}(t,s)\, g^{\ell}(s) \ , \ \{u^{\ell}(t)\} \sim \mathcal{GP}(0, \ ^{\ell-1}) \\
z^{\ell}(t) = & r^{\ell}(t) + \gamma \int_0^t ds \sum_{\ell} G^{\ell+1}(t,s)\Delta^{\ell}(s) + B^{\ell}(t,s)\, \phi(h^{\ell}(s)) \ , \ \{r^{\ell}(t)\} \sim \mathcal{GP}(0, \boldsymbol{G}^{\ell+1})
\end{aligned} \tag{33}
$$

with $g^{\ell}(t) = \dot{\phi}(h^{\ell}(t))z^{\ell}(t)$. The saddle point equations give the infinite width evolution of our order parameters.

$$
\begin{aligned}
\frac{\partial S}{\partial \hat{\Phi}^{\ell}(t,s)} &= \Phi^{\ell}(t,s) - \phi(h^{\ell}(t))\phi(h^{\ell}(s)) = 0 \\
\frac{\partial S}{\partial \hat{G}^{\ell}(t,s)} &= G^{\ell}(t,s) - g^{\ell}(t)g^{\ell}(s) = 0 \\
\frac{\partial S}{\partial A^{\ell}(s,t)} &= -\gamma^2 B^{\ell}(t,s) + \gamma \left\langle \frac{\partial \phi(h^{\ell}(t))}{\partial r^{\ell}(s)} \right\rangle^{+} = 0 \\
\frac{\partial S}{\partial B^{\ell}(s,t)} &= -\gamma^2 A^{\ell}(t,s) + \gamma \left\langle \frac{\partial g^{\ell}(t)}{\partial u^{\ell}(s)} \right\rangle^{+} = 0 \\
\frac{\partial S}{\partial \hat{K}^{\ell}(t)} &= K^{\ell}(t) - \sum_{\ell} G^{\ell+1}(t,t)\Phi^{\ell}(t,t) = 0 \\
\frac{\partial S}{\partial \hat{\Delta}^{\ell}(t)} &= \Delta^{\ell}(t) - y^{\ell} + \int_0^t ds \sum_{\ell} K^{\ell}(s)\Delta^{\ell}(s) = 0
\end{aligned} \tag{34}
$$

These equations exactly recover the mean field description obtained [9]. Note that $\langle\rangle$ for field averages is an average defined by $\mathcal{Z}^{\ell}$ and is distinct from the types averages $\langle\rangle$, $\langle\rangle_{\ell}$ we have been considering over the order parameters $\boldsymbol{q}$. The complementary set of equations for the primal variables, such as $\frac{\partial S}{\partial \ ^{\ell}(t;s)} = 0$, give that $\hat{K} = \hat{\Delta} = \hat{\Phi} = \hat{G} = 0$ at the saddle point. We now set out to compute the Hessan $\nabla_q^2 S$. To simplify the set of expressions, we will only explicitly write out the nonvanishing

blocks. We will start with second derivatives involving only pairs of dual variables $\{\hat{\Phi}, \hat{G}, A, B\}$

$$
\begin{aligned}
\frac{\partial^2 S}{\partial \hat{\Phi}(t,s)\partial \hat{\Phi}(t^0,s^0)} &= \phi(h(t))\phi(h(s))\phi(h(t^0))\phi(h(s^0)) - \Phi(t,s)\Phi(t^0,s^0) \\
&\equiv \kappa(t,s,t^0,s^0) \\
\frac{\partial^2 S}{\partial \hat{G}(t,s)\partial \hat{G}(t^0,s^0)} &= g(t)g(s)g(t^0)g(s^0) - G(t,s)G(t^0,s^0) \\
&\equiv \kappa^{G}(t,s,t^0,s^0) \\
\frac{\partial^2 S}{\partial \hat{\Phi}(t,s)\partial \hat{G}(t^0,s^0)} &= \phi(h(t))\phi(h(s))g(t^0)g(s^0) - \Phi(t,s)G(t^0,s^0) \\
&\equiv \kappa^{G}_{*}(t,s,t^0,s^0) \\
\frac{\partial^2 S}{\partial \hat{\Phi}(t,s)\partial A^{-1}(s^0,t^0)} &= -\gamma \left[\frac{\partial \phi(h(t))}{\partial u(s^0)} \phi(h(s)) g(t^0) \right]_{*}^{+} \\
&\quad - \gamma \left[\phi(h(t)) \frac{\partial \phi(h(t))}{\partial u(s^0)} g(t^0) \right]_{*}^{+} \\
&\quad - \gamma \left[\phi(h(t))\phi(h(s)) \frac{\partial g(t^0)}{\partial u(s^0)} \right]_{*}^{+} - \gamma^2 \Phi(t,s) B^{-1}(t^0,s^0) \\
&\equiv -\gamma \kappa^{B^{-1}}_{*}(t,s) \\
\frac{\partial^2 S}{\partial \hat{\Phi}(t,s)\partial B(s^0,t^0)} &= -\gamma \left[\frac{\partial \phi(h(t))}{\partial r(s^0)} \phi(h(s))\phi(h(t^0)) \right]_{*}^{+} \\
&\quad - \gamma \left[\phi(h(t)) \frac{\partial \phi(h(t))}{\partial r(s^0)} \phi(h(t^0)) \right]_{*}^{+} \\
&\quad - \gamma \left[\phi(h(t))\phi(h(s)) \frac{\partial \phi(h(t^0))}{\partial r(s^0)} \right]_{*}^{+} - \gamma^2 \Phi(t,s) A(t^0,s^0) \\
&\equiv -\gamma \kappa^{A}_{*}(t,s) \\
\frac{\partial^2 S}{\partial \hat{G}(t,s)\partial A^{-1}(s^0,t^0)} &= -\gamma \left[\frac{\partial g(t)}{\partial u(s^0)} g(s) g(t^0) \right]_{*}^{+} - \gamma \left[g(t) \frac{\partial g(t)}{\partial u(s^0)} g(t^0) \right]_{*}^{+} \\
&\quad - \gamma \left[g(t)g(s) \frac{\partial g(t^0)}{\partial u(s^0)} \right]_{*}^{+} - \gamma^2 G(t,s) B^{-1}(t^0,s^0) \\
&\equiv -\gamma \kappa^{G B^{-1}}_{*}(t,s) \\
\frac{\partial^2 S}{\partial \hat{G}(t,s)\partial B(s^0,t^0)} &= -\gamma \left[\frac{\partial g(t)}{\partial r(s^0)} g(s) \phi(h(t^0)) \right]_{*}^{+} - \gamma \left[g(t) \frac{\partial g(t)}{\partial r(s^0)} \phi(h(t^0)) \right]_{*}^{+} \\
&= -\gamma \left[g(t)g(s) \frac{\partial \phi(h(t^0))}{\partial r(s^0)} \right]_{*}^{+} - \gamma^2 G(t,s) A(t^0,s^0) \\
&\equiv -\gamma \kappa^{G A}(t,s) \\
\frac{\partial^2 S}{\partial A(t,s)\partial B(s^0,t^0)} &= -\gamma^2 \delta(t,t^0)\delta(s,s^0) \\
\frac{\partial^2 S}{\partial A^{-1}(s,t)\partial B(s^0,t^0)} &= \gamma^2 \left[\frac{\partial^2}{\partial u(s) \partial r(s^0)} g(t)\phi(h(t^0)) \right]_{*}^{+} - \gamma^4 B^{-1}(t,s) A(t^0,s^0) \\
&\equiv \kappa^{B^{-1}A}(t,s,t^0,s^0)
\end{aligned}
\tag{35}
$$

Next, we consider the second derivatives involving only primal variables $\{\Phi^{`}, G^{`}, K, \Delta\}$ which all vanish

$$\begin{aligned}
&\frac{\partial^2 S}{\partial \Phi^{`}\ (t,s)\partial \Phi^{`0}\ (t^0,s^0)} = 0 \\
&\frac{\partial^2 S}{\partial G^{`}\ (t,s)\partial G^{`0}\ (t^0,s^0)} = 0 \\
&\frac{\partial^2 S}{\partial \Phi^{`}\ (t,s)\partial G^{`0}\ (t^0,s^0)} = 0 \\
&\frac{\partial^2 S}{\partial \Phi^{`}\ (t,s)\partial K\quad (s^0)} = 0 \\
&\frac{\partial^2 S}{\partial G^{`}\ (t,s)\partial K\quad (s^0)} = 0 \\
&\frac{\partial^2 S}{\partial \Phi^{`}\ (t,s)\partial \Delta\ (s^0)} = 0 \\
&\frac{\partial^2 S}{\partial G^{`}\ (t,s)\partial \Delta\ (s^0)} = 0 \\
&\frac{\partial^2 S}{\partial K\quad (t)\partial K\quad (s)} = 0 \\
&\frac{\partial^2 S}{\partial K\quad (t)\partial \Delta\ (s)} = 0 \\
&\frac{\partial^2 S}{\partial \Delta\ (t)\partial \Delta\ (s)} = 0
\end{aligned} \tag{36}$$

Now we consider all derivatives which involve one of the dual variables $\{\hat{\Phi}^{`}, \hat{G}^{`}, A^{`}, B^{`}\}$ and the primal variable Δ

$$\begin{aligned}
&\frac{\partial^2 S}{\partial \hat{\Phi}^{`}\ (t,s)\partial \Delta\ (t^0)} = -\quad \frac{\partial}{\partial \Delta\ (t^0)}[\phi(h^{`}(t))\phi(h^{`}(s))] \quad \equiv -D^{\quad `}\quad (t,s,t^0) \\
&\frac{\partial^2 S}{\partial \hat{G}^{`}\ (t,s)\partial \Delta\ (t^0)} = -\quad \frac{\partial}{\partial \Delta\ (t^0)}[g^{`}(t)g^{`}(s)] \quad \equiv -D^{G^{`}}\quad (t,s,t^0) \\
&\frac{\partial^2 S}{\partial A^{`\ 1}(s,t)\partial \Delta\ (t^0)} = \gamma\quad \frac{\partial}{\partial \Delta\ (t^0)\partial u^{`}(s)} g^{`}(t) \quad \equiv \gamma D^{B^{`}\ ^{1};}\ (t,s,t^0) \\
&\frac{\partial^2 S}{\partial B^{`}\ (s,t)\partial \Delta\ (t^0)} = \gamma\quad \frac{\partial}{\partial \Delta\ (t^0)\partial r^{`}(s)} \phi(h^{`}(t)) \quad \equiv \gamma D^{A^{`}}\quad (t,s,t^0)
\end{aligned}$$

Now, we consider the second derivatives involving one derivative on a dual variable $\{\hat{\Phi}^{\ell}, \hat{G}^{\ell}, A, B\}$ and one of the primal variables $\{\Phi^{\ell}, G^{\ell}\}$.

$$
\begin{aligned}
\frac{\partial^2 S}{\partial \hat{\Phi}^{\ell}(t,s)\partial \Phi^{\ell'}(t',s')} &= \delta_{\ell,\ell'}\delta \quad \delta(t-t')\delta(s-s') \\
&\quad - \delta_{\ell \; 1,\ell'} \frac{\partial}{\partial \Phi^{\ell \; 1}(t',s')} \quad \phi(h^{\ell}(t))\phi(h^{\ell}(s)) \\
&\equiv \delta_{\ell,\ell'}\delta \quad \delta(t-t')\delta(s-s') - \delta_{\ell \; 1,\ell'} D^{\ell \; \ell \; 1}(t,s,t',s') \\
\frac{\partial^2 S}{\partial \hat{G}^{\ell}(t,s)\partial G^{\ell'}(t',s')} &= \delta_{\ell,\ell'}\delta \quad \delta(t-t')\delta(s-s') - \delta_{\ell+1,\ell'} \frac{\partial}{\partial G^{\ell+1}(t',s')} \quad g^{\ell}(t)g^{\ell}(s) \\
&\equiv \delta_{\ell,\ell'}\delta \quad \delta(t-t')\delta(s-s') - \delta_{\ell \; 1,\ell'} D^{G^{\ell} G^{\ell+1}}(t,s,t',s') \\
\frac{\partial^2 S}{\partial \hat{\Phi}^{\ell}(t,s)\partial G^{\ell+1}(t',s')} &= -\frac{\partial}{\partial G^{\ell+1}(t',s')} \quad \phi(h^{\ell}(t))\phi(h^{\ell}(s)) \quad \equiv -D^{\ell;G^{\ell+1}}(t,s,t',s') \\
\frac{\partial^2 S}{\partial \hat{G}^{\ell}(t,s)\partial \Phi^{\ell \; 1}(t',s')} &= -\frac{\partial}{\partial \Phi^{\ell \; 1}(t',s')} \quad g^{\ell}(t)g^{\ell}(s) \quad \equiv -D^{G^{\ell};\; \ell \; 1}(t,s,t',s') \\
\frac{\partial^2 S}{\partial A^{\ell \; 1}(s,t)\partial \Phi^{\ell \; 1}(t',s')} &= \gamma \frac{\partial}{\partial \Phi^{\ell \; 1}(t',s')} \left\langle \frac{\partial g^{\ell}(t)}{\partial r^{\ell}(s)} \right\rangle \equiv \gamma D^{B^{\ell \; 1};\; \ell \; 1}(t,s,t',s') \\
\frac{\partial^2 S}{\partial B^{\ell}(s,t)\partial \Phi^{\ell \; 1}(t',s')} &= \gamma \frac{\partial}{\partial \Phi^{\ell \; 1}(t',s')} \left\langle \frac{\partial \phi(h^{\ell}(t))}{\partial u^{\ell}(s)} \right\rangle \equiv \gamma D^{A^{\ell};\; \ell \; 1}(t,s,t',s') \\
\frac{\partial^2 S}{\partial A^{\ell \; 1}(s,t)\partial G^{\ell+1}(t',s')} &= \gamma \frac{\partial}{\partial G^{\ell+1}(t',s')} \left\langle \frac{\partial g^{\ell}(t)}{\partial r^{\ell}(s)} \right\rangle \equiv \gamma D^{B^{\ell \; 1};G^{\ell+1}}(t,s,t',s') \\
\frac{\partial^2 S}{\partial B^{\ell}(s,t)\partial G^{\ell+1}(t',s')} &= \gamma \frac{\partial}{\partial G^{\ell+1}(t',s')} \left\langle \frac{\partial \phi(h^{\ell}(t))}{\partial u^{\ell}(s)} \right\rangle \equiv \gamma D^{A^{\ell};G^{\ell+1}}(t,s,t',s')
\end{aligned} \tag{37}
$$

