Self-Consistent Dynamical Field Theory of Kernel Evolution in Wide Neural Networks

Anonymous Author(s)
Affiliation
Address
email

Abstract

We analyze feature learning in infinite width neural networks trained with gradient flow through a self-consistent dynamical field theory. We construct a collection of deterministic dynamical order parameters which are inner-product kernels for hidden unit activations and gradients in each layer at pairs of time points, providing a reduced description of network activity through training. These kernel order parameters collectively define the hidden layer activation distribution, the evolution of the neural tangent kernel, and consequently output predictions. For deep linear networks, these kernels satisfy a set of algebraic matrix equations. For nonlinear networks, we provide an alternating sampling procedure to self-consistently solve for the kernel order parameters. We provide comparisons of the self-consistent solution to various approximation schemes including the static NTK approximation, gradient independence assumption, and leading order perturbation theory, showing that each of these approximations can break down in regimes where general self-consistent solutions still provide an accurate description. Lastly, we provide experiments in more realistic settings which demonstrate that the loss and kernel dynamics of CNNs at fixed feature learning strength is preserved across different widths on a CIFAR classification task.

1 Introduction

Deep learning has emerged as a successful paradigm for solving challenging machine learning and computational problems across a variety of domains [1, 2]. However, theoretical understanding of the training and generalization of modern deep learning methods lags behind current practice. Ideally, a theory of deep learning would be analytically tractable, efficiently computable, capable of predicting network performance and internal features that the network learns, and interpretable through a reduced description involving desirably initialization-independent quantities.

Several recent theoretical advances have fruitfully considered the idealization of *wide neural networks*, where the number of hidden units in each layer is taken to be large. Under certain parameterization, Bayesian neural networks and gradient descent trained networks converge to gaussian processes (NNGPs) [3, 4] and neural tangent kernel (NTK) machines [5-7] in their respective infinite width limits. These limits provide both analytic tractability as well as detailed training and generalization analysis [8–15]. However, in this limit, with these parameterizations, data representations are fixed and do not adapt to data, termed the *lazy regime* of NN training, to contrast it from the *rich regime* where NNs significantly alter their internal features while fitting the data [16, 17]. The fact that the representation of data is fixed renders these kernel-based theories incapable of explaining feature learning, an ingredient which is crucial to the success of deep learning in practice [18, 19]. Thus, alternative theories capable of modeling feature learning dynamics are needed.

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

In this work, we attempt to take a step closer towards an ideal theory by deriving an exact analytical description of *feature learning* NNs at infinite width, but parameterized differently [20–22], in terms of a collection of deterministic dynamical feature and gradient kernels. We show that these kernels fully determine the distribution of any network observable. Our results thus provide a bridge between the kernel-centric philosophy of the lazy limit and the rich regime of feature learning, and are predictive of the dynamics of wide but finite networks in the feature learning regime.

Our contributions in this paper are the following:

1. We develop a path integral formulation of gradient flow dynamics in infinite width networks in the feature learning regime. Our parameterization allows interpolation between rich and lazy regimes.
2. From this path integral formulation, we identify a set of deterministic order parameters, which are feature and gradient kernels at each layer. We show that these order parameters are sufficient to define the distribution of hidden activations at any time of network training.
3. We identify a set of self-consistency criteria that the kernels satisfy at infinite width which relate these stochastic processes to the kernels and vice versa. For deep linear networks, the self-consistency conditions form a closed set of algebraic matrix equations. For nonlinear networks, we provide a numerical procedure to solve the field theory self-consistently.
4. In numerical experiments, we demonstrate that solutions to these self-consistency equations are predictive of network training at a variety of feature learning strengths, widths and depths. We provide comparisons of our theory to various approximate methods, such as perturbation theory.

1.1 Related Works

A natural extension to the lazy NTK/NNGP limit that allows the study of feature learning is to calculate finite width corrections to the infinite width limit. Finite width corrections to Bayesian inference in wide networks have been obtained with various perturbative [23–27] and self-consistent techniques [28–31]. In the gradient descent based setting, leading order corrections to the NTK dynamics have been analyzed to study finite width effects [32–34, 26]. These methods give approximate corrections which are accurate provided the strength of feature learning is small. In very rich feature learning regimes, however, the corrections can give incorrect predictions [35, 36].

Another approach to study feature learning is to alter NN parameterization in gradient-based learning to allow significant feature evolution even at infinite width, the *mean field* limit [20, 37]. Works on mean field NNs have yielded formal loss convergence results [38, 39] and shown equivalences of gradient flow dynamics to a partial differential equation (PDE) [40–42], however computation of the PDE for deep networks is often computationally expensive compared to the kernel limit. In a related approach, a set of recent works have demonstrated equivalence between one-pass stochastic gradient descent with mean field parameterization and a hierarchical stochastic process which can be computed efficiently [22, 43], enabling theoretical solutions to practical issues such as hyper-parameter search and transfer [44]. This stochastic process gave a simplification of training dynamics, but was not further reduced to a description involving small number of initialization independent kernels. Further, in the one pass setting, each sample and its representation is seen only once rather than computed throughout training, prohibiting tracking the full kernel through time.

Our theory is inspired by self-consistent dynamical mean field theory (DMFT) from statistical physics [45–51]. This framework has been utilized in the theory of random recurrent networks [52–56], tensor PCA [57, 58], phase retrieval [59], and high-dimensional linear classifiers [60–63], but has yet to be developed for deep feature learning. By developing a self-consistent DMFT of deep NNs, we gain insight into how features evolve in the rich regime of network training, while retaining many pleasant analytic properties of the infinite width limit.

2 Problem Setup and Definitions

Our theory applies to infinite width networks, both fully-connected and convolutional. For notational ease we will relegate convolutional results to later sections. For input $\boldsymbol{x}_\mu \in \mathbb{R}^D$, we define the hidden *pre-activation* vectors $\boldsymbol{h}^\ell \in \mathbb{R}^N$ for layers $\ell \in \{1, ..., L\}$ as

$$f_\mu = \frac{1}{\gamma\sqrt{N}} \boldsymbol{w}^L \cdot \phi(\boldsymbol{h}^L_\mu) \,, \quad \boldsymbol{h}^{\ell+1}_\mu = \frac{1}{\sqrt{N}} \boldsymbol{W}^\ell \phi(\boldsymbol{h}^\ell_\mu) \,, \quad \boldsymbol{h}^1_\mu = \frac{1}{\sqrt{D}} \boldsymbol{W}^0 \boldsymbol{x}_\mu, \tag{1}$$

where $\boldsymbol{\theta} = \text{Vec}\{\boldsymbol{W}^0, ..., \boldsymbol{w}^L\}$ are the trainable parameters of the network and ϕ is a twice differentiable activation function. Inspired by previous works on the mechanisms of lazy gradient based training, the parameter γ will control the laziness or richness of the training dynamics [16, 17, 22, 40]. Each of the trainable parameters are initialized as Gaussian random variables with unit variance $W_{ij}^\ell \sim \mathcal{N}(0,1)$. They evolve under gradient flow $\frac{d}{dt}\boldsymbol{\theta} = -\gamma^2 \nabla_{\boldsymbol{\theta}} \mathcal{L}$. The choice of learning rate γ^2 causes $\frac{d}{dt}\mathcal{L}|_{t=0}$ to be independent of γ. To characterize the evolution of weights, we introduce backpropagation variables $\boldsymbol{g}_\mu^\ell = \gamma\sqrt{N}\frac{\partial f_\mu}{\partial \boldsymbol{h}_\mu^\ell} = \dot{\phi}(\boldsymbol{h}_\mu^\ell) \odot \boldsymbol{z}_\mu^\ell$, where $\boldsymbol{z}_\mu^\ell = \frac{1}{\sqrt{N}}\boldsymbol{W}^{\ell\top}\boldsymbol{g}_\mu^{\ell+1}$ is the *pre-gradient* signal.

