
Adaptive kernel predictors from feature-learning infinite limits of neural networks

Clarissa Lauditi¹ Blake Bordelon^{1 2 3} Cengiz Pehlevan^{1 2 3}

Abstract

Previous influential work showed that infinite width limits of neural networks in the lazy training regime are described by kernel machines. Here, we show that neural networks trained in the rich, feature learning infinite-width regime in two different settings are also described by kernel machines, but with data-dependent kernels. For both cases, we provide explicit expressions for the kernel predictors and prescriptions to numerically calculate them. To derive the first predictor, we study the large-width limit of feature-learning Bayesian networks, showing how feature learning leads to task-relevant adaptation of layer kernels and pre-activation densities. The saddle point equations governing this limit result in a min-max optimization problem that defines the kernel predictor. To derive the second predictor, we study gradient flow training of randomly initialized networks trained with weight decay in the infinite-width limit using dynamical mean field theory (DMFT). The fixed point equations of the arising DMFT defines the task-adapted internal representations and the kernel predictor. We compare our kernel predictors to kernels derived from lazy regime and demonstrate that our adaptive kernels achieve lower test loss on benchmark datasets.

1. Introduction

As neural-network-based artificial intelligence is increasingly impacting many corners of human life, advancing the theory of learning in neural networks is becoming more and more important, both for intellectual and safety reasons.

^{*}Equal contribution ¹John A. Paulson School of Engineering and Applied Sciences, Harvard University ²Center for Brain Sciences ³Kempner Institute. Correspondence to: Clarissa Lauditi <clauditi@g.harvard.edu>, Blake Bordelon <blake.bordelon@g.harvard.edu>, Cengiz Pehlevan <cpehlevan@seas.harvard.edu>.

Progress in this endeavor is limited but promising. An influential set of results that motivates this paper identifies infinitely-wide neural networks under certain initializations as nonparametric kernel machines (Neal, 1995; Cho & Saul, 2009a; Lee et al., 2018; de G. Matthews et al., 2018; Arora et al., 2019b; Jacot et al., 2020; Lee et al., 2020). This is important because kernels are theoretically well-understood (Rasmussen & Williams, 2006; Scholkopf & Smola, 2001), offering rich mathematical frameworks for analyzing the expressivity and generalization properties of neural networks. A drawback of this identification is that it works in the *lazy* regime of neural network training (Chizat et al., 2020), i.e. when data representations are fixed at initialization and do not evolve during learning. However, state-of-the-art deep networks operate in the *rich*, feature learning regime (Geiger et al., 2020; Vyas et al., 2022; Yang & Hu, 2022; Vyas et al., 2023), where they adapt their internal representations to the structure of the data.

Motivated by these observations, here, we ask whether infinitely wide neural networks still admit nonparametric kernel predictors in the rich domain. And if so, what characterizes these predictors? Since in the rich regime data representations evolve dynamically according to the training dynamics, identifying the nature of predictors in this regime is crucial for uncovering the principles behind feature learning. Further, as wider networks are believed to perform better on the same amount of data (Hestness et al., 2017; Novak et al., 2018; Kaplan et al., 2020; Hoffmann et al., 2022), it opens the possibility of directly training infinitely-wide feature learning networks through their corresponding kernel machines for obtaining the best performing instances of a given model architecture.

In this paper, we derive kernel predictors for infinitely wide multilayer perceptrons (MLPs) and convolutional neural networks (CNNs) in the *rich* regime. They are kernel predictors in the sense that

$$f(\mathbf{x}) = \sigma \left(\sum_{\mu=1}^P a_{\mu} K(\mathbf{x}, \mathbf{x}_{\mu}) \right) \quad (1)$$

where σ is a nonlinear function, a^{μ} are scalar coefficients, $\{\mathbf{x}_{\mu}\}_{\mu=1}^P$ are training data, and $\mathbf{x}$ is the test point. The kernel K depends on the architecture of the network, and,

importantly, adapts to the training data, unlike the Neural Tangent Kernel (NTK) (Jacot et al., 2020) and Neural Network Gaussian Process Kernel (NNGPK) (Cho & Saul, 2009b; Lee et al., 2018; de G. Matthews et al., 2018) derived from lazy infinite limits.