We note that terms such as $\frac{\partial}{\partial \Phi^{\ell \; 1}(t';s')} \quad \phi(h^{\ell}(t))\phi(h^{\ell}(s))$ can be further decomposed since the average over the $\{u^{\ell}(t)\} \sim \mathcal{GP}(0, \quad ^{\ell \; 1})$ and h^{ℓ}'s explicit dynamics both depend on $\Phi^{\ell \; 1}$

$$
\begin{aligned}
\frac{\partial}{\partial \Phi^{\ell \; 1}(t',s')} \quad \phi(h^{\ell}(t))\phi(h^{\ell}(s)) \quad &= \frac{1}{2} \left\langle \frac{\partial^2}{\partial u^{\ell}(t')\partial u^{\ell}(s')} \phi(h^{\ell}(t))\phi(h^{\ell}(s)) \right\rangle \\
&\quad + \left\langle \frac{\partial}{\partial \Phi^{\ell \; 1}(t',s')} \phi(h^{\ell}(t))\phi(h^{\ell}(s)) \right\rangle
\end{aligned} \tag{38}
$$

where the first term comes from differentiating the Gaussian probability density for u^{ℓ} (e.g. Price's theorem) and the second term is an explicit derivative of the preactivation fields with u^{ℓ} treated as constant. Next we consider the nonvanishing terms which involve $\{\hat{\Delta}, \hat{K}, \Delta, K\}$ which give

$$
\begin{aligned}
\frac{\partial^2 S}{\partial \hat{\Delta}(t)\partial \Delta(s)} &= \delta \quad \delta(t-s) + \Theta(t-s)K \quad (s) \\
\frac{\partial^2}{\partial \hat{\Delta}(t)\partial K \quad (s)} &= \delta \quad \Theta(t-s)\Delta(s) \\
\frac{\partial^2 S}{\partial \hat{K} \quad (t)\partial K \quad (t')} &= \delta \quad \delta \quad \delta(t-t') \\
\frac{\partial^2 S}{\partial \hat{K} \quad (t)\partial \Phi^{\ell}(t',s')} &= \delta \quad \delta \quad G^{\ell+1}(t',s')\delta(t-t')\delta(t-s') \\
\frac{\partial^2 S}{\partial \hat{K} \quad (t)\partial G^{\ell}(t',s')} &= \delta \quad \delta \quad \Phi^{\ell \; 1}(t',s')\delta(t-t')\delta(t-s')
\end{aligned} \tag{39}
$$

This enumerates all possible non-vanishing terms in the Hessian. We can now construct a block matrix of these Hessians by partitioning our order parameters $q = [q_1, q_2]^\top$ where

$$q_1 = \text{Vec}\{\Phi^\ell(t,s), G^\ell(t,s), K(t), \Delta^\ell(t), \hat{\Phi}^\ell(t,s), \hat{G}^\ell(t,s), \hat{K}(t), \hat{\Delta}^\ell(t)\} \tag{40}$$

$$q_2 = \text{Vec}\{A^\ell(t,s), B^\ell(t,s)\}. \tag{41}$$

This choice will become apparent shortly.

$$\nabla^2_q S = \begin{matrix} \nabla^2_{q_1} S & \nabla^2_{q_1 q_2} S \\ \nabla^2_{q_2 q_1} S & \nabla^2_{q_2} S \end{matrix} \tag{42}$$

To calculate the full propagator $= -\left[\nabla^2_q S\right]^{-1}$, we will assume invertibility of the upper block $^0 = -\left[\nabla^2_{q_1} S\right]^{-1}$ and use this in the Schur complement

$$\begin{aligned} &= -\left[\nabla^2_q S\right]^{-1} = \begin{matrix} ^{11} & ^{12} \\ ^{21} & ^{22} \end{matrix} \\ ^{11} &= {}^0 - {}^0\left[\nabla^2_{q_1 q_2} S\right]\left[\nabla^2_{q_2} S + (\nabla^2_{q_2 q_1} S)\,{}^0(\nabla^2_{q_1 q_2} S)\right]^{-1}\left[\nabla^2_{q_2 q_1} S\right]{}^0 \\ ^{12} &= {}^{>}_{21} = -{}^0\left[\nabla^2_{q_1 q_2} S\right]\left[\nabla^2_{q_2} S + (\nabla^2_{q_2 q_1} S)\,{}^0(\nabla^2_{q_1 q_2} S)\right]^{-1} \\ ^{22} &= -\left[\nabla^2_{q_2} S + (\nabla^2_{q_2 q_1} S)\,{}^0(\nabla^2_{q_1 q_2} S)\right]^{-1} \end{aligned} \tag{43}$$

We now need to solve for $^0 = -\left[\nabla^2_{q_1} S\right]^{-1}$. To perform this inverse, we again partition q_1 into two sets of order parameters $q_1 = [q_1^1, q_1^2]$ where $q_1^1 = \text{Vec}\{\Phi^\ell(t,s), G^\ell(t,s), K(t), \Delta^\ell(t)\}$ and $q_1^2 = \text{Vec}\{\hat{\Phi}^\ell(t,s), \hat{G}^\ell(t,s), \hat{K}(t), \hat{\Delta}^\ell(t)\}$

$$\nabla^2_{q_1} S = \begin{matrix} 0 & U^\top \\ U & \end{matrix}, \quad \equiv \nabla^2_{q_1^2} S\,,\ U \equiv \nabla^2_{q_1^2 q_1^1} S \tag{44}$$

We seek a physically sensible inverse where the variance of q_1^2 is vanishing [51, 53]. This leads to the following sub-propagator ^{0}

$$^0 = -[\nabla^2_{q_1} S]^{-1} = \begin{matrix} U^{-1}[U^{-1}]^\top & -U^{-1} \\ -[U^\top]^{-1} & 0 \end{matrix} \tag{45}$$

Thus given $, U$, we can solve for ^{0} and ultimately for the full propagator $$. The relevant entries in $$ and U are given by those second derivatives calculated above. We note that each of the field derivatives needed for U can be computed implicitly from the field dynamics. For example, for the $\Delta^\ell(t)$ derivatives we have

$$\begin{aligned} \frac{\partial}{\partial \Delta^{\ell_0}(t^0)} h^\ell(t) &= \gamma \Theta(t-t^0)\Phi^{\ell 1}_{\ell_0}(t,t^0) g^\ell(t^0) \\ &\quad + \gamma \int_0^t ds \sum_\ell \left[A^{\ell 1}(t,s) + \Phi^{\ell 1}(t,s)\Delta^\ell(s) \right] \frac{\partial g^\ell(s)}{\partial \Delta^{\ell_0}(t^0)} \\ \frac{\partial}{\partial \Delta^\ell(t^0)} z^\ell(t) &= \gamma \Theta(t-t^0) G^{\ell+1}(t,t^0) \phi(h^\ell(t^0)) \\ &\quad + \gamma \int_0^t ds \sum_\ell \left[B^\ell(t,s) + G^{\ell+1}(t,s)\Delta^\ell(s) \right] \frac{\partial \phi(h^\ell(s))}{\partial \Delta^{\ell_0}(t^0)} \end{aligned} \tag{46}$$

These can then be used in the averages such as $\left\langle \frac{\partial}{\partial\, {}^{\ell_0}(t^0)} \phi(h^\ell(t)) \phi(h^\ell(s)) \right\rangle$. Similarly, we can compute terms such as $\frac{\partial h^\ell(t)}{\partial\, {}^{\ell}(t^0,s^0)}$ through the following closed equations

$$\begin{aligned} \frac{\partial h^\ell(t)}{\partial \Phi^{\ell 1}(t^0,s^0)} &= \gamma \delta(t-t^0)\delta \Theta(t-s^0)\Delta^\ell(s^0) \\ &\quad + \gamma \int_0^t ds \sum_\ell \left[A^{\ell 1}(t,s) + \Delta^\ell(s)\Phi^{\ell 1}(t,s) \right] \frac{\partial g^\ell(s)}{\partial \Phi^{\ell}(t^0,s^0)} \\ \frac{\partial z^\ell(t)}{\partial \Phi^{\ell 1}(t^0,s^0)} &= \gamma \int_0^t ds \sum_\ell \left[B^\ell(t,s) + \Delta^\ell(s) G^{\ell+1}(t,s) \right] \frac{\partial \phi(h^\ell(s))}{\partial \Phi^{\ell 1}(t^0,s^0)} \end{aligned} \tag{47}$$

These terms can then be used to compute quantities like D^ℓ.

F Solving for the Propagator

In this section we sketch out the required steps to obtain the propagator .

- Step 1: Solve the infinite width DMFT equations for q_1 which include the prediction error dynamics $\Delta(t)$, the feature kernels $\Phi^{\ell}(t,s)$, gradient kernels $G^{\ell}(t,s)$. This step corresponds to algorithm in Bordelon & Pehlevan '22 and defines the dynamics one would expect at infinite width [9]. See below for more detail.
- Step 2: Compute the entries of the Hessian of S evaluated at the q_1 computed in the first step. Some of these entries look like fourth cumulants of features like $\kappa = \phi(h)^4 - \phi(h)^2{}^2$ and some of them measure sensitivity of one order parameter to a perturbation in another order parameter $D^{\ell} = \frac{\partial}{\partial \quad_1} \phi(h^{\ell})^2$. The averages $\langle\rangle$ used to calculate κ and D^{ℓ} should be performed over the infinite width stochastic processes for preactivations h^{ℓ} which are defined in equation (19).
- Step 3: After populating the entries of the block matrix for the Hesssian $\nabla^2 S$, we then calculate the propagator Σ with a matrix inversion. Since we discretized time, this is a finite dimensional matrix.

The step 1 above demands a solution to the infinite width DMFT equations (solving for the saddle point q_1). We will now give a detailed set of instructions about how the infinite width limit for q_1 is solved (step 1 above). This corresponds to the algorithm of Bordelon & Pehlevan 2022 to solve the saddle point equations $\frac{\partial}{\partial q} S(q)|_{q_1} = 0$ [9].

- Step 1: Start with a guess for the kernels $\Phi^{\ell}(t,s), G^{\ell}(t,s)$ and for the predictions through time $f(t)$. We usually use the lazy limit (e.g. $\Phi^{\ell}(t,s) = \Phi^{\ell}(0,0)$ ...) as an initial guess.
- Step 2: Sample Gaussian sources $u^{\ell}(t)$ and $r^{\ell}(t)$ based on the current covariances Φ^{ℓ} and G^{ℓ}.
- Step 3: For each sample, solve integral equations for $h(t)$ and $z(t)$.

$$h^{\ell}(t) = u^{\ell}(t) + \gamma \int_0^t ds \sum_{\quad} [A^{\ell \quad 1}(t,s) + \Phi^{\ell \quad 1}(t,s)][\dot{\phi}(h^{\ell}(s)) z^{\ell}(s)]$$
$$z^{\ell}(t) = r^{\ell}(t) + \gamma \int_0^t ds \sum_{\quad} [B^{\ell}(t,s) + G^{\ell+1}(t,s)]\phi(h^{\ell}(s)) \tag{48}$$

  These will be samples from the single site distribution for h, z
- Step 4: Average over the Monte Carlo samples to produce a new estimate of the kernels: $\Phi^{\ell}(t,s) = \phi(h^{\ell}(t))\phi(h^{\ell}(s))$. A similar procedure is performed for G^{ℓ} and the response functions A^{ℓ}, B^{ℓ}.
- Step 5: Compute the NTK estimate $K(t) = \sum_{\quad} G^{\ell+1}(t,t)\Phi^{\ell}(t,t)$ and then integrate prediction dynamics from the dynamics of the NTK $\frac{d}{dt} f(t) = \quad K(t)\Delta(t)$.
- Repeat steps 2-5 until the order parameters converge.

Below we provide a pseudocode algorithm to solve for the propagator elements.