The relevant dynamical objects to characterize feature learning are feature and gradient kernels for each hidden layer $\ell \in \{1, ..., L\}$, defined as

$$\Phi_{\mu\alpha}^\ell(t,s) = \frac{1}{N}\phi(\boldsymbol{h}_\mu^\ell(t)) \cdot \phi(\boldsymbol{h}_\nu^\ell(s)) \ , \quad G_{\mu\alpha}^\ell(t,s) = \frac{1}{N}\boldsymbol{g}_\mu^\ell(t) \cdot \boldsymbol{g}_\alpha^\ell(s). \tag{2}$$

From the kernels $\{\Phi^\ell, G^\ell\}_{\ell=1}^L$, we can compute the *Neural Tangent Kernel* $K_{\mu\alpha}^{NTK}(t,s) = \nabla_\theta f_\mu(t) \cdot \nabla_\theta f_\alpha(s) = \sum_{\ell=0}^L G_{\mu\alpha}^{\ell+1}(t,s)\Phi_{\mu\alpha}^\ell(t,s)$, [5] and the dynamics of the network function f_μ

$$\frac{d}{dt}f_\mu(t) = \sum_{\alpha=1}^P K_{\mu\alpha}^{NTK}(t,t)\Delta_\alpha(t) \ , \quad \Delta_\mu(t) = -\frac{\partial}{\partial f_\mu}\mathcal{L}|_{f_\mu(t)}, \tag{3}$$

where we define base cases $G_{\mu\alpha}^{L+1}(t,s) = 1, \Phi_{\mu\alpha}^0(t,s) = K_{\mu\alpha}^x = \frac{1}{D}\boldsymbol{x}_\mu \cdot \boldsymbol{x}_\alpha$. We note that the above formula holds for any data point μ which may or may not be in the set of P training examples. The above expressions demonstrate that knowledge of the temporal trajectory of the NTK on the $t = s$ diagonal gives the temporal trajectory of the network predictions $f_\mu(t)$.

Following prior works on infinite width networks [20, 22, 38, 17], we study the mean field limit

$$N, \gamma \to \infty \ , \quad \gamma_0 = \frac{\gamma}{\sqrt{N}} = \mathcal{O}_N(1). \tag{4}$$

As we demonstrate in the Appendix D and L, this is the only scaling which allows feature learning as $N \to \infty$. The $\gamma_0 = 0$ limit recovers the static NTK limit [5]. We discuss other scalings and parameterizations in Appendix L, relating our work to the μP-parameterization of [22] and showing these parameterizations give identical feature dynamics in the infinite width limit. We also analyze the effect of different hidden layer widths and initialization variances in the Appendix D.7. We focus on equal widths and NTK parameterization (as in eq. (1)) in the main text to reduce complexity.

3 Self-consistent DMFT

Next, we derive our self-consistent DMFT. Our goal is to build a description of training dynamics purely based on representations, and independent of weights. Studying feature learning at infinite width enjoys several analytical properties:

- The kernel order parameters Φ^ℓ, G^ℓ concentrate over random initializations but are dynamical, allowing flexible adaptation of features to the task structure.
- In each layer ℓ, each neuron's preactivation h_i^ℓ and pregradient z_i^ℓ become i.i.d. draws from a distribution characterized by a set of order parameters $\{\Phi^\ell, G^\ell, A^\ell, B^\ell\}$.
- The kernels are defined as self-consistent averages (denoted by $\langle\rangle$) over this distribution of neurons in each layer $\Phi_{\mu\alpha}^\ell(t,s) = \left\langle \phi(h_\mu^\ell(t))\phi(h_\alpha^\ell(s)) \right\rangle$ and $G_{\mu\alpha}^\ell(t,s) = \left\langle g_\mu^\ell(t) g_\alpha^\ell(s) \right\rangle$.

The next section derives these facts from a path-integral formulation of gradient flow dynamics.

3.1 Path Integral Construction

Gradient flow after a random initialization of weights defines a high dimensional stochastic process over initalizations for variables $\{\boldsymbol{h}, \boldsymbol{g}\}$. Therefore, we will utilize DMFT formalism to obtain a reduced description of network activity during training. We separate the contribution on each forward/backward pass between the initial condition and gradient updates to weight matrix $\boldsymbol{W}^\ell$, defining new stochastic variables $\boldsymbol{\chi}^\ell, \boldsymbol{\xi}^\ell \in \mathbb{R}^N$ as

$$\boldsymbol{\chi}_\mu^{\ell+1}(t) = \frac{1}{\sqrt{N}}\boldsymbol{W}^\ell(0)\phi(\boldsymbol{h}_\mu^\ell(t)) \ , \quad \boldsymbol{\xi}_\mu^\ell(t) = \frac{1}{\sqrt{N}}\boldsymbol{W}^\ell(0)^\top \boldsymbol{g}_\mu^{\ell+1}(t). \tag{5}$$

We let Z represent the moment generating functional (MGF) for these stochastic fields

$$Z[\{\boldsymbol{j}^\ell, \boldsymbol{v}^\ell\}] = \left\langle \exp\left(\sum_{\ell,\mu} \int_0^\infty dt \left[\boldsymbol{j}_\mu^\ell(t) \cdot \boldsymbol{\chi}_\mu^\ell(t) + \boldsymbol{v}_\mu^\ell(t) \cdot \boldsymbol{\xi}_\mu^\ell(t) \right] \right) \right\rangle_{\{\boldsymbol{W}^0(0), \ldots \boldsymbol{w}^L(0)\}},$$

which requires, by construction the normalization condition $Z[\{\mathbf{0}, \mathbf{0}\}] = 1$. We enforce our definition of $\boldsymbol{\chi}, \boldsymbol{\xi}$ using an integral representation of the delta-function. Thus for each sample $\mu \in [P]$ and each time $t \in \mathbb{R}_+$, we multiply Z by

$$1 = \int_{\mathbb{R}^N} \int_{\mathbb{R}^N} \frac{d\boldsymbol{\chi}_\mu^{\ell+1}(t) d\hat{\boldsymbol{\chi}}_\mu^{\ell+1}(t)}{(2\pi)^N} \exp\left(i\hat{\boldsymbol{\chi}}_\mu^{\ell+1}(t) \cdot \left[\boldsymbol{\chi}_\mu^{\ell+1}(t) - \frac{1}{\sqrt{N}} \boldsymbol{W}^\ell(0) \phi(\boldsymbol{h}_\mu^\ell(t)) \right] \right), \tag{6}$$

for $\boldsymbol{\chi}$ and the respective expression for $\boldsymbol{\xi}$. After making such substitutions, we perform integration over initial Gaussian weight matrices to arrive at an integral expression for Z, which we derive in the appendix D.3. We show that Z can be described by set of order-parameters $\{\Phi^\ell, \hat{\Phi}^\ell, G^\ell, \hat{G}^\ell, A^\ell, B^\ell\}$

$$Z[\{\boldsymbol{j}^\ell, \boldsymbol{v}^\ell\}] \propto \int \prod_{\ell\mu\alpha ts} d\Phi_{\mu\alpha}^\ell(t,s) d\hat{\Phi}_{\mu\alpha}^\ell(t,s) dG_{\mu\alpha}^\ell(t,s) d\hat{G}_{\mu\alpha}^\ell(t,s) dA_{\mu\alpha}^\ell(t,s) dB_{\mu\alpha}^\ell(t,s) \tag{7}$$