To arrive at these predictors, we start from existing work characterizing the feature-learning infinite-width limits of deep neural networks under gradient flow dynamics. Previously, Bordelon & Pehlevan (2022) adopted the dynamical mean-field theory (DMFT) (Martin et al., 1973; De Dominicis, 1978; Sompolinsky & Zippelius, 1981; Arora et al., 2019a; Arous et al., 2004) approach to the training dynamics of deep MLPs and CNNs under the *maximal update parameterization* (μ P) (Yang & Hu, 2022; Yang et al., 2022; Mei et al., 2018; Rotskoff & Vanden-Eijnden, 2022), which leads to a feature-learning infinite-width limit. This DMFT results in a set of stochastic integro-differential equations in terms of summary statistics, which define the deterministic training time (t) evolution of the deterministic network predictor $f(\mathbf{x}, t)$. Performing inference with this framework requires solving these equations over the training time using sophisticated Monte Carlo techniques. Further, the complexity and history dependence of the equations make the predictor interpretation challenging.

Our main observation is that we can analyze the network predictor at convergence $f(\mathbf{x}, t = \infty)$ in two ways to obtain deterministic adaptive kernel predictors: (1) by interpreting the dynamics of gradient flow with added white noise as sampling the weights from a Bayesian posterior, (2) by studying the fixed points of DMFT equations for gradient flow with weight decay. Specifically, our contributions in this work are the following:

1. We study the noisy gradient-flow dynamics with weight decay in the rich regime for MLPs and CNNs. We identify two novel infinite-width limits (Table 1, Figure 1) that lead to adaptive kernel machine interpretations of neural networks. We name the corresponding kernels *adaptive Neural Bayesian Kernel* (aNBK) and *adaptive Neural Tangent Kernel* (aNTK) for reasons that will be apparent below.
2. To analyze the first of these limits, we introduce a novel Bayesian interpretation of feature-learning neural networks, where the posterior distribution at infinite width characterizes the network’s state after training. Analyzing this posterior in the infinite width limit using statistical mechanics methods, we identify a min-max optimization problem, arising from a saddle point argument, that defines the aNBK predictor.
3. To analyze the second limit, we invoke the DMFT analysis of gradient-flow dynamics (Bordelon & Pehlevan, 2022). We show that when weight decay is added

to gradient flow dynamics the final learned network predictors behave as kernel predictors. We provide the DMFT fixed point equations that define the aNTK predictor.

4. We develop numerical methods to solve for our predictors.
5. For kernels arising from deep infinitely-wide linear Bayesian networks, we solve the saddle point equations exactly via recursion, bypassing sampling strategies needed for the nonlinear case. We analyze the behavior of kernel-task overlap parameters across depth for whitened data.
6. We provide comparisons of our adaptive kernel machines to trained networks and *lazy* NTK and NNGPK predictors for MLPs and CNNs. We demonstrate that our adaptive-kernels are descriptive of feature-learning neural network training on a variety of metrics, including test loss, intermediate feature kernels, and pre-activation densities. In addition, they outperform *lazy* kernel predictors on benchmark datasets.

1.1. Related works

Neural networks as kernel machines. In certain initialization and parameterization schemes, taking the width of a neural network to infinity leads the model to learn a kernel machine, with a kernel that depends only on the initial task-independent features which do not change during training (Jacot et al., 2020; Lee et al., 2018). This “lazy training” regime has been extensively studied, particularly in the context of infinitely wide networks (Chizat et al., 2020; Jacot et al., 2020; Lee et al., 2018; 2020). However, neural networks outside of the lazy limit (in the so-called “rich regime”) often perform better than their corresponding initial NTKs (Vyas et al., 2022) and are not obviously related to kernel machines. Domingos (2020) argues that gradient descent training for any deep network corresponds to regression with a history-dependent “path kernel”, though this definition does not satisfy the standard representer theorem where coefficients are only functions of the training data as in Equation (1). In general, the learned function of a network trained with gradient flow can always be written as an integral over the history of error coefficients on training data and a time evolving NTK. In the present work, however, we are focused on when the final solution a network converges to satisfies a *history independent* representer theorem as in Equation 1. Some experimental and theoretical works have indicated that this is often the case, where regression with the *final* NTK (computed with gradients at the final parameters) of a network provides an accurate approximation of the learned neural network function during rich training (Geiger et al., 2020; Atanasov et al., 2021; Wei et al., 2022). A complete theoretical understanding of when