Algorithm 1: Propagator Solver

Data: K^x, y, Initial Guesses $\{\quad^{\ell}, G^{\ell}\}_{\quad=1}^{L}$, $\{A^{\ell}, B^{\ell}\}_{\quad=1}^{L \quad 1}$, Sample count $\mathcal{S}$, Update Speed β
Result: Propagator Matrix

Solve DMFT equations with Algorithm 2 for order parameters $f(t), \Phi^{\ell}(t,s), \ldots$;
Draw $\mathcal{S}$ samples $\{u^{\ell}{}_{;n}(t)\}_{n=1}^{S} \sim \mathcal{GP}(0, \quad^{\ell \quad 1})$, $\{r^{\ell}{}_{;n}(t)\}_{n=1}^{S} \sim \mathcal{GP}(0, G^{\ell+1})$;
Integrate dynamics for each sample to get $\{h^{\ell}{}_{;n}(t), z^{\ell}{}_{;n}(t)\}_{n=1}^{S}$;
Estimate κ functions with Monte Carlo integration, for instance
$\kappa^{\ell}{}_{\quad}(t,s,t^0,s^0) = \frac{1}{S} \sum_{n \in [S]} \phi(h^{\ell}{}_{;n}(t))\phi(h^{\ell}{}_{;n}(s))\phi(h^{\ell}{}_{;n}(t^0))\phi(h^{\ell}{}_{;n}(s^0)) - \Phi^{\ell}(t,s)\Phi^{\ell}(t^0,s^0)$;
For each sample, compute field sensitivities to error signals, such as $\frac{\partial h^{\ell}{}_{;n}(t)}{\partial \quad(s)}$, and kernels $\frac{\partial h^{\ell}{}_{;n}(t)}{\partial \quad^{\ell}(t^0;s^0)}$ implicitly using equations (46) (47) ;
Use these sensitivities to compute the necessary D tensors such as $D^{\ell} = \frac{1}{S} \sum_{n \in [S]} \frac{\partial}{\partial \quad(t^0)} \phi(h^{\ell}{}_{;n}(t))\phi(h^{\ell}{}_{;n}(s))$;
Invert U matrix and compute $\quad_0$ in equation (45);
Compute the Schur-complement in equation (43) to handle the response functions ;

The above propagator solver builds on the solution to the DMFT equations which is provided below.

Algorithm 2: Alternating Monte Carlo Solution to Saddle Point Equations

Data: K^x, y, Initial Guesses $\{\ ^{\ell}, G^{\ell}\}_{\ell=1}^{L}$, $\{A^{\ell}, B^{\ell}\}_{\ell=1}^{L-1}$, Sample count $\mathcal{S}$, Update Speed β
Result: Final Kernels $\{\ ^{\ell}, G^{\ell}\}_{\ell=1}^{L}$, $\{A^{\ell}, B^{\ell}\}_{\ell=1}^{L-1}$, Network predictions through training $f\ (t)$

$^{0} = K^x \otimes \mathbf{1}\mathbf{1}^{\top}$, $G^{L+1} = \mathbf{1}\mathbf{1}^{\top}$;
while *Kernels Not Converged* **do**
  From $\{\ ^{\ell}, G^{\ell}\}$ compute $K^{NTK}(t,t)$ and solve $\frac{d}{dt} f\ (t) = \ ^{P}\ \Delta\ (t) K^{NTK}(t,t)$;
  $\ell = 1$;
  while $\ell < L+1$ **do**
    Draw $\mathcal{S}$ samples $\{u^{\ell}_{;n}(t)\}_{n=1}^{S} \sim \mathcal{GP}(0,\ ^{\ell-1})$, $\{r^{\ell}_{;n}(t)\}_{n=1}^{S} \sim \mathcal{GP}(0, G^{\ell+1})$;
    Integrate dynamics for each sample to get $\{h^{\ell}_{;n}(t), z^{\ell}_{;n}(t)\}_{n=1}^{S}$;
    Compute new $\ ^{\ell}, G^{\ell}$ estimates:
    $\tilde{\Phi}^{\ell}\ (t,s) = \frac{1}{S} \sum_{n \in [S]} \phi(h^{\ell}_{;n}(t)) \phi(h^{\ell}_{;n}(s))$, $\tilde{G}^{\ell}\ (t,s) = \frac{1}{S} \sum_{n \in [S]} g^{\ell}_{;n}(t) g^{\ell}_{;n}(s)$;
    Solve for Jacobians on each sample $\frac{\partial\ (h^{\ell}_{n})}{\partial r^{\ell\top}_{n}}$, $\frac{\partial g^{\ell}_{n}}{\partial u^{\ell\top}_{n}}$;
    Compute new $A^{\ell}, B^{\ell-1}$ estimates:
    $\tilde{A}^{\ell} = \frac{1}{S} \sum_{n \in [S]} \frac{\partial\ (h^{\ell}_{n})}{\partial r^{\ell\top}_{n}}$, $\tilde{B}^{\ell-1} = \frac{1}{S} \sum_{n \in [S]} \frac{\partial g^{\ell}_{n}}{\partial u^{\ell\top}_{n}}$;
    $\ell \leftarrow \ell + 1$;
  end
  $\ell = 1$;
  while $\ell < L+1$ **do**
    Update feature kernels: $\ ^{\ell} \leftarrow (1-\beta)\ ^{\ell} + \beta\, \tilde{}^{\ell}$, $G^{\ell} \leftarrow (1-\beta) G^{\ell} + \beta \tilde{G}^{\ell}$;
    if $\ell < L$ **then**
      Update $A^{\ell} \leftarrow (1-\beta) A^{\ell} + \beta \tilde{A}^{\ell}$, $B^{\ell} \leftarrow (1-\beta) B^{\ell} + \beta \tilde{B}^{\ell}$
    end
    $\ell \leftarrow \ell + 1$
  end
end
return $\{\ ^{\ell}, G^{\ell}\}_{\ell=1}^{L}, \{A^{\ell}, B^{\ell}\}_{\ell=1}^{L-1}, \{f\ (t)\}_{=1}^{P}$

G Leading Correction to the Mean Order Parameters

In this section we use the propagator structure derived in the last section to reason about the leading finite size correction to $\langle q \rangle$ at width N. Letting the indices i, j, k, n enumerate all entries of the order parameters in q (technically this is a sum over samples and an integral over time for gradient flow), we find the leading Pade Approximant for the mean has the form (App D)

$$\langle q_i - q_i^{1} \rangle_N = \frac{N \langle (q_i - q_i^{1}) V \rangle_{\ 1} + \frac{N^2}{2} \langle (q_i - q_i^{1}) V^2 \rangle_{\ 1} \cdots}{1 + N \langle V \rangle_{\ 1} + \frac{N^2}{2} \langle V^2 \rangle_{\ 1} + \ldots}$$

$$\sim \frac{1}{3! N} \sum_{jkl} \frac{\partial^3 S}{\partial q_j \partial q_k \partial q_l} \langle \delta_i \delta_j \delta_k \delta_l \rangle_{\ 1} + \mathcal{O}(N^{\ -2}). \tag{49}$$

$$= \frac{1}{2N} \sum_{jkl} \frac{\partial^3 S}{\partial q_j \partial q_k \partial q_l} \Sigma_{ij} \Sigma_{kl} + \mathcal{O}(N^{\ -2}) \tag{50}$$

where $\delta_j = \sqrt{N}(q_j - q_j^{1})$ and the derivatives are computed at the saddle point. In the last line, we utilized Wick's theorem and the permutation symmetry of the third derivative $\frac{\partial^3 S}{\partial q_i \partial q_j \partial q_k}$ to evaluate the four point averages in terms of the propagator Σ_{ij}, which was provided in the preceding section E. In practice computing even the full set of second derivatives for the DMFT action to get Σ is quite challenging. Despite the challenge of computing the mean order parameter correction, these corrections are relevant in practice and crucially distinguish the training timescales of deep networks at different widths as we show in Figures 7 and A.4.

G.1 Correction to Mean Predictions and Full MSE Correction

Supposing that we solved for the propagator , using the formalism in the preceeding section, we can compute the $\mathcal{O}(N^{-1})$ correction to the average network prediction error due to finite size. We let $\langle (t) \rangle$ represent the average of errors over an ensemble of width N networks.

$$\begin{aligned}
\frac{d}{dt}\langle \Delta (t)\rangle &= - \sum \langle K (t) \Delta (t)\rangle \\
&= - \sum \langle K (t)\rangle \langle \Delta (t)\rangle - \sum \mathrm{Cov}\left(K (t), \Delta (t)\right) \\
&\sim - \sum \langle K (t)\rangle \langle \Delta (t)\rangle - \frac{1}{N} \sum \Sigma^{K} (t,t) + \mathcal{O}(N^{-2})
\end{aligned} \tag{51}$$

where $\Sigma^{K} (t,t)$ is the leading covariance (propagator element) between the kernel $K (t)$ and prediction error $\Delta (t)$. We see that the average kernel $\langle K (t)\rangle$ (which depends on the finite width N) plays an important role in characterizing the timescales of the average prediction dynamics. Once this equation is solved for $\langle \Delta (t)\rangle$, the square loss at width N and time t has the form

$$\sum \Delta (t)^2 \sim 1 - \frac{2}{N} \sum \Delta^{1} (t)^2 + \frac{2}{N} \sum \langle \Delta (t)\rangle_{1} \Delta^{1} (t) + \frac{1}{N} \sum \Sigma (t,t) + \mathcal{O}(N^{-2}) \tag{52}$$

We will now comment on the structure of the cross term in this above solution. First, if $\langle \boldsymbol{K} \rangle \succeq \boldsymbol{K}^{1}$ and Σ^{K} is negligible then the average errors at finite width will decay more rapidly than the infinite width model. However, we suspect that in general, $\langle \boldsymbol{K} \rangle - \boldsymbol{K}^{1}$ contains many negative eigenvalues since signal propagation at finite width tends to reduce the scale of feature kernels [14]. We suspect that this is the cause of the slower dynamics of ensembled predictors for narrower networks in Figure 7 and Figure A.4. Additionally, the term involving Σ^{K} will generically increase the cross term since the dynamics of Δ cause its fluctuations to become anti-correlated with the fluctuations in K. In general, it is challenging to make strong definitive statements about the relative scale of these competing effects on the cross term. However, we can say more about this solution in the lazy limit, where we find that the cross term will generically be positive, leading to larger MSE (Appendix H.2).

G.2 Perturbation Theory in Rates rather than Predictions

In experiments on deep CNNs trained on CIFAR-10 in 7 and A.4, we find that the loss curves for the ensemble averaged predictors are effectively time rescaled by a function of network width. In this section, we argue that a proper way to account for this is to compute a perturbation expansion in the *exponent* which defines the rate of decay of the training errors. To illustrate the point, we first consider the case of a single training example before describing larger datasets. In this case, we consider the change of variables $\Delta(t) = e^{-r(t)} y$. We now treat r as an order parameter of the theory with dynamics

$$\frac{d}{dt} r(t) = K(t) \tag{53}$$

Note that this equation is now a linear relation between two order parameters $(r(t), K(t))$, whereas the relation was previously quadratic. In the lazy limit, if $K \to K - \epsilon$ then $r \to r - \epsilon t$, giving an effective rescaling of training time by $1 - \overline{K}$.

For multiple training examples, we introduce the notion of a transition matrix $\boldsymbol{T}(t) \in \mathbb{R}^{P \times P}$ which has dynamics

$$\frac{d}{dt} \boldsymbol{T}(t) = -\boldsymbol{K}(t) \boldsymbol{T}(t) \,, \; \boldsymbol{T}(0) = \mathbb{I}. \tag{54}$$

The solution to the training prediction errors can be obtained at any time t by multiplying the initial condition $(0) = \boldsymbol{y}$ with the transition matrix $(t) = \boldsymbol{T}(t)\boldsymbol{y}$, where $\boldsymbol{y}$ are the training targets. In this case, the relevant *rate matrix*, which would be an alternative order parameter is

$$\boldsymbol{R}(t) = -\log \boldsymbol{T}(t) \tag{55}$$

where log is the matrix logarithm function. Note that in general $\boldsymbol{T}(t)$ admits a Peano-Baker series solution [62–64]. In the special case where $\boldsymbol{K}(t)$ commutes with $\bar{\boldsymbol{K}}(t) = \frac{1}{t}\int_0^t ds\, \boldsymbol{K}(s)$, we obtain the following simplified formula for the rate matrix $\boldsymbol{R}$

$$\boldsymbol{R}(t) = \int_0^t ds\ \boldsymbol{K}(s) \tag{56}$$

The benefit of this representation is the elimination of coupled order parameter dynamics which are quadratic in fluctuations (in and $\boldsymbol{K}$) into a linear dynamical relation between order parameters $\boldsymbol{R}$ and $\boldsymbol{K}$. An expansion in $\boldsymbol{R}$ will thus give better predictions at long times t than a direct expansion in . In the lazy $\gamma \to 0$ limit, the constancy of $\boldsymbol{K}(t) = \boldsymbol{K}$ gives the further simplification $\boldsymbol{R} = \boldsymbol{K}t$. Working with this representation, we have the following finite width expression for the training loss

$$\begin{aligned} | \quad (t)|^2 \ &= \boldsymbol{y}^\top \langle \exp\left(-2\boldsymbol{R}(t)\right)\rangle \boldsymbol{y} \\ &\sim \boldsymbol{y}^\top \exp \quad -2 \quad \boldsymbol{R}_{\boldsymbol{1}}(t) + \frac{1}{N}\boldsymbol{R}^{\boldsymbol{1}}(t) \qquad \boldsymbol{y} \\ &+ \frac{1}{2} \sum \Sigma^{\boldsymbol{R}} \quad (t,t)\frac{\partial^2}{\partial R \quad \partial R}\boldsymbol{y}^\top \exp\left(-2\boldsymbol{R}\right)\boldsymbol{y}|_{\boldsymbol{R}=\boldsymbol{R}_{\boldsymbol{1}}(t)+\frac{1}{N}\boldsymbol{R}^{\boldsymbol{1}}(t)} + \mathcal{O}(N^{-2}) \end{aligned} \tag{57}$$

where $\langle \boldsymbol{R}\rangle \sim \boldsymbol{R}_{\boldsymbol{1}} + \frac{1}{N}\boldsymbol{R}^{\boldsymbol{1}} + \mathcal{O}(N^{-2})$ is the leading correction to the mean $\boldsymbol{R}$. In this representation, it is clear that finite width can alter the timescale of the dynamics through a correction to the mean of $\boldsymbol{R}$, as well as contribute an additive correction from fluctuations. This justifies the study perturbation analysis of rates R_N as a function of $1/N$ in Figures 7 and A.4.