$$\times \exp\left(NS[\{\Phi, \hat{\Phi}, G, \hat{G}, A, B, j, v\}] \right),$$

$$S = \sum_{\ell\mu\alpha} \int_0^\infty dt \int_0^\infty ds \left[\Phi_{\mu\alpha}^\ell(t,s) \hat{\Phi}_{\mu\alpha}^\ell(t,s) + G_{\mu\alpha}^\ell(t,s) \hat{G}_{\mu\alpha}^\ell(t,s) - A_{\mu\alpha}^\ell(t,s) B_{\mu\alpha}^\ell(t,s) \right]$$

$$+ \ln \mathcal{Z}[\{\Phi, \hat{\Phi}, G, \hat{G}, A, B, j, v\}], \tag{8}$$

where $\mathcal{Z}$ is a single-site MGF, which defines the distribution of fields $\{\chi^\ell, \xi^\ell\}$ over the neural population in each layer. The kernels A and B are related to the correlations between feedforward and feedback signals in the network. We provide a detailed formula for $\mathcal{Z}$ in the Appendix D.3 and show that it factorizes over different layers $\mathcal{Z} = \prod_{\ell=1}^L \mathcal{Z}_\ell$.

3.2 Deriving the DMFT Equations from the Path Integral Saddle Point

As $N \to \infty$, the moment-generating function Z is exponentially dominated by the saddle point of S. The equations that define this saddle point also define our DMFT. We thus identify the kernels that render S locally stationary. The most important equations are those which define $\{\Phi^\ell, G^\ell\}$

$$\frac{\delta S}{\delta \hat{\Phi}_{\mu\alpha}^\ell(t,s)} = \Phi_{\mu\alpha}^\ell(t,s) + \frac{1}{\mathcal{Z}} \frac{\delta \mathcal{Z}}{\delta \hat{\Phi}_{\mu\alpha}^\ell(t,s)} = \Phi_{\mu\alpha}^\ell(t,s) - \left\langle \phi(h_\mu^\ell(t)) \phi(h_\alpha^\ell(s)) \right\rangle = 0,$$

$$\frac{\delta S}{\delta \hat{G}_{\mu\alpha}^\ell(t,s)} = G_{\mu\alpha}^\ell(t,s) + \frac{1}{\mathcal{Z}} \frac{\delta \mathcal{Z}}{\delta \hat{G}_{\mu\alpha}^\ell(t,s)} = G_{\mu\alpha}^\ell(t,s) - \left\langle g_\mu^\ell(t) g_\alpha^\ell(s) \right\rangle = 0, \tag{9}$$

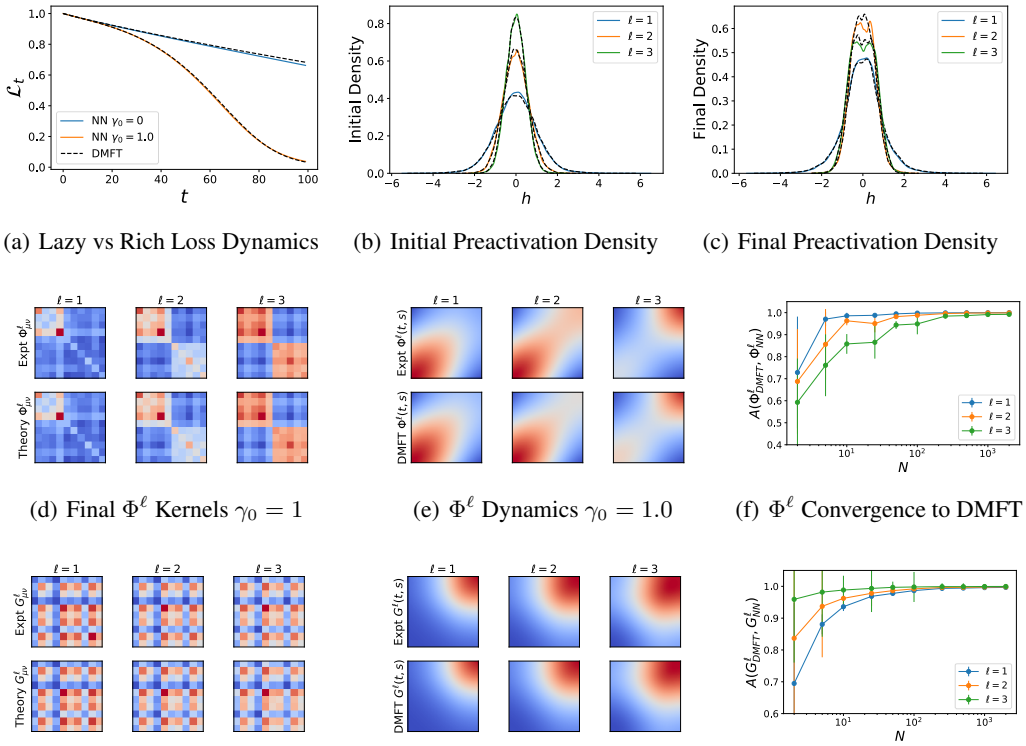
where $\langle\rangle$ denotes an average over the stochastic process induced by $\mathcal{Z}$, which is defined below

$$\{u_\mu^\ell(t)\}_{\mu\in[P], t\in\mathbb{R}_+} \sim \mathcal{GP}(0, \boldsymbol{\Phi}^{\ell-1}), \ \{r_\mu^\ell(t)\}_{\mu\in[P], t\in\mathbb{R}_+} \sim \mathcal{GP}(0, \boldsymbol{G}^{\ell+1}),$$

$$h_\mu^\ell(t) = u_\mu^\ell(t) + \gamma_0 \int_0^\infty ds \sum_{\alpha=1}^P \left[A_{\mu\alpha}^{\ell-1}(t,s) + \Theta(t-s) \Delta_\alpha(s) \Phi_{\mu\alpha}^{\ell-1}(t,s) \right] z_\alpha^\ell(s) \dot{\phi}(h_\alpha^\ell(s)),$$

$$z_\mu^\ell(t) = r_\mu^\ell(t) + \gamma_0 \int_0^\infty ds \sum_{\alpha=1}^P \left[B_{\mu\alpha}^\ell(t,s) + \Theta(t-s) \Delta_\alpha(s) G_{\mu\alpha}^{\ell+1}(t,s) \right] \phi(h_\alpha^\ell(s)), \tag{10}$$

where we define base cases $\Phi_{\mu\alpha}^0(t,s) = K_{\mu\alpha}^x$ and $G_{\mu\alpha}^{L+1}(t,s) = 1$, $A^0 = B^L = 0$. We see that the fields $\{h^\ell, z^\ell\}$, which represent the single site preactivations and pre-gradients, are implicit functionals of the mean-zero Gaussian processes $\{u^\ell, r^\ell\}$ which have covariances $\left\langle u_\mu^\ell(t) u_\alpha^\ell(s) \right\rangle = \Phi_{\mu\alpha}^{\ell-1}(t,s)$ and $\left\langle r_\mu^\ell(t) r_\alpha^\ell(s) \right\rangle = G_{\mu\alpha}^{\ell+1}(t,s)$. The other saddle point equations give $A_{\mu\alpha}^\ell(t,s) = \gamma_0^{-1} \left\langle \frac{\delta \phi(h_\mu^\ell(t))}{\delta r_\alpha^\ell(s)} \right\rangle$, $B_{\mu\alpha}^\ell(t,s) = \gamma_0^{-1} \left\langle \frac{\delta g_\mu^{\ell+1}(t)}{\delta u_\alpha^{\ell+1}(s)} \right\rangle$ which arise due to coupling between