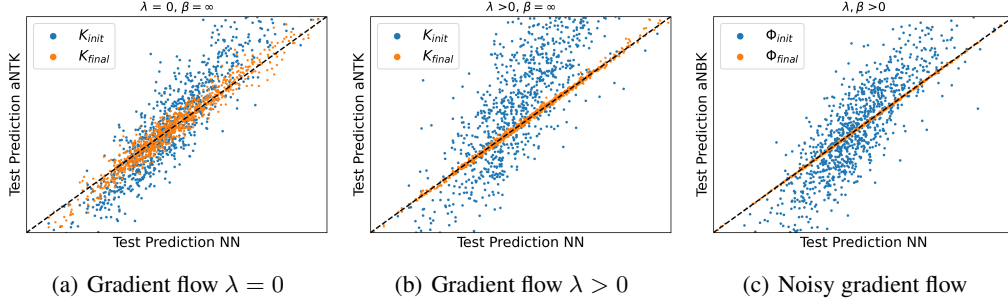


Figure 1. Test network predictors of a two-layer MLP (width $N = 5000$) trained with $P = 300$ data of two-classes of CIFAR10 compared with the theoretical kernel regression predictors. Three panels are for different regimes of regularization λ and temperature $1/\beta$. In all the three cases of λ , K_{init} represent the network predictors at initialization, while $K_{\text{final}}/\Phi_{\text{final}}$ correspond either to aNTK at convergence for (a)/(b) or aNBK that at convergence for (c). (a) When $\lambda = 0$ the NN predictor of gradient flow without weight decay is not a kernel predictor; (b) instead, when $\lambda > 0$ the test prediction is well captured by f_{aNTK} (Eq. (11)). (c) Bayesian empirical predictor is well-described by aNBK predictor at convergence. Here, the Bayesian predictor refers to the predictor obtained after thermalization when training with noisy gradient flow (i.e., Langevin dynamics). We expect these matches to be exact for adaptive kernels at infinite width.

and why the correspondence between final NTK and final network function holds is currently lacking.

Adaptive kernels. In Bayesian neural networks, several works have identified methods that describe learned network solutions beyond the lazy infinite-width description of NNGP regression. Some works pursue perturbative approximations to the posterior in powers of $1/\text{width}$ (Zavatone-Veth et al., 2022a; Roberts et al., 2022) or alternative cumulant expansions of the predictor statistics (Naveh & Ringel, 2021). Others analyze the Bayesian posterior for deep linear networks, which are more analytically tractable (Aitchison, 2020; Hanin & Zlokapa, 2023; Zavatone-Veth et al., 2022b; Bassetti et al., 2024) and actually also capture the behavior of deep Bayesian nonlinear student-teacher learning in a particular scaling limit (Cui et al., 2023). Several works on Bayesian deep learning have argued in favor of a proportional limit where the samples and width are comparable (Li & Sompolinsky, 2021; Pacelli et al., 2023; Aiudi et al., 2023; van Meegen & Sompolinsky, 2024; Baglioni et al., 2024; Fischer et al., 2024). In the context of fully connected (MLP) linear neural networks, (Li & Sompolinsky, 2021; Pacelli et al., 2023) argued that, the mean predictor under the posterior is the same as regression with the lazy NNGP kernel, though with a rescaled ridge and a predictor variance that depends on scale factor $Q(\alpha)$ that both change as a function of $\alpha = P/N$. Extending this result to deep networks Pacelli et al. (2023) found that each layer has a scale renormalization constant $Q_\ell(\alpha)$, while Aiudi et al. (2023) showed that convolutional architectures are characterized by a matrix of scale renormalization constants $Q_{s,s'}(\alpha)$ that capture learned space $\times$ space that adapt to the data.

In the same NTK parameterization, Fischer et al. (2024) and Seroussi et al. (2023) developed theories of feature learning where kernels adapt more flexibly (each entry in the kernel

can adapt due to feature learning). In this setting, feature learning acts as a $1/\text{width}$ effect in the posterior measure. Fischer et al. (2024) derive a general large deviation principle for the distribution of kernels under the posterior, which holds for any $\alpha = P/N$, however, they focus on linear response theory for the kernel updates since feature learning induces $\mathcal{O}(N^{-1})$ corrections to the the posterior distribution of kernels. Since the kernels do not concentrate in the proportional limit $P, N \rightarrow \infty$ with $P/N = \alpha$, Fischer et al. (2024) also compute how fluctuations in the kernels propagate across layers. They show that the linear response theory used to track fluctuations can also be used to compute feature learning corrections when linearizing the action around the NNGP kernels. Their results are thus non-perturbative in $\alpha = P/N$ but perturbative in what we call “richness” γ_0 . Similarly, (Seroussi et al., 2023) explore variational approximations of the hidden neuron activation densities at finite width. More recent versions of these proportional theories have begun to explore other parameterizations (Rubin et al., 2024a) in linear networks, resorting to cumulant expansion of activation densities (up to Gaussian order) in one-hidden layer non-linear networks.