H Variance in the Lazy Limit

We can simplify the propagator equations in the lazy $\gamma \to 0$ limit. To demonstrate how to use our formalism, we go through the complete process of inverting the Hessian, however, for this case, this procedure is a bit cumbersome. A simplified derivation for the lazy limit can be found below in section H.1 which relies only on linearizing the dynamics around the infinite width solution. In the $\gamma \to 0$ limit, all of the D tensors vanish and the κ tensors are constant in time. Thus, it suffices to analyze the kernels restricted to $t = 0$ and study the evolution of the prediction variance (t).

$$\begin{aligned} S = &\int dt \sum \hat{\Delta}\ (t) \quad \Delta\ (t) - y\ + \int ds \sum \Theta(t-s)K \quad \Delta\ (s) \\ &+ \sum_{\ell} \sum \hat{\Phi}^{\ell}\ \Phi^{\ell}\ + G^{\ell}\ \hat{G}^{\ell} + \sum \hat{K} \quad K \quad - \sum_{\ell} G^{\ell+1}\Phi^{\ell} + \sum_{\ell} \ln \mathcal{Z}^{\ell} \\ \mathcal{Z}^{\ell} = &\mathsf{E}_{fu^{\ell}\ g;fr^{\ell}\ g} \exp \quad - \sum \hat{\Phi}^{\ell}\ \phi(u^{\ell})\phi(u^{\ell}) - \sum \hat{G}^{\ell}\ g^{\ell} g^{\ell} \quad , g^{\ell} = r^{\ell}\dot{\phi}(u^{\ell}) \end{aligned} \tag{58}$$

where $\{u^{\ell}\} \sim \mathcal{N}(0, \quad ^{\ell-1}), \{r^{\ell}\} \sim \mathcal{N}(0, \boldsymbol{G}^{\ell+1})$. Taking two derivatives with respect to $\{\hat{\Phi}^{\ell}, \hat{G}^{\ell}\}$ give terms of the form

$$\begin{aligned} \kappa^{\ell} \quad &= \ \phi(u^{\ell})\phi(u^{\ell})\phi(u^{\ell})\phi(u^{\ell}) \ - \Phi^{\ell}\ \Phi^{\ell} \\ \kappa^{G^{\ell}} \quad &= \ g^{\ell} g^{\ell} g^{\ell} g^{\ell} \ - G^{\ell}\ G^{\ell} \\ \kappa^{\ell;G^{\ell}} \quad &= \ \phi(u^{\ell})\phi(u^{\ell}) g^{\ell} g^{\ell} \ - \Phi^{\ell}\ G^{\ell} \end{aligned} \tag{59}$$

Given these we also have the relevant non-vanishing sensitivity tensors

$$D^{\ell+1\ \ell} = \frac{\partial^2}{\partial \Phi^{\ell}} \quad \phi(u^{\ell+1})\phi(u^{\ell+1}) \quad , \; D^{G^{\ell}\, G^{\ell+1}} = \frac{\partial}{\partial G^{\ell+1}} \quad g^{\ell} g^{\ell}$$

$$D^{G^{\ell}\ \ell-1} = \frac{\partial}{\partial \Phi^{\ell-1}} \quad g^{\ell} g^{\ell} \tag{60}$$

$$D^{K^{\ell}\ \ell} = \delta \quad \delta \quad G^{\ell+1}, \; D^{K G^{\ell}} = \delta \quad \delta \quad \Phi^{\ell-1}$$

$$D^{\ K}(t) = \int ds \Theta(t-s)\delta \quad \Delta^{\ell}(s) \tag{61}$$

As before we let $q_1 = \text{Vec}\{\Delta^{\ell}(t), \Phi^{\ell}, G^{\ell}, K^{\ell}\}$ and $q_2 = \text{Vec}\{\hat{\Delta}^{\ell}(t), \hat{\Phi}^{\ell}, \hat{G}^{\ell}, \hat{K}^{\ell}\}$. The propagator has the form

$$U \equiv \nabla^2_{q_2 q_1} S = \begin{bmatrix} I + & K & 0 & 0 & D^{K} \\ 0 & & I - D & ; & 0 \\ 0 & & -D^{G} & & I - D^{GG} \\ 0 & & -D^{K} & & -D^{KG} \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & & G & 0 \\ 0 & & GG & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \; \nabla^2_{q_2 q_2} S = \begin{bmatrix} 0 & & & \\ & & ; & \\ & & & \\ & & & \end{bmatrix} \tag{62}$$

The propagator of interest is $_{q_1} = U^{-1} \nabla^2_{q_2 q_2} S \; U^{-1>}$. We can exploit the block structure of U to find an inverse

$$U^{-1} = \begin{bmatrix} U^{-1} & U^{-1} & U^{-1}_{G} & U^{-1}_{K} \\ 0 & U^{-1} & 0 & 0 \\ 0 & U_{G}^{-1} & U^{-1}_{GG} & 0 \\ 0 & U_{K}^{-1} & U^{-1}_{KG} & I \end{bmatrix} \tag{63}$$

where each sub-block can be computed with the Schur-complement formula. Altogether, we multiply through to get the propagator

$$= \begin{bmatrix} 0 & U^{-1} & + U^{-1}_{G} \; G & U^{-1} \; G + U^{-1}_{G} \; GG & 0 \\ 0 & & U^{-1} & U^{-1} \; G + U^{-1}_{G} \; GG & 0 \\ 0 & U_{G}^{-1} & + U^{-1}_{GG} \; G & U_{G}^{-1} \; G + U^{-1}_{GG} \; GG & 0 \\ 0 & U_{K}^{-1} & + U^{-1}_{KG} \; G & U_{K}^{-1} \; G + U^{-1}_{KG} \; GG & 0 \end{bmatrix}$$
$$\times \begin{bmatrix} U^{-1} & 0 & 0 & 0 \\ [U^{-1}]^{>} & U^{-1} & [U_{G}^{-1}]^{>} & [U_{K}^{-1}]^{>} \\ [U^{-1}_{G}]^{>} & 0 & U^{-1}_{GG} & [U^{-1}_{KG}]^{>} \\ [U^{-1}_{K}]^{>} & 0 & 0 & I \end{bmatrix} \tag{64}$$

Two of these blocks corresponding to K, Δ are especially important for characterizing the fluctuations of network predictions. The covariance structure for K has the form

$$_{K} = U_{K}^{-1} \quad [U_{K}^{-1}]^{>} + U_{KG}^{-1} \; G \; [U_{K}^{-1}]^{>} + U_{K}^{-1} \quad G[U_{KG}^{-1}]^{>} + U_{KG}^{-1} \; GG[U_{KG}^{-1}]^{>} \tag{65}$$

Next we use the fact that $U^{-1} = U^{-1}_{K} U_{K}^{-1}$ and that $U^{-1}_{G} = U^{-1}_{K} U_{KG}^{-1}$, which follows from the block structure of U. Consequently we arrive at the identity

$$= U^{-1} \quad [U^{-1}]^{-1} + U^{-1}_{G} \; G \; [U^{-1}]^{-1} + U^{-1}_{G} \; G \; [U^{-1}]^{-1} + U^{-1}_{G} \; GG[U^{-1}_{G}]^{-1}$$
$$= U^{-1}_{K} \quad _{K}[U_{K}^{-1}]^{>}. \tag{66}$$

Lastly, we note that, by the Schur-complement formula that $U^{-1}_{K} = -(I + \quad _{K})^{-1} D^{\ K}$. Thus, writing $(I + \quad _{K}) \quad (I + \quad _{K})^{>} = D^{\ K} \quad _{K}[D^{\ K}]^{>}$ as an integral equation, we find

$$\Sigma \quad (t,s) + \int_0^t dt' \quad K \quad \Sigma \quad (t',s) + \int_0^s ds' \quad K \quad \Sigma \quad (t,s')$$
$$+ \int_0^t dt' \int_0^s ds' \quad K \quad K \quad \Sigma \quad (t',s') = \int_0^t \Delta^{\ell}(t') \int_0^s ds' \Delta^{\ell}(s') \Sigma^{K}, \tag{67}$$

Differentiation with respect to t and s gives a simple differential equation

$$\frac{\partial^2}{\partial t \partial s}\Sigma \quad (t,s) + \sum^{\mathsf{X}} K \quad \frac{\partial}{\partial s}\Sigma \quad (t,s) + \sum^{\mathsf{X}} K \quad \frac{\partial}{\partial t}\Sigma \quad (t,s) + \sum^{\mathsf{X}} K \quad K \quad \Sigma \quad (t,s) = \sum^{\mathsf{X}} \Delta \ (t)\Delta \ (s)\Sigma^{K} \ ; \tag{68}$$

Let $\{ \ _k\}$ be the eigenvectors of the kernel matrix K. Projecting these dynamics on the eigenspace $\Sigma_{k`}(t,s) = \ ^{>}_{k} \ (t,s) \ `$ recovers the equation in the main text

$$\frac{\partial}{\partial t} + \lambda_k \quad \frac{\partial}{\partial s} + \lambda` \quad \Sigma_{k`}(t,s) = \sum_{k^0`^0}^{\mathsf{X}} \Delta_{k^0}(t)\Delta_{`^0}(s)\Sigma^{K}_{kk^0`^0} \tag{69}$$

Replacing $\Sigma^{K} = \kappa$ recovers the equation (7) in the main text.

H.1 Perturbed Linear System

In this section, we provide a simpler derivation of the lazy limit training error variance dynamics. In this case, we merely perturb the dynamics around its infinite width value $\ (t) = \ _1(t) + \ (t)$ and $\mathsf{K} = \mathsf{K}_1 + \ ^{K}$, and keep terms only linear in these perturbations. The perturbation $\ ^{K}$ is fixed in time and the dynamics of $\ (t)$ are

$$\frac{d}{dt} \ (t) = -\mathsf{K}_1 \ (t) - \ ^{K} \ _1(t) \tag{70}$$

Projecting this equation on the eigenspace of K_1 gives

$$\frac{d}{dt}\epsilon_k \ (t) = -\lambda_k \epsilon_k(t) - \sum_{k^0}^{\mathsf{X}} \epsilon^{K}_{kk^0}\Delta^{1}_{k^0}(t) \tag{71}$$

This immediately recovers the final result of the last section

$$N \quad \frac{\partial}{\partial t} + \lambda_k \quad \frac{\partial}{\partial t} + \lambda_k \quad \epsilon_k \ (t)\epsilon_` \ (s) \ = \ \frac{\partial}{\partial t} + \lambda_k \quad \frac{\partial}{\partial t} + \lambda_k \quad \Sigma_{k`}(t,s) = \sum_{k^0`^0}^{\mathsf{X}} \Sigma^{K}_{kk^0`^0}\Delta^{1}_{k^0}(t)\Delta^{1}_{`^0}(s) \tag{72}$$

Qualitatively, the process of computing this linear correction (in ϵ^{K}) to the dynamics of is identical to the argument utilized in prior work on perturbative feature learning corrections [11]. In that context, the perturbation is caused by small amounts of feature learning, rather than initialization fluctuations.