(a) Lazy vs Rich Loss Dynamics (b) Initial Preactivation Density (c) Final Preactivation Density

(d) Final Φ^ℓ Kernels $\gamma_0 = 1$ (e) Φ^ℓ Dynamics $\gamma_0 = 1.0$ (f) Φ^ℓ Convergence to DMFT

(g) Final G^ℓ kernels $\gamma_0 = 1.0$ (h) G^ℓ Dynamics $\gamma_0 = 1.0$ (i) G^ℓ Convergence to DMFT

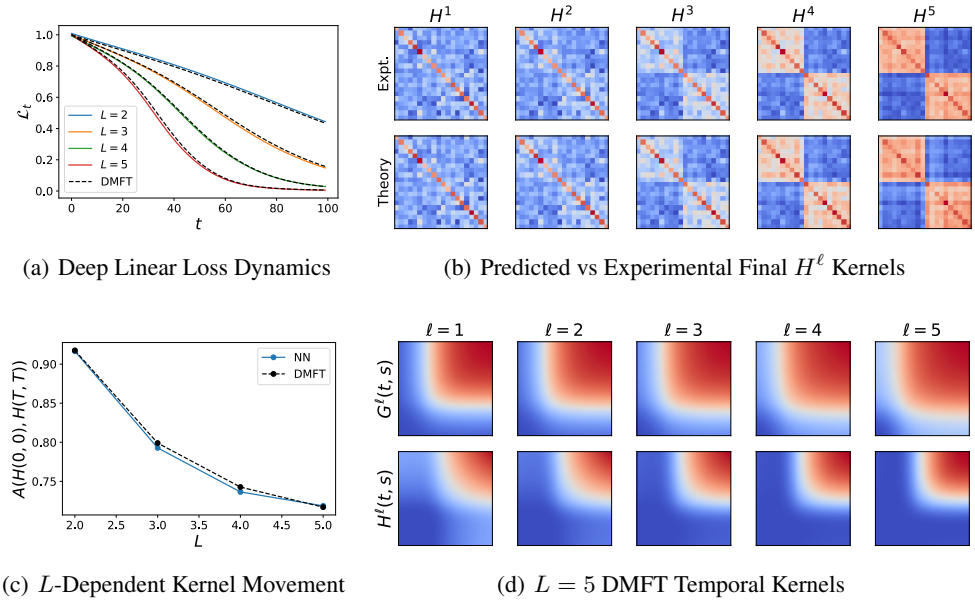
Figure 1: Neural network feature learning dynamics is captured by self-consistent dynamical mean field theory (DMFT). (a) Training loss curves on a subsample of $P = 10$ CIFAR-10 training points in a depth 4 ($L = 3$, $N = 2500$) tanh network ($\phi(h) = \tanh(h)$) trained with MSE. Increasing γ_0 accelerates training. (b)-(c) The distribution of preactivations at the beginning and end of training matches predictions of the DMFT. (d) The final Φ^ℓ (at $t = 100$) kernel order parameters match the finite width network. (e) The temporal dynamics of the sample-traced kernels $\sum_\mu \Phi^\ell_{\mu\mu}(t, s)$ matches experiment and reveals rich dynamics across layers. (f) The alignment $A(\boldsymbol{\Phi}^\ell_{DMFT}, \boldsymbol{\Phi}^\ell_{NN})$, defined as cosine similarity, of the kernel $\Phi^\ell_{\mu\alpha}(t, s)$ predicted by theory (DMFT) and width N networks for different N but fixed $\gamma_0 = \gamma/\sqrt{N}$. Errorbars show standard deviation computed over 10 repeats. Around $N \sim 500$ DMFT begins to show near perfect agreement with the NN. (g)-(i) The same plots but for the gradient kernel $\boldsymbol{G}^\ell$. Whereas finite width effects for $\boldsymbol{\Phi}^\ell$ are larger at later layers ℓ since variance accumulates on the forward pass, fluctuations in $\boldsymbol{G}^\ell$ are large in early layers.

the feedforward and feedback signals. We note that, in the lazy limit $\gamma_0 \to 0$, the fields approach Gaussian processes $h^\ell \to u^\ell$, $z^\ell \to r^\ell$. Lastly, the final saddle point equations $\frac{\delta S}{\delta \Phi^\ell} = 0$, $\frac{\delta S}{\delta G^\ell} = 0$ imply that $\hat{\Phi}^\ell = \hat{G}^\ell = 0$. The full set of equations that define the DMFT are given in D.6.

This theory is easily extended to more general architectures such as networks with varying widths by layer (App. D.7), trainable bias parameter (App. H), multiple (but $\mathcal{O}_N(1)$) output channels (App. I), convolutional architectures (App. G), networks trained with momentum (App. J), discrete time training (App. K), and alternative parameterization schemes (App. L), showing our setup is equivalent to the μP scheme of [22, 43]. Though prior μP analyses focus on one-pass training, our field theory accommodates batch training on P examples to capture how kernels evolve in time.

4 Solving the Self-Consistent DMFT

The saddle point equations obtained from the field theory discussed in the previous section must be solved self-consistently. By this we mean that, given knowledge of the kernels, we can characterize the



(a) Deep Linear Loss Dynamics

(b) Predicted vs Experimental Final H^ℓ Kernels

(c) L-Dependent Kernel Movement

(d) $L = 5$ DMFT Temporal Kernels

Figure 2: Deep linear network with the full DMFT. (a) The train loss for NNs of varying L. (b) For a $L = 5, N = 1000$ NN, the kernels H^ℓ at the end of training compared to DMFT theory on $P = 20$ datapoints. (c) The average displacement of feature kernels for different depth networks at same γ_0 value. For equal values of γ_0, deeper networks exhibit larger changes to their features, manifested in lower alignment with their initial $t = 0$ kernels $\boldsymbol{H}$. (d) The solution to the temporal components of the $G^\ell(t, s)$ and $\sum_\mu H^\ell_{\mu\mu}(t, s)$ kernels obtained from the self-consistent equations.

distribution of $\{h^\ell, z^\ell\}$, and given the distribution of $\{h^\ell, z^\ell\}$, we can compute the kernels [64, 61]. In the Appendix B, we provide Algorithm 1, a numerical procedure based on this idea to efficiently solve for the kernels with an alternating Monte-Carlo strategy. The output of the algorithm are the dynamical kernels $\Phi^\ell_{\mu\alpha}(t, s), G^\ell_{\mu\alpha}(t, s), A^\ell_{\mu\alpha}(t, s), B^\ell_{\mu\alpha}(t, s)$, from which any network observable can be computed as we discuss in Appendix D. We provide an example of the solution to the saddle point equations compared to training a finite NN in Figure 1. We plot Φ^ℓ, G^ℓ at the end of training and the sample-trace of these kernels through time. Additionally, we compare the kernels of finite width N network to the DMFT predicted kernels using a cosine-similarity alignment metric $A(\boldsymbol{\Phi}^{DMFT}, \boldsymbol{\Phi}^{NN}) = \frac{\text{Tr}\, \boldsymbol{\Phi}^{DMFT} \boldsymbol{\Phi}^{NN}}{|\boldsymbol{\Phi}^{DMFT}||\boldsymbol{\Phi}^{NN}|}$. Additional examples are in Appendix Figures 5 and Figure 6.