Closer to our approach, Yang et al. (2023) analyze rescalings of the Bayesian likelihood to induce representation learning at infinite width. This approach has also been used to explain sharp transitions in the behavior of the posterior as hyperparameters, such as prior weight variance, are varied in large width networks (Rubin et al., 2024b). Alternatively, some works have developed non-neural adaptive kernel algorithms, which filter information based on gradients of a kernel solution with respect to the input variables (Radhakrishnan et al., 2022), which exhibit improvements in performance over the initial kernel and can capture interesting feature learning phenomena such as grokking (Mallinar

et al., 2024).

In our work, we study the infinite width $N \rightarrow \infty$ limit of Bayesian networks at fixed P in the *mean-field* (μP) parameterization setting. As a consequence, feature learning is not driven by finite width and cannot generally be extracted from linear response theory (or can be interpreted in terms of fluctuations since kernels concentrate). Contrary to previous results (Li & Sompolinsky, 2021; Pacelli et al., 2023; Aiudi et al., 2023; van Meegen & Sompolinsky, 2024; Baglioni et al., 2024), in our theories, the kernels deterministically adapt to data. These kernels can exhibit arbitrarily large changes in their structure (rather than just scale or spatial renormalization) at infinite width, and these cannot be obtained from linear response theory, at odds with previous computations (Fischer et al., 2024; Seroussi et al., 2023; Rubin et al., 2024a) (see Figures 1 & 3). Additionally, our adaptive kernels at infinite width are deterministic quantities, while in previous scaling theories, one has to compute kernel fluctuation statistics to get non-parametric predictors in the feature learning regime. To the best of our knowledge, we are also the first to solve for these adaptive kernels in the Bayesian setting without resorting to any perturbative (in richness γ_0) approach nor any Gaussian approximation of hidden layer pre-activation distributions, which are non-Gaussian at infinite width.

2. Preliminaries

We start by describing our setup. Here, for ease of presentation, we discuss the fully connected MLP setting, referring to Appendix D for the case of CNNs.

For an empirical training dataset $\mathcal{D} = \{\mathbf{x}_\mu, y_\mu\}_{\mu=1}^P$ of size P , input vectors $\mathbf{x}_\mu \in \mathbb{R}^D$ and labels y_μ , we define the output of an MLP with L layers as

$$f_\mu = \sigma \left(\frac{1}{\gamma \sqrt{N_L}} \mathbf{w}^{(L)} \cdot \phi(\mathbf{h}_\mu^L) \right), \quad (2)$$

$$\mathbf{h}_\mu^{\ell+1} = \frac{1}{\sqrt{N_\ell}} \mathbf{W}^\ell \phi(\mathbf{h}_\mu^\ell), \quad \mathbf{h}_\mu^1 = \frac{1}{\sqrt{D}} \mathbf{W}^{(0)} \mathbf{x}_\mu,$$

where $\phi(\cdot)$ represents a homogeneous transfer function in its parameters $\phi(a\theta) = a^\kappa \phi(\theta)$, $\mathbf{h}_\mu^\ell \in \mathbb{R}^{N_\ell}$ at layer ℓ is the pre-activation vector and $\mathbf{W}^\ell \in \mathbb{R}^{N_{\ell+1} \times N_\ell}$ is the matrix of weights to be learned. We initialize each trainable parameter as a Gaussian random variable $W_{ij}^\ell \sim \mathcal{N}(0, 1)$ with unit variance, in such a way that in the infinite width limit $N_\ell = N \rightarrow \infty$, $\forall \ell \in \{L\}$ the pre-activations at each layer will remain $\Theta_N(1)$. At the same time, we scale the network output by a factor $\gamma = \gamma_0 \sqrt{N}$ in order to study feature learning. This allows to interpolate between a lazy limit description of NNs when $\gamma_0 \rightarrow 0$, and a rich regime description when $\gamma_0 = \Theta_N(1)$. This parameterization, known as *maximal update parameterization* (μP) (Yang & Hu, 2022), allows feature learning by enabling preactivations to evolve from their initialization during training even in the infinite-

width limit.