H.2 Mean Prediction Error Correction in the Lazy Limit

Using a similar heuristic as in the preceeding section, we now consider the correction to the mean predictor $\langle \Delta \ (t)\rangle$ in the lazy limit. Taylor expanding $\langle \ (t)\rangle$ in powers of $1/N$, we find

$$\begin{aligned}
\frac{d}{dt}\langle \ (t)\rangle &= \frac{d}{dt} \ ^{1}(t) + \frac{1}{N}\frac{d}{dt} \ ^{1}(t) + ... \\
&= -\langle (\mathsf{K} - \mathsf{K}^{1} + \mathsf{K}^{1})(\ - \ ^{1} + \ ^{1})\rangle \\
&= -\mathsf{K}^{1} \ ^{1} - \mathsf{K}^{1}\langle \ - \ ^{1}\rangle \\
&\quad - \langle (\mathsf{K} - \mathsf{K}^{1})\rangle \ ^{1} - \langle (\mathsf{K} - \mathsf{K}^{1})(\ - \ ^{1})\rangle \\
&\sim -\mathsf{K}^{1} \ ^{1} - \frac{1}{N}\mathsf{K}^{1} \ ^{1} - \frac{1}{N}\mathsf{K}^{1} \ ^{1} - \frac{1}{N} \ ^{K} \ _1 + \mathcal{O}(N^{-2})
\end{aligned} \tag{73}$$

From the previous section we have that

$$\frac{d}{dt} \ = -\mathsf{K}^{1} \ - \ ^{K} \ ^{1} \implies \ (t) = -\int_0^t ds \exp\left(-\mathsf{K}^{1}(t-s)\right) \ ^{K} \exp\left(-\mathsf{K}^{1}s\right) \mathsf{y} \tag{74}$$

Projecting these dynamics onto the eigenspace of the kernel gives

$$\epsilon_k(t) = -\sum_\ell \epsilon_k^\ell \frac{e^{-\lambda_\ell t} - e^{-\lambda_k t}}{\lambda_k - \lambda_\ell} y_\ell \tag{75}$$

where $\ell = k$ should be seen as the limit where $\lambda_k \to \lambda_\ell$ of the above. Thus we find that the leading mean correction to the error solves the following differential equation

$$\begin{aligned}\frac{d}{dt} + \lambda_k \quad \Delta_k^1(t) &= -\sum_\ell K_{k\ell}^1 y_\ell e^{-\lambda_\ell t} + \sum_{\ell\ell_0} \Sigma_{k\ell\ell_0}^K \frac{e^{-\lambda_{\ell_0} t} - e^{-\lambda_\ell t}}{\lambda_\ell - \lambda_{\ell_0}} y_{\ell_0}. \\ &= \sum_\ell y_\ell e^{-\lambda_\ell t} \left(-K_{k\ell}^1 + \Sigma_{k\ell\ell} t\right) + \sum_{\ell \neq \ell_0} \Sigma_{k\ell\ell_0}^K \frac{e^{-\lambda_{\ell_0} t} - e^{-\lambda_\ell t}}{\lambda_\ell - \lambda_{\ell_0}} y_{\ell_0}\end{aligned} \tag{76}$$

We see that at late sufficiently large t, that the terms involving Σ^K will dominate. We can gain more intuition by considering the special case of a single training data point where the mean error correction has the form

$$\begin{aligned}&\frac{d}{dt} + \lambda \quad \Delta^1(t) = y e^{-\lambda t}\left(-K^1 + t\Sigma^K\right) \implies \Delta^1(t) = y\left(-tK^1 + \frac{1}{2}t^2\Sigma^K\right) e^{-\lambda t} \\ &\implies \Delta(t)^2 \sim \Delta^1(t)^2 + \frac{1}{N}\left(2y^2 t e^{-2\lambda t}\left(-K^1 + \frac{1}{2}t\Sigma^K\right) + \Sigma(t,t) + \mathcal{O}(N^{-2})\right) \\ &\sim \Delta^1(t)^2 + \frac{2}{N} y^2 t e^{-2\lambda t}\left(-K^1 + \Sigma^K t\right) + \mathcal{O}(N^{-2})\end{aligned} \tag{77}$$

While the term involving Σ^K is positive for all t, K^1 could be positive or negative for a given architecture. If K^1 is positive, then MSE is initially improved at early times but after $t > \frac{K^1}{K}$ the MSE is worse than the infinite width. On the other hand, if K^1 is negative (as we suspect is typically the case), then the MSE will strictly decrease with network width for any time t.

I Two Layer Equations and Time/Time Diagonal

In this section, we analyze two layer networks in greater detail. Unlike the deep network case, two layer networks can be analyzed on the time-time diagonal: ie the dynamics only depend on $\Phi(t,t)$ and $G(t,t)$ rather than on all possible off-diagonal pairs of time points. Further, there are no response functions A^ℓ, B^ℓ which complicate the recipe for calculating the propagator (Appendix E).

I.1 A Single Training Point

For a two layer network trained on a single training point with norm constraint $|x|^2 = D$, we have the following DMFT action

$$\begin{aligned}&S[\{K(t), \hat{K}(t), \Delta(t), \hat{\Delta}(t)\}] \\ &= \int dt \left(K(t)\hat{K}(t) + \hat{\Delta}(t)\right)\left(\Delta(t) - y + \int ds\, \Theta(t-s)\Delta(s)K(s)\right) \\ &\quad + \ln \mathcal{Z}[\hat{K}, f]\,,\ \mathcal{Z} = \mathbb{E}_{h;g} \exp\left(-\int dt \hat{K}(t)[\phi(h(t))^2 + g(t)^2]\right)\,.\end{aligned} \tag{78}$$

The saddle point equations are

$$\begin{aligned}\frac{\partial S}{\partial \hat{K}(t)} &= K(t) - [\phi(h(t))^2 + g(t)^2] = 0 \\ \frac{\partial S}{\partial \hat{\Delta}(t)} &= \Delta(t) - y + \int ds\, \Theta(t-s)\Delta(s)K(s) = 0 \\ \frac{\partial S}{\partial K(s)} &= \hat{K}(s) + \Delta(s) \int dt\, \hat{\Delta}(t)\Theta(t-s) = 0 \\ \frac{\partial S}{\partial \Delta(s)} &= \hat{\Delta}(s) + K(s) \int dt\, \hat{\Delta}(t)\Theta(t-s) = 0\end{aligned} \tag{79}$$

From these equations, we can compute the entries in the Hessian of the DMFT action S. Letting $\boldsymbol{q}(t) = \begin{matrix}\Delta(t)\\ K(t)\end{matrix}$ and $\hat{\boldsymbol{q}}(t) = \begin{matrix}\hat{\Delta}(t)\\ \hat{K}(t)\end{matrix}$

$$\begin{aligned}
\frac{\partial^2 S}{\partial \boldsymbol{q}(t)\partial \boldsymbol{q}(s)^\top} &= \mathbf{0} \\
\frac{\partial^2 S}{\partial \hat{\boldsymbol{q}}(t)\partial \boldsymbol{q}(s)^\top} &= \begin{bmatrix} \delta(t-s) + \Theta(t-s)K(s) & \Theta(t-s)\Delta(s) \\ -\frac{\partial}{\partial \Delta(s)}(\phi(h(t))^2 + g(t)^2) & \delta(t-s) \end{bmatrix} \\
\frac{\partial^2 S}{\partial \hat{\boldsymbol{q}}(t)\partial \hat{\boldsymbol{q}}(s)^\top} &= \begin{bmatrix} 0 & 0 \\ 0 & \kappa(t,s) \end{bmatrix}
\end{aligned} \tag{80}$$

where $\kappa(t,s) = \langle (\phi(h(t))^2 + g(t)^2)(\phi(h(s))^2 + g(s)^2) \rangle - K(t)K(s)$ is the NTK's fourth cumulant. We now vectorize our order parameters over time $\boldsymbol{q} = \text{Vec}\{\boldsymbol{q}(t)\}_{t \in \mathbb{R}_+}$ and $\hat{\boldsymbol{q}} = \text{Vec}\{\hat{\boldsymbol{q}}(t)\}_{t \in \mathbb{R}_+}$ and express the full Hessian

$$\nabla^2 S = \begin{bmatrix} \mathbf{0} & \frac{\partial^2 S}{\partial \boldsymbol{q} \partial \hat{\boldsymbol{q}}^\top} \\ \frac{\partial^2 S}{\partial \hat{\boldsymbol{q}} \partial \boldsymbol{q}^\top} & \frac{\partial^2 S}{\partial \hat{\boldsymbol{q}} \partial \hat{\boldsymbol{q}}^\top} \end{bmatrix} \implies -[\nabla^2 S]^{-1} = \begin{bmatrix} (\frac{\partial^2 S}{\partial \hat{\boldsymbol{q}} \partial \boldsymbol{q}^\top})^{-1} \frac{\partial^2 S}{\partial \hat{\boldsymbol{q}} \partial \hat{\boldsymbol{q}}^\top} (\frac{\partial^2 S}{\partial \boldsymbol{q} \partial \hat{\boldsymbol{q}}^\top})^{-1} & -(\frac{\partial^2 S}{\partial \hat{\boldsymbol{q}} \partial \boldsymbol{q}^\top})^{-1} \\ -(\frac{\partial^2 S}{\partial \boldsymbol{q} \partial \hat{\boldsymbol{q}}^\top})^{-1} & \mathbf{0} \end{bmatrix} \tag{81}$$

The covariance matrix of interest (for $\boldsymbol{q}(t)$) is thus

$$_{\boldsymbol{q}} = \begin{bmatrix} \mathbf{I} + \quad _K \\ -\boldsymbol{D} \end{bmatrix}^{-1} \begin{bmatrix} 0 & 0 \\ 0 & \end{bmatrix} \begin{bmatrix} \mathbf{I} + \quad _K \\ -\boldsymbol{D} \end{bmatrix}^{-1\top}. \tag{82}$$

where $[\quad _K](t,s) = \Theta(t-s)K(s)$ and $[\quad\ ](t,s) = \Theta(t-s)\Delta(s)$. The above equations allow one to use the infinite width DMFT dynamics for $K(t), \Delta(t)$ to compute the finite size fluctuation dynamics of the kernel K and the error signal Δ.

I.1.1 Computing Field Sensitivities

In this section, we compute $D(t,s)$ by solving for the sensitivity of order parameters. We start with the DMFT field equations

$$h(t) = u + \gamma \int_0^t ds \Delta(s) g(s) \ , \ z(t) = r + \gamma \int_0^t ds \Delta(s) \phi(h(t)). \tag{83}$$

Now, differentiating both sides with respect to $\Delta(s^0)$ gives

$$\begin{aligned}
\frac{\partial h(t)}{\partial \Delta(s^0)} &= \gamma \Theta(t - s^0) g(s^0) + \gamma \int_0^t ds \Delta(s) \frac{\partial g(s)}{\partial \Delta(s^0)} \\
\frac{\partial z(t)}{\partial \Delta(s^0)} &= \gamma \Theta(t - s^0) \phi(h(s^0)) + \gamma \int_0^t ds \Delta(s) \frac{\partial \phi(h(s))}{\partial \Delta(s^0)}.
\end{aligned} \tag{84}$$

We can compute D Monte carlo by iteratively solving the above equations for each sampled trajectory $\{h(t), z(t)\}$ [65, 46]. Averaging the necessary fields over the Monte Carlo samples will give us the final expressions for $D(t,s)$.

$$D(t,s) = \quad \frac{\partial}{\partial \Delta(s)} \langle \phi(h(t))^2 + g(t)^2 \rangle \tag{85}$$

Similarly, the uncoupled kernel variance $\kappa(t,s)$ can be evaluated via Monte Carlo sampling for nonlinear networks.

I.2 Test Point Fluctuation Dynamics

We now are in a position to calculate the test/train kernel and test prediction fluctuations. To do this systematically, we augment S with the test point prediction $f_?$ and field $h_?$ and introduce the kernel

$K_?(t) = \langle \phi(h(t))\phi(h_?(t)) + g(t)g_?(t) \rangle$. The test prediction $f_?$ and field $h_?$ have dynamics

$$h_?(t) = u_? + \gamma \int_0^t ds \Delta(s) \dot{\phi}(h_?(s)) z(s) K_?^x \ , \ \langle u_? u \rangle = K_?^x$$

$$\frac{\partial}{\partial t} f_?(t) = K_?(t)\Delta(t) \ , \ K_?(t) = \langle \phi(h(t))\phi(h_?(t)) + g(t)g_?(t) \rangle \tag{86}$$

The augmented action for this DMFT has the form

$$\begin{aligned} S = & \int dt \ \hat{f}_?(t) \left[f_?(t) - \int ds \Theta(t-s)\Delta(s) K_?(s) \right] + \int dt \ \hat{K}_?(t) K_?(t) \\ & + \int dt \ \hat{\Delta}(t) \left[\Delta(t) - y + \int ds \ \Theta(t-s)\Delta(s) K(s) \right] + \int dt \ \hat{K}(t) K(t) \\ & + \ln \mathbb{E} \exp \left[- \int \hat{K}(t)(\phi(h(t))^2 + g(t)^2) - \int \hat{K}_?(t)(\phi(h(t))\phi(h_?(t)) + g(t)g_?(t)) \right] \end{aligned} \tag{87}$$

We let $q(t) = [\Delta(t), f_?(t), K(t), K_?(t)]^\top$

$$\nabla^2_{\hat{q}\hat{q}} S[q, \hat{q}] = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & & \\ 0 & 0 & ? & ?? \end{bmatrix} \ , \ \nabla^2_{\hat{q},q} S[q, \hat{q}] = \begin{bmatrix} I + K & 0 & & 0 \\ -K_? & I & 0 & -\\ -D & 0 & I & 0 \\ -D_? & 0 & 0 & I \end{bmatrix}$$

$$D(t,s) = \frac{\partial}{\partial \Delta(s)} (\phi(h(t))^2 + g(t)^2) \tag{88}$$

$$D_?(t,s) = \frac{\partial}{\partial \Delta(s)} (\phi(h(t))\phi(h_?(t)) + g(t)g_?(t)) \tag{89}$$

Our total covariance matrix / propagator is thus

$$= \begin{bmatrix} I + K & 0 & & 0 \\ -K_? & I & 0 & -\\ -D & 0 & I & 0 \\ -D_? & 0 & 0 & I \end{bmatrix}^{-1} \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & & \\ 0 & 0 & ? & ?? \end{bmatrix} \begin{bmatrix} I + K & 0 & & 0 \\ -K_? & I & 0 & -\\ -D & 0 & I & 0 \\ -D_? & 0 & 0 & I \end{bmatrix}^{-1 \top} \tag{90}$$

This is the equation provided in the main text Equation (8).