4.1 Deep Linear Networks: Closed Form Self-Consistent Equations

Deep linear networks ($\phi(h) = h$) are of theoretical interest since they are simpler to analyze than nonlinear networks but preserve non-trivial training dynamics and feature learning [65-69, 25, 30, 23]. In a deep linear network, we can simplify our saddle point equations to algebraic formulas that close in terms of the kernels $H^\ell_{\mu\alpha}(t, s) = \langle h^\ell_\mu(t) h^\ell_\alpha(s) \rangle$, $G^\ell(t, s) = \langle g^\ell(t) g^\ell(s) \rangle$. This is a significant simplification since it allows solution of the saddle point equations without a sampling procedure.

To describe the result, we first introduce a vectorization notation $\boldsymbol{h}^\ell = \text{Vec}\{h^\ell_\mu(t)\}_{\mu \in [P], t \in \mathbb{R}_+}$. Likewise we convert kernels $\boldsymbol{H}^\ell = \text{Mat}\{H^\ell_{\mu\alpha}(t, s)\}_{\mu,\alpha \in [P], t,s \in \mathbb{R}_+}$ into matrices. The inner product under this vectorization is defined as $\boldsymbol{a} \cdot \boldsymbol{b} = \int_0^\infty dt \sum_{\mu=1}^P a_\mu(t) b_\mu(t)$. In a practical computational implementation, the theory would be evaluated on a grid of T time points with discrete time gradient descent, so these kernels $\boldsymbol{H}^\ell \in \mathbb{R}^{PT \times PT}$ would indeed be matrices of the appropriate size. We can write the following algebraic expressions for fields $\boldsymbol{h}^\ell, \boldsymbol{g}^\ell$ in terms of independent Gaussian processes $\boldsymbol{u}^\ell, \boldsymbol{r}^\ell$ which have covariances $\boldsymbol{H}^{\ell-1}$ and $\boldsymbol{G}^{\ell+1}$ respectively

$$(\mathbf{I} - \gamma_0^2 \boldsymbol{C}^\ell \boldsymbol{D}^\ell) \boldsymbol{h}^\ell = \boldsymbol{u}^\ell + \gamma_0 \boldsymbol{C}^\ell \boldsymbol{r}^\ell \ , \ (\mathbf{I} - \gamma_0^2 \boldsymbol{D}^\ell \boldsymbol{C}^\ell) \boldsymbol{g}^\ell = \boldsymbol{r}^\ell + \gamma_0 \boldsymbol{D}^\ell \boldsymbol{u}^\ell. \tag{11}$$

The matrices $\boldsymbol{C}^\ell$ and $\boldsymbol{D}^\ell$ are causal integral operators which depend on $\{\boldsymbol{A}^{\ell-1}, \boldsymbol{H}^{\ell-1}\}$ and $\{\boldsymbol{B}^\ell, \boldsymbol{G}^{\ell+1}\}$ respectively which we define in Appendix F. We see that the $\boldsymbol{h}, \boldsymbol{g}$ fields are Gaussian in the linear network, not just at initialization, but throughout training. The saddle point equations which define the kernels in terms of two point correlators are

$$\boldsymbol{H}^\ell = \left\langle \boldsymbol{h}^\ell \boldsymbol{h}^{\ell\top} \right\rangle = (\mathbf{I} - \gamma_0^2 \boldsymbol{C}^\ell \boldsymbol{D}^\ell)^{-1} [\boldsymbol{H}^{\ell-1} + \gamma_0^2 \boldsymbol{C}^\ell \boldsymbol{G}^{\ell+1} \boldsymbol{C}^{\ell\top}] \left[(\mathbf{I} - \gamma_0^2 \boldsymbol{C}^\ell \boldsymbol{D}^\ell)^{-1}\right]^\top$$
$$\boldsymbol{G}^\ell = \left\langle \boldsymbol{g}^\ell \boldsymbol{g}^{\ell\top} \right\rangle = \left(\mathbf{I} - \gamma_0^2 \boldsymbol{D}^\ell \boldsymbol{C}^\ell\right)^{-1} \left[\boldsymbol{G}^{\ell+1} + \gamma_0^2 \boldsymbol{D}^\ell \boldsymbol{H}^{\ell-1} \boldsymbol{D}^{\ell\top}\right] \left[\left(\mathbf{I} - \gamma_0^2 \boldsymbol{D}^\ell \boldsymbol{C}^\ell\right)^{-1}\right]^\top . \quad (12)$$

Examples of the predictions obtained by solving these systems of equations are provided in Figure 2. We see that these DMFT equations describe kernel evolution for networks of a variety of depths and that the change in each layer's kernel increases with the depth of the network. We note the inverse $\left(\mathbf{I} - \gamma_0^2 \boldsymbol{C}^\ell \boldsymbol{D}^\ell\right)^{-1}$, when viewed as a function of γ_0^2 has simple poles at the reciprocal singular values of $\boldsymbol{C}^\ell \boldsymbol{D}^\ell$, which suggests the existence of a maximal stable γ_0. In experiments we have observed kernels diverging for sufficiently large γ_0, but leave in-depth analysis of this to future work.

Unlike many prior results [65–68], our DMFT does not require any restrictions on the structure of the input data but hold for any $\boldsymbol{K}^x, \boldsymbol{y}$. However, for whitened data $\boldsymbol{K}^x = \mathbf{I}$ we show in Appendix F.1.1 that our DMFT learning curves interpolate between NTK dynamics and the sigmoidal trajectories of prior works [65, 66] as γ_0 is increased. For example, in the two layer ($L = 1$) linear network with $\boldsymbol{K}^x = \mathbf{I}$, the dynamics of the error norm $\Delta(t) = ||\boldsymbol{\Delta}(t)||$ takes the form $\frac{\partial}{\partial t}\Delta(t) = -\sqrt{1 + \gamma_0^2 (y - \Delta(t))^2}\Delta(t)$ where $y = ||\boldsymbol{y}||$. These dynamics give the linear convergence rate of the NTK if $\gamma_0 \to 0$ but approaches logistic dynamics of [66] as $\gamma_0 \to \infty$. Further, $\boldsymbol{H}(t) = \left\langle \boldsymbol{h}^1(t) \boldsymbol{h}^1(t)^\top \right\rangle \in \mathbb{R}^{P \times P}$ only grows in the $\boldsymbol{y}\boldsymbol{y}^\top$ direction with $H_y(t) = \frac{1}{y^2} \boldsymbol{y}^\top \boldsymbol{H}(t) \boldsymbol{y} = \sqrt{1 + \gamma_0^2 (y - \Delta(t))^2}$. At the end of training $\boldsymbol{H}(t) \to \mathbf{I} + \frac{1}{y^2}[\sqrt{1 + \gamma_0^2 y^2} - 1]\boldsymbol{y}\boldsymbol{y}^\top$, recovering the rank one spike which was recently obtained in the small initialization limit [**?**].

5 Approximation Schemes

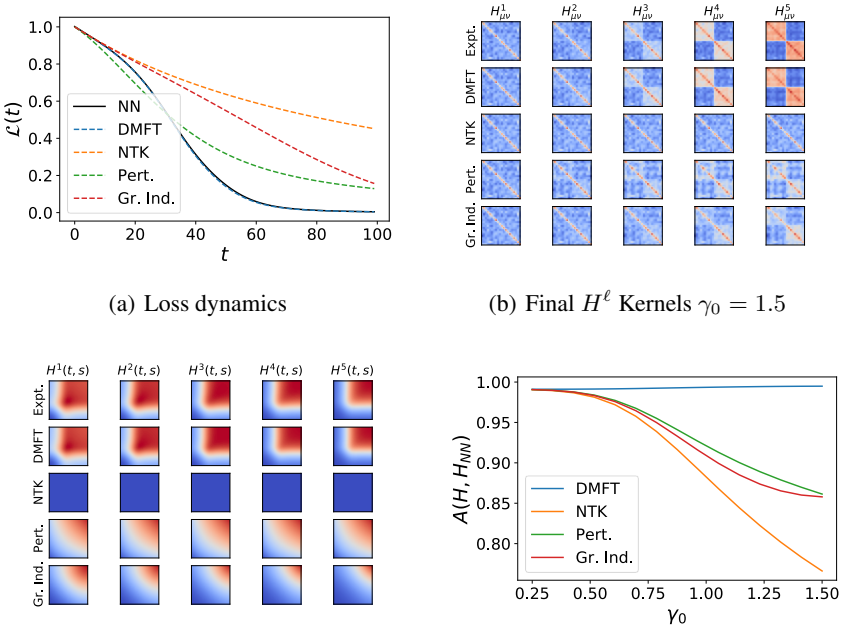
We now compare our exact DMFT with approximations of prior works, providing an explanation of when these approximations give accurate predictions and when they break down.