I.3 Two Layer Linear Network Closed Form

For a linear network on a single data point, we can compute $D(t,s)$ and $\kappa(t,s)$ analytically. We start from the field equations

$$\frac{dh(t)}{dt} = \gamma \Delta(t) z(t) \ , \ \frac{dz(t)}{dt} = \gamma \Delta(t) h(t) \tag{91}$$

We can make a change of variables $v_+(t) = \frac{1}{\sqrt{2}}(h(t) + z(t))$ and $v_-(t) = \frac{1}{\sqrt{2}}(h(t) - z(t))$. We note that $v_+(0) = \frac{1}{\sqrt{2}}(u + r)$ and $v_-(0) = \frac{1}{\sqrt{2}}(u - r)$ are independent Gaussians. These functions $v_+(t), v_-(t)$ satisfy dynamics

$$\begin{aligned} & \frac{dv_+}{dt} = \gamma \Delta(t) v_+(t) \ , \ \frac{dv_-(t)}{dt} = -\gamma \Delta(t) v_-(t) \\ & \implies v_+(t) = \exp \left[\gamma \int_0^t ds \Delta(s) \right] v_+(0) \implies \frac{\partial v_+(t)}{\partial \Delta(s)} = \gamma v_+(t) \Theta(t-s) \\ & \implies v_-(t) = \exp \left[-\gamma \int_0^t ds \Delta(s) \right] v_-(0) \implies \frac{\partial v_+(t)}{\partial \Delta(s)} = -\gamma v_-(t) \Theta(t-s) \end{aligned} \tag{92}$$

Now, we use the fact that $v_+(0) = \frac{1}{\sqrt{2}}(u+r)$ and $v\ \ (0) = \frac{1}{\sqrt{2}}(u-r)$ are independent standard normal random variables to compute $K(t) = \ h(t)^2 + z(t)^2\ \ = \ v_+(t)^2 + v\ \ (t)^2$

$$\begin{aligned} D(t,s) &= \frac{\partial}{\partial \Delta(s)}\ \ h(t)^2 + z(t)^2\ \ = 2\gamma\ \ \ v_+(t)^2\ \ - \ \ v\ \ (t)^2\ \ \ \Theta(t-s) \\ &= 2\gamma\ \ \exp\ \ 2\gamma \int_0^t ds\Delta(s)\ \ - \exp\ \ -2\gamma \int_0^t ds\Delta(s)\ \ \ \Theta(t-s) \end{aligned} \tag{93}$$

This operator is causal ($D(t,s) = 0$ for $s > t$) as expected and vanishes as $t \to 0$. If we take $\gamma \to 0$, we have $D(t,s) \to 0$ which agrees with our reasoning that fields h, z only depend on Δ in the feature learning regime. Since all fields are Gaussian in the linear network case, we can use Wick's theorem to obtain the exact uncoupled kernel variance in the two layer case.

$$\begin{aligned} \kappa(t,s) &= \ (h(t)^2 + z(t)^2)(h(s)^2 + z(s)^2)\ \ - K(t)K(s) \\ &= 2\left\langle h(t)h(s)\right\rangle^2 + 2\left\langle h(t)z(s)\right\rangle^2 + 2\left\langle z(t)h(s)\right\rangle^2 + 2\left\langle z(t)z(s)\right\rangle^2 \\ &= \left\langle v_+(t)v_+(s) + v\ \ (t)v\ \ (s)\right\rangle^2 + \left\langle v_+(t)v_+(s) - v\ \ (t)v\ \ (s)\right\rangle^2 \end{aligned} \tag{94}$$

The $v\ \ (t)$ functions are those given above. Using the fact that $\ v_+(0)^2\ \ = \ v\ \ (0)^2\ \ = 1$ allows us to easily compute the single site average above.

J Multiple Samples with Whitened Data

In this section, we analyze the role that sample number plays in dynamics in a simplified model of a two layer linear network trained on whitened data. Concretely, we assume that $\frac{x\ \ x}{D} = \delta\ \ \ $. The field equations for preactivations $h\ \ (t)$ and pregradients $z(t)$ obey

$$\frac{d}{dt}h\ \ (t) = \gamma\Delta\ \ (t)z(t)\ ,\ \frac{d}{dt}z(t) = \gamma \sum_{=1}^{N} \Delta\ \ (t)h\ \ (t) \tag{95}$$

We will assume the targets have unit norm $|y|^2 = 1$ and we define the projection of $\ \ $ onto the target as $\Delta_y(t) = y \cdot \ \ (t)$. The other $P-1$ orthogonal components are denoted $\ \ _?(t)$ so that $\ \ = \Delta_y(t)y + \ \ _?(t)$ with $\ \ _?(t) \cdot y = 0$. At infinite width, $\ \ _? = 0$ and our field equations become

$$\frac{d}{dt}h_y(t) = \Delta_y(t)z(t)\ ,\ \frac{d}{dt}z(t) = \Delta_y(t)h_y(t)\ ,\ \ \ \ _?(t) = 0\ ,\ h_? \sim \mathcal{N}(0,1) \tag{96}$$

However, at finite width N, the off-target predictions $\ \ _?$ fluctuate over random initialization. To model all of the fluctuations simultaneously, we consider the following action

$$S = \gamma \int dt \sum_{\ \ } \hat{\Delta}\ \ (t)(\Delta\ \ (t) - y\ \ ) + \ln \mathbb{E} \exp \int dt \sum_{\ \ } \hat{\Delta}\ \ (t)z(t)h\ \ (t) \tag{97}$$

which enforces the constraint that $\Delta\ \ (t) = y\ \ - \frac{1}{\ \ } \left\langle z(t)h\ \ (t)\right\rangle$ at infinite width. The Hessian over order parameters $q = \text{Vec}\{\Delta\ \ (t), \hat{\Delta}\ \ (t)\}$ has the form

$$\nabla_q^2 S = \begin{matrix} 0 & (\gamma I + D)^\top \\ \gamma I + D & \end{matrix}\ \ ,\ D\ \ \ (t,s) = \ \ \frac{\partial}{\partial \Delta\ \ (s)} z(t)h\ \ (t) \tag{98}$$

We thus get the following covariance for predictions $\ \ = (\gamma I + D)^{\ \ -1}\ \ \ (\gamma I + D)^{\ \ -1\ \ \top}$. We now compute the necessary components of the D tensor

$$\begin{aligned} \frac{\partial h\ \ (t)}{\partial \Delta\ \ (s)} &= \gamma\delta\ \ \ \Theta(t-s)z(s) + \gamma \int_0^t dt'\Delta\ \ (t')\frac{\partial z(t')}{\partial \Delta\ \ (s)} \\ \frac{\partial z(t)}{\partial \Delta\ \ (s)} &= \gamma\Theta(t-s)h\ \ (s) + \gamma \int_0^t dt' \sum_{\ \ } \Delta\ \ (t')\frac{\partial h\ \ (t')}{\partial \Delta\ \ (s)} \\ &= \gamma\Theta(t-s)h\ \ (s) + \gamma \int_0^t dt'\Delta_y(t')\frac{\partial h_y(t')}{\partial \Delta\ \ (s)} \end{aligned} \tag{99}$$

In the last line, we used the fact that these equations are to be evaluated at the mean field infinite width stochastic process where $\Delta_?(t) = 0$. To compute the sensitivity tensor D, we find the following equations for our correlators of interest:

$$\begin{aligned}
\frac{\partial h\ (t)}{\partial \Delta\ (s)} z(t) &= \delta\quad \gamma\Theta(t-s)\left\langle z(s)z(t)\right\rangle\ ,\ \mu,\nu \neq y \\
\frac{\partial z(t)}{\partial \Delta\ (s)} h\ (t) &= \gamma\Theta(t-s)\delta\quad\ ,\ \mu,\nu \neq y \\
\frac{\partial h_y(t)}{\partial \Delta_y(s)} z(t) &= \gamma\Theta(t-s)\left\langle z(s)z(t)\right\rangle + \gamma \int_0^t dt' \Delta_y(t') \quad \frac{\partial z(t')}{\partial \Delta_y(s)} z(t) \\
\frac{\partial z(t)}{\partial \Delta_y(s)} z(t') &= \gamma\Theta(t-s)\left\langle h_y(s)z(t)\right\rangle + \gamma \int_0^t dt'' \Delta_y(t'') \quad \frac{\partial h_y(t'')}{\partial \Delta_y(s)} z(t')
\end{aligned} \tag{100}$$

We therefore see that the components of D decouple over indices. In the y direction, we have the following equations

$$D_y(t,s) = \quad \frac{\partial h_y(t)}{\partial \Delta_y(s)} z(t) \quad + \quad \frac{\partial z(t)}{\partial \Delta_y(s)} h_y(t) \tag{101}$$

where the correlators must be solved self-consistently. We will provide this solution in one moment, but first, we will look at the orthogonal directions. For the $P-1$ orthogonal directions, we obtain the explicit formula for D in each of these directions

$$\begin{aligned}
D_?(t,s) &= \quad \frac{\partial h_?(t)}{\partial \Delta_?(s)} z(t) \quad + \quad \frac{\partial z(t)}{\partial \Delta_?(s)} h_?(t) \\
&= \gamma\Theta(t-s)\left\langle z(t)z(s)\right\rangle + \gamma\Theta(t-s)
\end{aligned} \tag{102}$$

Now, we return to D_y. To solve these equations we utilize the change of variables employed in the single sample case $v_+(t) = \frac{1}{2}(h_y(t) + z(t)), v\ \ (t) = \frac{1}{2}(h_y(t) - z(t))$ (see Appendix I.3). This orthogonal transformation decouples the dynamics

$$\frac{d}{dt} v_+(t) = \gamma\Delta_y(t) v_+(t)\ ,\ \frac{d}{dt} v\ \ (t) = -\gamma\Delta_y(t) v\ \ (t) \tag{103}$$

As a consequence, the field derivatives close

$$\begin{aligned}
\frac{\partial v_+(t)}{\partial \Delta_y(s)} &= \gamma\Theta(t-s) v_+(s) + \int_0^t dt' \Delta_y(t') \frac{\partial v_+(t')}{\partial \Delta_y(s)} \\
\frac{\partial v\ \ (t)}{\partial \Delta_y(s)} &= -\gamma\Theta(t-s) v\ \ (s) - \int_0^t dt' \Delta_y(t') \frac{\partial v\ \ (t')}{\partial \Delta_y(s)}
\end{aligned} \tag{104}$$

The correlator of interest is

$$\left\langle h_y(t) z(t)\right\rangle = \frac{1}{2}\left\langle [v_+(t) + v\ \ (t)][v_+(t) - v\ \ (t)]\right\rangle = \frac{1}{2}\ \ v_+(t)^2 - v\ \ (t)^2 \tag{105}$$

So we get that

$$\begin{aligned}
D_y(t,s) &= \frac{1}{2} \quad \frac{\partial}{\partial \Delta_y(s)}\ \ v_+(t)^2 - v\ \ (t)^2 \\
&= \quad v_+(t) \frac{\partial v_+(t)}{\partial \Delta_y(s)} \quad - \quad v\ \ (t) \frac{\partial v\ \ (t)}{\partial \Delta_y(s)}
\end{aligned} \tag{106}$$

Similarly, we can derive the on-target and off-target uncoupled variances $\kappa_y(t,s)$ and $\kappa_?(t,s)$, which satisfy

$$\begin{aligned}
\kappa_y(t,s) &= \left\langle v_+(t)v_+(s) + v\ \ (t)v\ \ (s)\right\rangle^2 + \left\langle v_+(t)v_+(s) - v\ \ (t)v\ \ (s)\right\rangle^2 \\
\kappa_?(t,s) &= \frac{1}{2}\left\langle v_+(t)v_+(s) + v\ \ (t)v\ \ (s)\right\rangle
\end{aligned} \tag{107}$$

Using these functions, we arrive at the following variance for each of the P dimensions

$$
\begin{aligned}
{}_{y} &= (\gamma \mathsf{I} + \boldsymbol{D}_{y})^{\ 1}\ {}_{y}\,(\gamma \mathsf{I} + \boldsymbol{D}_{y})^{\ 1} \\
{}_{?} &= (\gamma \mathsf{I} + \boldsymbol{D}_{?})^{\ 1}\ {}_{?}\,(\gamma \mathsf{I} + \boldsymbol{D}_{?})^{\ 1}
\end{aligned} \tag{108}
$$

Using the fact that all $\Delta_{?}$ variables are independent and identically distributed under the leading order picture, the expected training loss has the form

$$
|\ \ |^{2}\ \approx \Delta_{y}^{1}(t)^{2} + \frac{2}{N}\Delta_{y}^{1}(t)\Delta_{y}^{1}(t) + \frac{1}{N}\Sigma\ {}_{y}(t,t) + \frac{(P-1)}{N}\Sigma\ {}_{?}(t,t) + \mathcal{O}(N^{\ 2}). \tag{109}
$$

where $\Delta_{y} - \Delta_{y}^{1} = \frac{1}{N}\Delta_{y}^{1}(t) + \mathcal{O}(N^{\ 2})$. We note that the bias correction if $\mathcal{O}(N^{\ 1})$ while the variance is $\mathcal{O}(P/N)$. We compare the above leading order theory with and without the bias correction in Appendix Figure A.2.

K Online Learning

Our technology for computing finite size effects can easily be translated to a setting where the neural network is trained in an online fashion, disregarding the effect of SGD noise. At each step, we compute the gradient over the full data distribution $p(\boldsymbol{x})$. Focusing on MSE loss, we study the following equation

$$
\frac{d}{dt}\Delta(\boldsymbol{x},t) = -\mathsf{E}_{\boldsymbol{x}^{0}\ \ p(\boldsymbol{x}^{0})}K(\boldsymbol{x},\boldsymbol{x}^{0};t)\Delta(\boldsymbol{x}^{0},t) \tag{110}
$$

where $K(\boldsymbol{x},\boldsymbol{x}^{0};t)$ is the dynamic NTK and $\Delta(\boldsymbol{x},t) = y(\boldsymbol{x}) - f(\boldsymbol{x},t)$ is the prediction error. In general the distribution involves integration over an uncountable set of possible inputs $\boldsymbol{x}$. To remedy this, we utilize a countable orthonormal basis of functions for the data distribution $\{\psi_{k}(\boldsymbol{x})\}_{k=1}^{1}$. For example, if $p(\boldsymbol{x})$ were the isotropic Gaussian density for $\mathcal{N}(0,\mathsf{I})$, then ψ_{k} could be Hermite polynomials. We expand Δ and K in this basis ψ_{k}, and arrive at the following differential equation

$$
\frac{d}{dt}\Delta_{k}(t) = -\sum_{`} K_{k`}(t)\Delta_{`}(t) \tag{111}
$$

By orthonormality, the average turned into a sum over all possible orthonormal functions $\{\psi_{k}\}$. We note that since K is evolving in time, there is not generally a fixed basis of functions that diagonalize K, resulting in the couplings across eigenmodes in Equation (111). Since, in online learning, there is no distinction between the training and test distribution, our error of interest is simply $\mathcal{L}(t) = \sum_{k}\Delta_{k}(t)^{2}$. To obtain the finite size corrections to this quantity, we compute the joint propagator for all variables $\{K_{k`}(t),\Delta_{k}(t)\}$. If we wanted to pursue a perturbation theory in rates (Appendix G.2), we could again define a transition matrix $\boldsymbol{T}$ and rate matrix $\boldsymbol{R}(t)$ as

$$
\boldsymbol{R}(t) = -\log \boldsymbol{T}(t)\ ,\ \frac{d}{dt}T_{k`}(t) = -\sum_{k^{0}} K_{kk^{0}}(t)T_{k^{0}`}(t)\ ,\ T_{k`}(0) = \delta_{k`} \tag{112}
$$

We can then obtain $\quad = \exp(-\boldsymbol{R}(t))\boldsymbol{y}$, where $y_{k} = \mathsf{E}_{\boldsymbol{x}}\psi_{k}(\boldsymbol{x})y(\boldsymbol{x})$. Since $\boldsymbol{R}$ has a finite size mean correction and finite size fluctuations, so too does the error $\Delta_{k}(t)$ and the loss $\mathcal{L}$ (Appendix G.2).

K.1 Two Layer Networks

In the two layer case, instead of tracking kernels, we could instead deal with the distribution over read-in vectors $\boldsymbol{w} \in \mathsf{R}^{D}$ and readout scalars $a \in \mathsf{R}$ as in the original works on mean field networks [6, 66]. When training on the population risk equations for $\boldsymbol{x} \sim \mathcal{N}(0,\mathsf{I})$

$$
\begin{aligned}
\frac{d}{dt}\boldsymbol{w} &= a\mathsf{E}_{\boldsymbol{x}}\Delta(\boldsymbol{x})\dot{\phi}(\boldsymbol{w}\cdot\boldsymbol{x})\boldsymbol{x} = \mathsf{E}_{\boldsymbol{x}}\frac{\partial\Delta(\boldsymbol{x})}{\partial\boldsymbol{x}}\dot{\phi}(\boldsymbol{w}\cdot\boldsymbol{x}) + \mathsf{E}\Delta(\boldsymbol{x})\ddot{\phi}(\boldsymbol{w}\cdot\boldsymbol{x})\boldsymbol{w} \\
\frac{d}{dt}a &= \mathsf{E}_{\boldsymbol{x}}\Delta(\boldsymbol{x})\phi(\boldsymbol{w}\cdot\boldsymbol{x})
\end{aligned} \tag{113}
$$

The action has the form

$$
S = \gamma \int dtd\boldsymbol{x}\hat{\Delta}(t,\boldsymbol{x})(\Delta(t,\boldsymbol{x}) - y(\boldsymbol{x})) + \ln \mathsf{E}_{a;\boldsymbol{w}} \exp \int dtd\boldsymbol{x}\hat{\Delta}(t,\boldsymbol{x})a(t)\phi(\boldsymbol{w}(t)\cdot\boldsymbol{x}) \tag{114}
$$

The Hessian over $q = \{\Delta\ (t), \hat{\Delta}\ (t)\}$ is

$$\nabla^2 S = \begin{matrix} 0 & I + D \\ I + D & \end{matrix} . \tag{115}$$

where $D\ \ (t, x; s, x^0) = \begin{matrix} D \\ \frac{@}{@\ \ (s;x^0)} a(t)\phi(w(t) \cdot x) \end{matrix} \begin{matrix} E \end{matrix}$ We can use the following implicit rule

$$\begin{aligned}
\frac{\partial a(t)}{\partial \Delta(s, x)} &= \gamma \Theta(t-s) p(x) \phi(w(s) \cdot x) + \gamma \mathbb{E}_{x^0} \int_0^t dt^0 \Delta(t^0, x^0) \dot{\phi}(w \cdot x^0) x^0 \cdot \frac{\partial w(t)}{\partial \Delta(s, x)} \\
\frac{\partial w(t)}{\partial \Delta(s, x)} &= \gamma \Theta(t-s) p(x) a(s) \dot{\phi}(w(s) \cdot x) x \\
&\quad + \gamma \mathbb{E}_{x^0} \int_0^t dt^0 \Delta(t^0, x^0)\ \ \frac{\partial a(t^0)}{\partial \Delta(s, x)} \dot{\phi}(w \cdot x^0) + a(t^0) \ddot{\phi}(w \cdot x^0) \frac{\partial w(t^0)}{\partial \Delta(s, x)} \cdot x^0
\end{aligned} \tag{116}$$

The above equations could be solved and then used to compute $D\ \ (t, x; s, x^0)$ which must then be inverted to get the observed prediction variance.

K.2 Linear Activations

Using the ideas in the preceding sections, we can make more progress in the case of a two layer linear network in the online learning setting. The key idea is to track the kernel and prediction error projections onto the space of linear functions. In this case we get the following DMFT over the order parameter $\ \ (t) = \frac{1}{N} W^{>} a \in \mathbb{R}^D$.

$$\begin{aligned}
\frac{d}{dt} a(t) &= \gamma(\ \ _? - \ \ (t)) \cdot w(t) \\
\frac{d}{dt} w(t) &= \gamma a(t)(\ \ _? - \ \ (t)) \\
\ \ (t) &= \frac{1}{\gamma} \langle a(t) w(t) \rangle
\end{aligned} \tag{117}$$

At infinite width, we see that the dynamics can be reduced to tracking the projection of the weights w and $\ \ $ on the $\ \ _?$ direction. The $D-1$ off-target dimensions vanish $\ \ _?(t) = 0$. At infinite width, we arrive at the alignment dynamics studied in prior work [64, 9]

$$\begin{aligned}
\frac{d}{dt}\ \ (t) &= M\ (t)(\ \ _? - \ \ (t)) \\
\frac{d}{dt} M\ (t) &= \gamma^2\ \ (t)(\ \ _? - \ \ (t))^{>} + \gamma^2\ \ (t)(\ \ _? - \ \ (t))^{>} \\
&\quad + 2\gamma^2(\ \ _? - \ \ (t)) \cdot \ \ (t) I
\end{aligned} \tag{118}$$

We note that $\ \ (t) = \beta(t)\ \ _?$ and that M has only one special eigenvector $\ \ _?$ with eigenvalue $m_?(t)$. It thus suffices to track evolution in this single direction

$$\frac{d}{dt}\beta(t) = m_?(t)(\beta_? - \beta(t))\ , \ \frac{d}{dt} m_?(t) = 4\gamma^2 \beta(t)(\beta_? - \beta(t)) \tag{119}$$

We note that this equation is identical to the differential equation for a single training example in Appendix J. Here $\beta_? - \beta(t)$ plays the role of $\Delta_y(t)$ and $m_?(t)$ plays the role of the kernel $K_y(t)$. A key observation is the conservation law $4\gamma^2 \frac{d}{dt}\beta(t)^2 = \frac{d}{dt} m_?(t)^2$, from which it follows that $m_?(t)^2 - 4 = 4\gamma^2 \beta(t)$ [9]

$$\frac{d}{dt}\beta(t) = 2 \sqrt{1 + \gamma^2 \beta(t)^2}(\beta_? - \beta(t)) \tag{120}$$

This is identical to the differential equations for a single sample (producing prediction $f(t)$ and kernel $K(t)$) if the following substitutions are made

$$f(t) \leftrightarrow \beta(t)\ , \ K(t) \leftrightarrow m_?(t) \tag{121}$$

We now proceed to compute finite size corrections starting from the action

$$S = \gamma \int dt\, \hat{}(t) \cdot \quad (t) + \ln \mathsf{E} \exp \quad - \int dt\, \hat{}(t) \cdot \boldsymbol{w}(t) a(t) \tag{122}$$

The necessary ingredients are

$$\begin{aligned} (t,s) &= \quad a(t)a(s)\boldsymbol{w}(t)\boldsymbol{w}(s)^{\top} \quad - \gamma^2 \quad (t) \quad (s) \\ &= \langle a(t)a(s) \rangle \quad \boldsymbol{w}(t)\boldsymbol{w}(s)^{\top} \quad + \langle a(s)\boldsymbol{w}(t) \rangle \quad a(t)\boldsymbol{w}(s)^{\top} \quad \in \mathsf{R}^{D \quad D} \end{aligned} \tag{123}$$

Similarly we have to compute the sensitivity tensor

$$\boldsymbol{D}(t,s) = \quad \frac{\partial}{\partial \quad (s)^{\top}} a(t)\boldsymbol{w}(t) \quad \in \mathsf{R}^{D \quad D} \tag{124}$$

We start from the dynamics

$$\frac{d}{dt}\boldsymbol{w}(t) = \gamma a(t)(\quad _{?} - \quad (t)) \, , \; \frac{d}{dt} a(t) = \gamma (\quad _{?} - \quad (t)) \cdot \boldsymbol{w}(t) \tag{125}$$

Next, we have to calculate causal derivatives for fields

$$\begin{aligned} \frac{\partial}{\partial \quad (s)^{\top}} \boldsymbol{w}(t) &= -\gamma \Theta(t-s) a(s) \mathsf{I} + \gamma \int_{0}^{t} dt' (\quad _{?} - \quad (t')) \frac{\partial a(t')}{\partial \quad (s)^{\top}} \\ \frac{\partial}{\partial \quad (s)} a(t) &= -\gamma \Theta(t-s) \boldsymbol{w}(s) + \gamma \int_{0}^{t} dt' (\quad _{?} - \quad (t')) \cdot \frac{\partial \boldsymbol{w}(t')}{\partial \quad (s)} \end{aligned} \tag{126}$$

Following an identical argument as in J, we see that $\boldsymbol{D}$ has block diagonal structure with $D \quad _{?}(t,s)$ on the $\quad _{?} \quad _{?}$ direction and $D_{?}(t,s)$ in any of the $D-1$ remaining directions

$$D \quad _{?}(t,s) = \quad \frac{\partial}{\partial \beta(s)} a(t) w \quad _{?}(t) \quad , \; D_{?}(t,s) = \quad \frac{\partial}{\partial \beta_{?}(s)} a(t) w_{?}(t) \tag{127}$$

Similarly, $\quad (t,s)$ has a similar decomposition

$$\begin{aligned} \kappa \quad _{?}(t,s) &= \langle a(t)a(s) \rangle \langle w \quad _{?}(t) w \quad _{?}(s) \rangle + \langle a(s) w \quad _{?}(t) \rangle \langle a(t) w \quad _{?}(s) \rangle \\ \kappa_{?}(t,s) &= \langle a(t)a(s) \rangle \langle w_{?}(t) w_{?}(s) \rangle + \langle a(s) w_{?}(t) \rangle \langle a(t) w_{?}(s) \rangle \end{aligned} \tag{128}$$

The processes have the following equations at infinite width

$$\frac{d}{dt} w \quad _{?}(t) = \gamma a(t)(\beta_{?} - \beta(t)) \, , \; \frac{d}{dt} a(t) = \gamma w \quad _{?}(t)(\beta_{?} - \beta(t)) \, , \; \frac{d}{dt} w_{?}(t) = 0 \tag{129}$$

As a consequence we note that $\langle w_{?}(t) a(s) \rangle = 0$ so that $\kappa_{?}(t,s) = \langle a(t)a(s) \rangle$. Letting $v_{+}(t) = \frac{1}{\sqrt{2}}(w \quad _{?}(t) + a(t))$ and $v \quad (t) = \frac{1}{\sqrt{2}}(w \quad _{?}(t) + a(t))$, we find the same decoupled stochastic processes as in Appendix I.3.

$$\frac{d}{dt} v_{+}(t) = \gamma(\beta_{?} - \beta(t)) v_{+}(t) \, , \; \frac{d}{dt} v \quad (t) = -\gamma(\beta_{?} - \beta(t)) v \quad (t) \tag{130}$$

We can use these equations to perform the necessary averages for $\kappa \quad _{?}$ and $D \quad _{?}$. Lastly, we use

$$\frac{\partial}{\partial \beta_{?}(s)} w_{?}(t) = -\gamma \Theta(t-s) a(s) \tag{131}$$

to evaluate $D_{?}(t,s)$. The observed covariances are just

$$\quad _{?} = (\gamma \mathsf{I} - \boldsymbol{D} \quad _{?}) \quad ^{-1} \quad _{?} (\gamma \mathsf{I} - \boldsymbol{D} \quad _{?}) \quad ^{-1 \top} \, , \quad \quad _{?} = (\gamma \mathsf{I} - \boldsymbol{D}_{?}) \quad ^{-1} \quad _{?} (\gamma \mathsf{I} - \boldsymbol{D}_{?}) \quad ^{-1 \top} \tag{132}$$

We note that these expressions are identical to those in Appendix J under the substitution $\beta_{?} - \beta(t) \to \Delta(t)$ and $D \to P$. Thus the expected test risk is

$$| \quad (t) - \quad _{?}|^2 \quad \sim (\beta(t) - \beta_{?})^2 + \frac{1}{N} \Sigma \quad _{?}(t,t) + \frac{(D-1)}{N} \Sigma \quad _{?}(t,t) + \mathcal{O}(N^{-\quad 2}) \tag{133}$$

This recovers the variance we obtained in the multiple-sample whitened data case J.