5.1 Gradient Independence Ansatz

We can study the accuracy of the ansatz $\boldsymbol{A}^\ell = \boldsymbol{B}^\ell = 0$, which is equivalent to treating the weight matrices $\boldsymbol{W}^\ell(0)$ and $\boldsymbol{W}^\ell(0)^\top$ which appear in forward and backward passes respectively as independent Gaussian matrices. This assumption was utilized in prior works on signal propagation in deep networks in the lazy regime [70–74]. A consequence of this approximation is the Gaussianity and statistical independence of χ^ℓ and ξ^ℓ (conditional on $\{\boldsymbol{\Phi}^\ell, \boldsymbol{G}^\ell\}$) in each layer as we show in Appendix M. This ansatz works very well near $\gamma_0 \approx 0$ (the static kernel regime) since $\frac{d\boldsymbol{h}}{d\boldsymbol{r}}, \frac{d\boldsymbol{z}}{d\boldsymbol{u}} \sim \mathcal{O}(\gamma_0)$ or around initialization $t \approx 0$ but begins to fail at larger values of γ_0, t (Figure 3).

5.2 Perturbation theory in γ_0 at Infinite Width

In the $\gamma_0 \to 0$ limit, we recover static kernels, giving linear dynamics identical to the NTK limit [5]. Corrections to this lazy limit can be extracted at small but finite γ_0. This is conceptually similar to recent works which consider perturbation series for the NTK in powers of $1/N$ [33, 26, 27] (though not identical, see Appendix N.7). We expand all observables $q(\gamma_0)$ in a power series in γ_0, giving $q(\gamma_0) = q^{(0)} + \gamma_0 q^{(1)} + \gamma_0^2 q^{(2)} + ...$ and compute corrections up to $\mathcal{O}(\gamma_0^2)$. We show that the $\mathcal{O}(\gamma_0)$ and $\mathcal{O}(\gamma_0^3)$ corrections to kernels vanish, giving leading order expansions of the form $\boldsymbol{\Phi} = \boldsymbol{\Phi}^0 + \gamma_0^2 \boldsymbol{\Phi}^2 + \mathcal{O}(\gamma_0^4)$ and $\boldsymbol{G} = \boldsymbol{G}^0 + \gamma_0^2 \boldsymbol{G}^2 + \mathcal{O}(\gamma_0^4)$ (see Appendix N.2). Further, we show that the NTK has relative change at leading order which scales linearly with depth $|\Delta K^{NTK}|/|K^{NTK,0}| \sim \mathcal{O}_{\gamma_0, L}(L\gamma_0^2) = \mathcal{O}_{N,\gamma,L}(\frac{\gamma^2 L}{N})$, which is consistent with finite width effective field theory at $\gamma = \mathcal{O}_N(1)$ [26, 27] (Appendix N.6). Further, at the leading order correction, all temporal dependencies are controlled by $P(P+1)$ functions $v_\alpha(t) = \int_0^t ds \Delta_\alpha^0(s)$ and $v_{\alpha\beta}(t) = \int_0^t ds \Delta_\alpha^0(s) \int_0^s ds' \Delta_\beta^0(s')$, which is consistent with those derived for finite width NNs using a truncation of the Neural Tangent Hierarchy [32, 33, 26]. To lighten notation, we focus our main



(a) Loss dynamics

(b) Final H^ℓ Kernels $\gamma_0 = 1.5$

(c) H^ℓ Kernel Dynamics $\gamma_0 = 1.5$

(d) Theory $\boldsymbol{H}^\ell$ vs NN with $N = 1000$

Figure 3: Comparison of DMFT to various approximation schemes in a $L = 5$ hidden layer, width $N = 1000$ linear network with $\gamma_0 = 1.0$ and $P = 100$. (a) The loss for the various approximations do not track the true trajectory induced by gradient descent in the large γ_0 regime. (b)-(c) The feature kernels $H^\ell_{\mu\alpha}(t, s)$ across each of the $L = 5$ hidden layers for each of the theories is compared to a width 1000 neural network. Again, we plot the sample-traced dynamics $\sum_{\mu\mu} H^\ell_{\mu\mu}(t, s)$. (d) The alignment of $\boldsymbol{H}^\ell$ compared to the finite NN $A(\boldsymbol{H}^\ell, \boldsymbol{H}^\ell_{NN})$ averaged across $\ell \in \{1, ..., 5\}$ for varying γ. The predictions of all of these theories coincide in the $\gamma_0 = 0$ limit but begin to deviate in the feature learning regime. Only the non-perturbative DMFT is accurate over a wide range of γ_0.

text comparison of our non-perturbative DMFT to perturbation theory in the deep linear case. Full perturbation theory is in Appendix N.2.

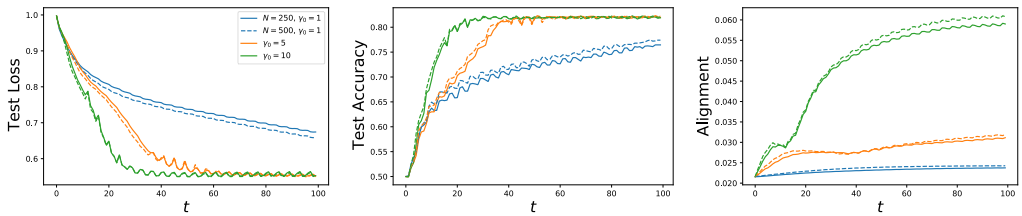
Using the timescales derived in the previous section, we find that the leading order correction to the kernels in infinite width deep linear network have the form

$$K^{NTK}_{\mu\nu}(t, s) = (L + 1)K^x_{\mu\nu} + \gamma_0^2 \frac{L(L + 1)}{2} K^x_{\mu\nu} \sum_{\alpha\beta} K^x_{\alpha\beta}[v_{\alpha\beta}(t) + v_{\beta\alpha}(s) + v_\alpha(t)v_\beta(s)]$$
$$+ \gamma_0^2 \frac{L(L + 1)}{2} \left[\sum_{\alpha\beta} K^x_{\mu\alpha} K^x_{\nu\beta}[v_{\alpha\beta}(t) + v_{\beta\alpha}(s)] + \sum_{\alpha\beta} K^x_{\mu\alpha} K^x_{\nu\beta} v_\alpha(t)v_\beta(s) \right] + \mathcal{O}(\gamma_0^4). \quad (13)$$

We see that the relative change in the NTK $|\boldsymbol{K}^{NTK} - \boldsymbol{K}^{NTK}(0)|/|\boldsymbol{K}^{NTK}(0)| \sim \mathcal{O}(\gamma_0^2 L) = \mathcal{O}(\gamma^2 L/N)$, so that large depth L networks exhibit more significant kernel evolution, which agrees with other perturbative studies [33, 26, 25] as well as the non-perturbative results in Figure 2. However at large γ_0 and large L, this theory begins to break down as we show in Figure 3.

6 Feature Learning Dynamics is Preserved at Fixed γ_0

Our DMFT suggests that for networks sufficiently wide for their kernels to concentrate, the dynamics of loss and kernels should be invariant under the rescaling $N \to RN, \gamma \to \gamma/\sqrt{R}$, which keeps γ_0 fixed. To evaluate how well this idea holds in a realistic deep learning problem, we trained CNNs of varying channel counts N on two-class CIFAR classification [75]. We tracked the dynamics of



(a) Test MSE (b) Classification Error (c) $A(\Phi^L, yy^\top)$ Dynamics

Figure 4: The dynamics of a depth 5 ($L = 4$ hidden) CNNs trained on first two classes of CIFAR (boat vs plane) exhibit consistency for different channel counts $N \in \{250, 500\}$ for fixed $\gamma_0 = \gamma/\sqrt{N}$. (a) We plot the test loss (MSE) and (b) test classification error. Networks with higher γ_0 train more rapidly. Time is measured in every 100 update steps. (c) The dynamics of the last layer feature kernel Φ^L, shown as alignment to the target function. As predicted by the DMFT, higher γ_0 corresponds to more active kernel evolution, evidenced by larger change in the alignment.

the loss and the last layer Φ^L kernel. The results are provided in Figure 4. We see that dynamics are largely independent of rescaling as predicted. Further, as expected, larger γ_0 leads to larger changes in kernel norm and faster alignment to the target function y, as was also found in [76]. Consequently, the higher γ_0 networks train more rapidly. The trend is consistent for width $N = 250$ and $N = 500$. More details about the experiment can be found in Appendix C.2.

7 Discussion

In this work, we provided a unifying DMFT for feature learning in infinite networks trained with gradient based training. This theory smoothly interpolates between lazy infinite width behavior of a static NTK in $\gamma_0 \to 0$ and rich feature learning. At infinite width, each neuron's pre-activation and pre-gradient (or each channel in a CNN) is independent and identically distributed throughout training and the kernels in each layer can be computed from averages over the distribution of neurons. The saddle point equations for kernels are exactly solveable in deep linear networks and can be solved numerically method in the nonlinear case. Experimental comparisons with other approximation methods such as gradient independence and perturbation theory, show that DMFT can be accurate at a much wider range of γ_0. We believe this framework could be a useful starting point for future theoretical analyses of feature learning and generalization in wide networks.

Though our self-consistent DMFT is quite general in regards to the data, architecture and nonlinearity, a limitation is that it is still computationally expensive to evaluate. In Table 1, we compare the time taken for various theories to compute the feature kernels throughout T steps of gradient descent. For a width N network, computation of each forward pass on all P data points takes $\mathcal{O}(PN^2)$ computations. The static NTK requires computation of $\mathcal{O}(P^2)$ entries in the kernel which do not need to be recomputed. However, the DMFT requires matrix multiplications on $PT \times PT$ matrices giving a $\mathcal{O}(P^3T^3)$ time scaling. Future work could aim to improve the computational overhead of the algorithm, by considering data averaged theories [61] or one pass SGD [22].

Requirements	Width-N NN	Static NTK	Perturbative	Full DMFT
Memory for Kernels	$\mathcal{O}(N^2)$	$\mathcal{O}(P^2)$	$\mathcal{O}(P^4T)$	$\mathcal{O}(P^2T^2)$
Time for Kernels	$\mathcal{O}(PN^2T)$	$\mathcal{O}(P^2)$	$\mathcal{O}(P^4T)$	$\mathcal{O}(P^3T^3)$
Time for Final Outputs	$\mathcal{O}(PN^2T)$	$\mathcal{O}(P^3)$	$\mathcal{O}(P^4)$	$\mathcal{O}(P^3T^3)$

Table 1: Computational requirements to compute kernel dynamics and trained network predictions on P points in a depth N neural network on a grid of T time points trained with P data points for various theories. DMFT is faster and less memory intensive than a width N network only if $N \gg PT$. It is more computationally efficient to compute full DMFT kernels than leading order perturbation theory when $T \ll \sqrt{P}$. The expensive scaling with both samples and time are the cost of a full-batch non-perturbative theory of feature learning.

References

[1] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning*. MIT press, 2016.

[2] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

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

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International conference on machine learning*, pages 1675–1685. PMLR, 2019.

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

[10] Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature communications*, 12(1):1–12, 2021.

[11] Omry Cohen, Or Malka, and Zohar Ringel. Learning curves for overparametrized deep neural networks: A field theory perspective. *Physical Review Research*, 3(2):023034, 2021.

[12] Arthur Jacot, Berfin Simsek, Francesco Spadaro, Clément Hongler, and Franck Gabriel. Kernel alignment risk estimator: Risk prediction from training data. *Advances in Neural Information Processing Systems*, 33:15568–15578, 2020.

[13] Bruno Loureiro, Cedric Gerbelot, Hugo Cui, Sebastian Goldt, Florent Krzakala, Marc Mezard, and Lenka Zdeborova. Learning curves of generic features maps for realistic datasets with a teacher-student model. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021.

[14] James B Simon, Madeline Dickens, and Michael R DeWeese. Neural tangent kernel eigenvalues accurately predict generalization. *arXiv preprint arXiv:2110.03922*, 2021.

[15] Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pages 242–252. PMLR, 2019.

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

[17] 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.

[18] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

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

[20] Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018.

[21] Phan-Minh Nguyen and Huy Tuan Pham. A rigorous framework for the mean field limit of multilayer neural networks. *arXiv preprint arXiv:2001.11443*, 2020.

[22] Greg Yang and Edward J Hu. Feature learning in infinite-width neural networks. *arXiv preprint arXiv:2011.14522*, 2020.

[23] Laurence Aitchison. Why bigger is not always better: on finite and infinite neural networks. In *International Conference on Machine Learning*, pages 156–164. PMLR, 2020.

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

[25] 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.

[26] Daniel A Roberts, Sho Yaida, and Boris Hanin. The principles of deep learning theory. *arXiv preprint arXiv:2106.10165*, 2021.

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

[28] 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.

[29] Inbar Seroussi and Zohar Ringel. Separation of scales and a thermodynamic description of feature learning in some cnns. *arXiv preprint arXiv:2112.15383*, 2021.

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

[31] 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.

[32] 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.

[33] Ethan Dyer and Guy Gur-Ari. Asymptotics of wide networks from feynman diagrams. *arXiv preprint arXiv:1909.11304*, 2019.

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

[35] Jacob A Zavatone-Veth, William L Tong, and Cengiz Pehlevan. Contrasting random and learned features in deep bayesian linear regression. *arXiv preprint arXiv:2203.00573*, 2022.

[36] 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.

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

[38] 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.

[39] Grant M Rotskoff and Eric Vanden-Eijnden. Trainability and accuracy of neural networks: An interacting particle system approach. *arXiv preprint arXiv:1805.00915*, 2018.

[40] 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.

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

[42] Cong Fang, Jason Lee, Pengkun Yang, and Tong Zhang. Modeling from features: a mean-field framework for over-parameterized deep neural networks. In *Conference on learning theory*, pages 1887–1936. PMLR, 2021.

[43] Greg Yang, Michael Santacroce, and Edward J Hu. Efficient computation of deep nonlinear infinite-width neural networks that learn features. In *International Conference on Learning Representations*, 2022.

[44] 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.

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

[46] C De Dominicis. Dynamics as a substitute for replicas in systems with quenched random impurities. *Physical Review B*, 18(9):4913, 1978.