K.3 Connections to Offline Learning in Linear Model

Remark 1 *The finite size variance of generalization error in an online learning setting with linear target function $y = \cdot x$ has an identical form as the model described above. In this setting, we sample infinitely many fresh data points $x \sim \mathcal{N}(0, I)$ at each step leading to the flow $\frac{d}{dt} w_i(t) = \gamma a_i(t) E_x \Delta(x) x$ and $\frac{d}{dt} a_i(t) = \gamma w_i(t) \cdot E_x \Delta(x) x$. The order parameter of interest in this setting is $(t) = \frac{1}{N} \sum_{i=1}^{N} w_i(t) a_i(t)$. The precise correspondence between this setting and the offline setting is summarized in Table 2. We note that this argument could be extended to higher degree monomial activations as well, at the cost of tracking higher degree tensors (eg for quadratic activations $M = \frac{1}{N} \sum_{i=1}^{N} a_i w_i w_i^\top \in \mathbb{R}^{D \times D}$ is sufficient).*

Setting	Order Params.	Target	Off-target Dims.	Loss	Variance	Infinite Quantity
Offline	$= y - f$	y	$P-1$	Train	$\mathcal{O}(\frac{P}{N})$	D
Online	$? -$	$?$	$D-1$	Test	$\mathcal{O}(\frac{D}{N})$	P

Table 2: Summary of the equivalence between the leading $1/N$ correction in the offline setting and the online setting for two layer linear networks. In the offline training setting, the order parameters are the errors $= y - f \in \mathbb{R}^P$ while in the online case they are $? - \in \mathbb{R}^D$.

As in the offline case, in Fig. 4 (c) and (d) we see that the variance contribution to test loss $| - ?|^2$ increases with input dimension D. We note that this perturbative effect to the loss dynamics is reminiscent of the deviations from mean field behavior studied in SGD [43, 44], though this present work concerns fluctuations driven by initialization variance rather than stochastic sampling of data. In Fig. 4 (e) we show that richer networks have lower variance at fixed N. Similarly, leading order theory for richer networks more accurately captures their dynamics as D/N increases (Fig. 4 (f)).

L Deep Linear Networks

For deep linear networks, the fields $h^\ell(t), g^\ell(t)$ are Gaussian and have the following self-consistent equations

$$h^\ell(t) = u^\ell(t) + \gamma \int_0^t ds \sum_{} A^{\ell-1}(t,s) + \Delta(s) H^{\ell-1}(t,s)\ g^\ell(s)\ ,\ u^\ell(t) \sim \mathcal{GP}(0, H^{\ell-1})$$
$$g^\ell(t) = r^\ell(t) + \gamma \int_0^t ds \sum_{} B^\ell(t,s) + \Delta(s) G^{\ell+1}(t,s)\ h^\ell(s)\ ,\ r^\ell(t) \sim \mathcal{GP}(0, G^{\ell+1}). \tag{134}$$

where $H^\ell(t,s) = h^\ell(t) h^\ell(s)$ and $G^\ell(t,s) = g^\ell(t) g^\ell(s)$ and $A^\ell(t,s) = \frac{\partial h^\ell(t)}{\partial r(s)}$ and $B^\ell(t,s) = \frac{\partial h^\ell(t)}{\partial r(s)}$ [9]. Therefore, we express the action as a differentiable function of the order parameters by integrating over the Gaussian field distribution. For concreteness, we vectorize our fields over time and samples $h^\ell = \mathrm{Vec}\{h^\ell(t)\}_{f\ 2[P];t2\mathbb{R}_+}, g^\ell = \mathrm{Vec}\{g^\ell(t)\}_{f\ 2[P];t2\mathbb{R}_+}$ we consider the contribution of a single hidden layer.

$$\mathcal{Z}^\ell = \int d\hat{h}^\ell d\hat{g}^\ell dh^\ell dg^\ell \exp\ -\frac{1}{2}\hat{h}^\ell\ {}_u\hat{h}^\ell + i\hat{h}^\ell \cdot (h^\ell - C^\ell g^\ell) - \frac{1}{2} h^{\ell\top} \hat{H}^\ell h^\ell$$
$$\exp\ -\frac{1}{2}\hat{g}^\ell\ {}_r\hat{g}^\ell + i\hat{g}^\ell \cdot (g^\ell - D^\ell h^\ell) - \frac{1}{2} g^{\ell\top} \hat{G}^\ell g^\ell$$

where $C^\ell(t,s) = \gamma\Theta(t-s)\ A^{\ell-1}(t,s) + H^{\ell-1}(t,s)\Delta(s)$ and $D^\ell(t,s) = \gamma\Theta(t-s)\ B^\ell(t,s) + G^{\ell+1}(t,s)\Delta(s)$. Performing the joint Gaussian integrals over $(h^\ell, g^\ell, \hat{h}^\ell, \hat{g}^\ell)$ we find

$$\ln \mathcal{Z}^\ell = -\frac{1}{2} \ln \det \begin{bmatrix} -\hat{H}^\ell & 0 & I & -D^{\ell\top} \\ 0 & -\hat{G}^\ell & -C^{\ell\top} & I \\ I & -C^\ell & {}_u & 0 \\ -D^\ell & I & 0 & {}_r \end{bmatrix} \tag{135}$$

We can then automatically differentiate the DMFT action to get the propagator. For example, for a three layer linear network, the full DMFT action has the form

$$
\begin{aligned}
S = &\frac{1}{2}\mathrm{Tr}\left[\hat{H}^1 H^1 + \hat{H}^2 H^2 + \hat{G}^1 G^1 + \hat{G}^2 G^2\right] - \gamma^2 \mathrm{Tr} A B \\
&- \frac{1}{2}\ln\det\begin{bmatrix} -\hat{H}^1 & 0 & I & -D^{1\top} \\ 0 & -\hat{G}^1 & -C^{1\top} & I \\ I & -C^1 & 11^\top & 0 \\ -D^1 & I & 0 & G^2 \end{bmatrix} \\
&- \frac{1}{2}\ln\det\begin{bmatrix} -\hat{H}^2 & 0 & I & -D^{2\top} \\ 0 & -\hat{G}^2 & -C^{2\top} & I \\ I & -C^2 & H^1 & 0 \\ -D^2 & I & 0 & 11^\top \end{bmatrix}
\end{aligned} \tag{136}
$$

where $C^1 = \gamma \quad$ and $C^2 = \gamma \quad \odot H^1 + \gamma A$ and $D^1 = \gamma \quad \odot G^2 + \gamma B$ and $D^2 = \gamma \quad$. This above example can be extended to deeper networks. The total size of the block matrices which we compute determinants over is $4PT \times 4PT$ for a dataset of size P trained for T steps.

M Discrete Time Dynamics and Edge of Stability Effects

Large step size effects can induce qualitatively different dynamics in neural network training. For instance, if the step size exceeds that required for linear stability with the initial kernel, the kernel can decrease in order to stabilize the dynamics [57]. Alternatively, during training the kernel may exhibit a "progressive sharpening" phase where its top eigenvalue grows before reaching a stability bound set by the learning rate [19]. It is therefore well motivated to study how dynamics in this regime alter finite size effects in neural networks. We will first solve a special model which was considered in prior work [57]: a two layer linear network trained on a single training point. We will then provide the full DMFT equations for the discrete time case and provide an outline for how one could obtain finite size effects in that picture.

M.1 Two Layer Linear Equations

In a two layer linear network, the DMFT equations are

$$
\begin{aligned}
h(t+1) = h(t) + \eta\gamma\Delta(t) z(t)\ ,\ z(t+1) = z(t) + \eta\gamma\Delta(t) h(t) \\
f(t) = \frac{1}{\gamma}\left\langle z(t) h(t) \right\rangle
\end{aligned} \tag{137}
$$

The NTK has the form $K(t) = \quad h(t)^2 + z(t)^2 \quad$. We can easily show that the kernel and error have coupled dynamics

$$
\begin{aligned}
f(t+1) &= f(t) + \eta \quad h(t)^2 + z(t)^2 \quad \Delta(t) + \eta^2\gamma\Delta(t)^2 \left\langle h(t) z(t) \right\rangle \\
&= f(t) + \eta K(t)\Delta(t) + \eta^2\gamma^2\Delta(t)^2 f(t) \qquad (138) \\
K(t+1) &= K(t) + 4\eta\gamma\Delta(t)\left\langle h(t) z(t) \right\rangle + \eta^2\gamma^2\Delta(t)^2 \quad h(t)^2 + z(t)^2 \\
&= K(t) + 4\eta\gamma^2\Delta(t) f(t) + \eta^2\gamma^2\Delta(t)^2 K(t) \qquad (139)
\end{aligned}
$$

These equations define the infinite width evolution of $\Delta(t)$ and $K(t)$. Already at this level of analysis, we can reason about the evolution of $K(t)$. In the small η limit, we could disregard terms of order $\mathcal{O}(\eta^2)$ and arrive at the following gradient flow approximation for $K(t) \sim 2\sqrt{1+\gamma^2 f(t)^2}$ [9]. This evolution will not reach the edge of stability provided that $\eta < \frac{1}{\sqrt{1+\quad^2 y^2}}$. For large γ and $y = 1$, this leads to the constraint $\eta\gamma < 1$. However, if η exceeds this bound, the gradient flow approximation is no longer reasonable and the system reaches an edge of stability effect as shown in Figure 6.

To calculate the finite size effects, we need to compute κ and $D(t,s) = \frac{\partial}{\partial\ (s)} \quad h(t)^2 + z(t)^2 \quad$. To evaluate these quantities we utilize the same change of variables employed in Appendix I.3. In discrete time, these decoupled equations are

$$
v_+(t+1) = v_+(t) + \eta\gamma\Delta(t) v_+(t)\ ,\ v \quad (t+1) = v \quad (t) - \eta\gamma\Delta(t) v \quad (t). \tag{140}
$$

Given $\Delta(t)$, these can be expressed as linear systems of equations. Now, we can easily compute the uncoupled kernel variance

$$\begin{aligned}\kappa(t,s) &= 2\left\langle h(t)h(s)\right\rangle^2 + 2\left\langle z(t)z(s)\right\rangle^2 + 2\left\langle h(t)z(s)\right\rangle^2 + 2\left\langle z(t)h(s)\right\rangle^2 \\ &= \left\langle v_+(t)v_+(s) + v\ \ (t)v\ \ (s)\right\rangle^2 + \left\langle v_+(t)v_+(s) - v\ \ (t)v\ \ (s)\right\rangle^2 . \end{aligned} \tag{141}$$

Similarly, we can calculate $D(t,s)$ by using the fact $h(t)^2 + z(t)^2 = v_+(t)^2 + v\ \ (t)^2$

$$\begin{aligned} D(t,s) &= 2\quad v_+(t)\frac{\partial v_+(t)}{\partial\Delta(s)}\quad + 2\quad v\ \ (t)\frac{\partial v\ \ (t)}{\partial\Delta(s)} \\ \frac{\partial v_+(t)}{\partial\Delta(s)} &= \gamma\Theta(t-s)v_+(s) + \sum_{t^0<t}\Delta(t^0)\frac{\partial v_+(t^0)}{\partial\Delta(s)} \\ \frac{\partial v\ \ (t)}{\partial\Delta(s)} &= -\gamma\Theta(t-s)v\ \ (s) - \sum_{t^0<t}\Delta(t^0)\frac{\partial v\ \ (t^0)}{\partial\Delta(s)} \end{aligned} \tag{142}$$

These can be directly solved as a linear system of equations.

N Computing Details

Experiments for Figures 3, 6 and 2 were conducted on a Google Colab GPU with JAX. Experiments for Figures 5, A.3, 7 were performed on a NVIDIA SMX4-A100-80GB GPU. The total compute required for all Figures in the paper took around 4 hours. Jupyter Notebooks to reproduce plots can be found at `https://github.com/Pehlevan-Group/dmft_fluctuations`.