[47] Haim Sompolinsky and Annette Zippelius. Dynamic theory of the spin-glass phase. *Physical Review Letters*, 47(5):359, 1981.

[48] Haim Sompolinsky and Annette Zippelius. Relaxational dynamics of the edwards-anderson model and the mean-field theory of spin-glasses. *Physical Review B*, 25(11):6860, 1982.

[49] G Ben Arous and Alice Guionnet. Large deviations for langevin spin glass dynamics. *Probability Theory and Related Fields*, 102(4):455–509, 1995.

[50] G Ben Arous and Alice Guionnet. Symmetric langevin spin glass dynamics. *The Annals of Probability*, 25(3):1367–1422, 1997.

[51] Gérard Ben Arous, Amir Dembo, and Alice Guionnet. Cugliandolo-kurchan equations for dynamics of spin-glasses. *Probability theory and related fields*, 136(4):619–660, 2006.

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

[53] Haim Sompolinsky, Andrea Crisanti, and Hans-Jurgen Sommers. Chaos in random neural networks. *Physical review letters*, 61(3):259, 1988.

[54] Lutz Molgedey, J Schuchhardt, and Heinz G Schuster. Suppressing chaos in neural networks by noise. *Physical review letters*, 69(26):3717, 1992.

[55] M Samuelides and Bruno Cessac. Random recurrent neural networks dynamics. *The European Physical Journal Special Topics*, 142(1):89–122, 2007.

[56] Kanaka Rajan, LF Abbott, and Haim Sompolinsky. Stimulus-dependent suppression of chaos in recurrent neural networks. *Physical review e*, 82(1):011903, 2010.

[57] Stefano Sarao Mannelli, Florent Krzakala, Pierfrancesco Urbani, and Lenka Zdeborova. Passed & spurious: Descent algorithms and local minima in spiked matrix-tensor models. In *international conference on machine learning*, pages 4333–4342. PMLR, 2019.

[58] Stefano Sarao Mannelli, Giulio Biroli, Chiara Cammarota, Florent Krzakala, Pierfrancesco Urbani, and Lenka Zdeborová. Marvels and pitfalls of the langevin algorithm in noisy high-dimensional inference. *Physical Review X*, 10(1):011057, 2020.

[59] Francesca Mignacco, Pierfrancesco Urbani, and Lenka Zdeborová. Stochasticity helps to navigate rough landscapes: comparing gradient-descent-based algorithms in the phase retrieval problem. *Machine Learning: Science and Technology*, 2(3):035029, 2021.

[60] Elisabeth Agoritsas, Giulio Biroli, Pierfrancesco Urbani, and Francesco Zamponi. Out-of-equilibrium dynamical mean-field equations for the perceptron model. *Journal of Physics A: Mathematical and Theoretical*, 51(8):085002, 2018.

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

[62] Michael Celentano, Chen Cheng, and Andrea Montanari. The high-dimensional asymptotics of first order methods with random data. *arXiv preprint arXiv:2112.07572*, 2021.

[63] Francesca Mignacco and Pierfrancesco Urbani. The effective noise of stochastic gradient descent. *arXiv preprint arXiv:2112.10852*, 2021.

[64] Alessandro Manacorda, Grégory Schehr, and Francesco Zamponi. Numerical solution of the dynamical mean field theory of infinite-dimensional equilibrium liquids. *The Journal of chemical physics*, 152(16):164506, 2020.

[65] Kenji Fukumizu. Dynamics of batch learning in multilayer neural networks. In *International Conference on Artificial Neural Networks*, pages 189–194. Springer, 1998.

[66] Andrew M Saxe, James L McClelland, and Surya Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[67] Sanjeev Arora, Nadav Cohen, Noah Golowich, and Wei Hu. A convergence analysis of gradient descent for deep linear neural networks. In *International Conference on Learning Representations*, 2019.

[68] Madhu S Advani, Andrew M Saxe, and Haim Sompolinsky. High-dimensional dynamics of generalization error in neural networks. *Neural Networks*, 132:428–446, 2020.

[69] Arthur Jacot, François Ged, Franck Gabriel, Berfin Şimşek, and Clément Hongler. Deep linear networks dynamics: Low-rank biases induced by initialization scale and l2 regularization. *arXiv preprint arXiv:2106.15933*, 2021.

[70] Ben Poole, Subhaneil Lahiri, Maithra Raghu, Jascha Sohl-Dickstein, and Surya Ganguli. Exponential expressivity in deep neural networks through transient chaos. *Advances in neural information processing systems*, 29, 2016.

[71] Samuel S Schoenholz, Justin Gilmer, Surya Ganguli, and Jascha Sohl-Dickstein. Deep information propagation. 2016.

[72] Greg Yang and Samuel Schoenholz. Mean field residual networks: On the edge of chaos. *Advances in neural information processing systems*, 30, 2017.

[73] Greg Yang. Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. *arXiv preprint arXiv:1902.04760*, 2019.

[74] Greg Yang. Tensor programs ii: Neural tangent kernel for any architecture, 2020.

[75] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[76] Haozhe Shan and Blake Bordelon. A theory of neural tangent kernel alignment and its influence on training, 2021.

[77] James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, George Necula, Adam Paszke, Jake VanderPlas, Skye Wanderman-Milne, and Qiao Zhang. JAX: composable transformations of Python+NumPy programs, 2018.

[78] 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.

[79] John Hubbard. Calculation of partition functions. *Physical Review Letters*, 3(2):77, 1959.

[80] Charles Stein. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proceedings of the sixth Berkeley symposium on mathematical statistics and probability, volume 2: Probability theory*, volume 6, pages 583–603. University of California Press, 1972.

[81] Roman Novak, Lechao Xiao, Jaehoon Lee, Yasaman Bahri, Greg Yang, Jiri Hron, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Bayesian deep convolutional networks with many channels are gaussian processes. *arXiv preprint arXiv:1810.05148*, 2018.

[82] Greg Yang. Tensor programs i: Wide feedforward or recurrent neural networks of any architecture are gaussian processes, 2019.

[83] Yurii E Nesterov. A method for solving the convex programming problem with convergence rate o (1/kˆ 2). In *Dokl. akad. nauk Sssr*, volume 269, pages 543–547, 1983.

[84] Yurii Nesterov and Boris T Polyak. Cubic regularization of newton method and its global performance. *Mathematical Programming*, 108(1):177–205, 2006.

[85] Gabriel Goh. Why momentum really works. *Distill*, 2017.

[86] Michael Muehlebach and Michael I Jordan. Optimization with momentum: Dynamical, control-theoretic, and symplectic perspectives. *Journal of Machine Learning Research*, 22(73):1–50, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] As described in the abstract and introduction, we provide a dynamical field theory of deep networks based on kernel evolution.
   (b) Did you describe the limitations of your work? [Yes] We have an explicit limitations as the last paragraph of the paper in Section 7.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A] This work is theoretical and is very unlikely to present negative social impacts.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] We describe that our theory holds for NN architectures in the infinite width $N \to \infty$ limit.
   (b) Did you include complete proofs of all theoretical results? [Yes] All claims made in the main text are supported by derivations in the Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] Code to reproduce experimental results is provided in the supplementary material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] We provide details of all experiments in C.

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] We provided errorbars in the alignment scores of DMFT as a function of width N in Figure 1. All other runs were over a single wide network, where performance is predicted to concentrate over initialization.

(d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes] We mention our GPU usage in C.2.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] We cited the creators of Jax, Neural Tangents, and CIFAR-10.
   (b) Did you mention the license of the assets? [N/A] These are all open source provided they are appropriately credited in academic research.
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]