3. Adaptive kernel limits of training dynamics

Next, we study the rich training dynamics of the NN defined as in Eq. (2) to arrive at adaptive kernel predictors. In particular, we study the infinite time limit $t \rightarrow \infty$ of the noisy gradient-flow dynamics

$$d\theta(t) = -\gamma^2 \nabla_\theta \mathcal{L} dt - \lambda \beta^{-1} \theta(t) dt + \sqrt{2\beta^{-1}} d\epsilon(t) \quad (3)$$

for the collection of weights $\theta = \text{Vec}\{\mathbf{W}^{(0)}, \dots, \mathbf{w}^{(L)}\}$, a loss function $\mathcal{L}(\theta)$ and for a ridge λ , $\forall \ell \in \{L\}$. Here, $\gamma^2 = \gamma_0^2 N$ ensures the feature updates to be $\Theta_N(1)$ in the infinite width limit (Bordelon & Pehlevan, 2022), and $d\epsilon$ is a Brownian motion with covariance structure $\langle d\epsilon(t) d\epsilon(t') \rangle = \delta(t - t') \mathbf{I}$ (being δ the Dirac delta), whose contribution to the dynamics can be switched off by tuning the temperature $T = \frac{1}{\beta} \rightarrow 0$. When the Brownian motion is on, this dynamics can be interpreted as sampling from a Bayesian posterior, which will be detailed below. When it is turned off, this is gradient-flow dynamics with weight decay.

It is well known that infinite limits of this training dynamics in the lazy regime ($\gamma_0 \rightarrow 0$) lead to kernel machines defined by the NTK (Jacot et al., 2020) and NNGPK (Cho & Saul, 2009b; Lee et al., 2018; de G. Matthews et al., 2018). Here, we show that there are infinite limits in the rich regime that also lead to kernel predictors, but this time these kernels adapt to data. The order of limits for width, time, temperature, and feature learning strength parameters $\{N, t, \beta, \gamma_0\}$ to get either the already known *lazy* (NNGPK, NTK) or novel adaptive kernel predictors is shown in Table 1. The latter kernels correspond either to the infinite width limit of a NN at convergence (i.e. $t \rightarrow \infty$) that learns with $\beta > 0$ (aNPK) or to the infinite time limit of an infinitely wide NN learning with gradient flow and weight decay (i.e. $\beta \rightarrow \infty$) (aNTK).

NNGPK	aNBK (ours)
$\lim_{\gamma_0 \rightarrow 0} \lim_{N \rightarrow \infty} \lim_{t \rightarrow \infty}$ $\beta = \Theta_N(1)$	$\lim_{N \rightarrow \infty} \lim_{t \rightarrow \infty}$ $\{\gamma_0, \beta\} = \Theta_N(1)$
NTK	aNTK (ours)
$\lim_{\gamma_0 \rightarrow 0} \lim_{N \rightarrow \infty} \lim_{\beta \rightarrow \infty}$ $t = \Theta_N(1)$	$\lim_{t \rightarrow \infty} \lim_{N \rightarrow \infty} \lim_{\beta \rightarrow \infty}$ $\gamma_0 = \Theta_N(1)$

Table 1. Limiting orders for $\{N, t, \beta, \gamma_0\}$ in the dynamics (3) to get either (known) static kernels NNGPK & NTK (left column), or (new) adaptive kernel predictors (right column).²

Next, we give the functional forms of our novel aNTK and aNBK predictors and briefly discuss their derivations. Details are given in Appendix A.

3.1. aNBK

As described in Table 1, if we take first the $t \rightarrow \infty$ limit, and then the $N \rightarrow \infty$ limit with temperature $(1/\beta)$ and feature learning strength (γ_0) fixed, we arrive at an adaptive kernel predictor (see Fig. 1)

$$f_{\text{aNBK}}(\mathbf{x}) = \sigma \left(\frac{\beta}{\lambda_L} \sum_{\mu=1}^P \Delta_{\mu} \Phi^L(\mathbf{x}_{\mu}, \mathbf{x}) \right), \quad (4)$$

with $\Delta_{\mu} = -\frac{\partial \mathcal{L}}{\partial s_{\mu}}$ being the error signal, $s_{\mu} = \frac{1}{\gamma_0 N} \mathbf{w}^{(L)} \cdot \phi(\mathbf{h}_{\mu}^L)$ the pre-readout as in Eq. 2, and $\Phi_{\mu}^L = \frac{1}{N} \phi(\mathbf{h}_{\mu}^L) \cdot \phi(\mathbf{h}^L)$ the train-test feature kernel at the last layer L . In this notation, the kernel matrix element is given by $\Phi_{\mu\nu}^L = \Phi^L(\mathbf{x}_{\mu}, \mathbf{x}_{\nu})$. In the squared loss case with a linear readout ($\sigma(s) \equiv s$), the predictor $f(\mathbf{x})$ becomes a kernel regression predictor of the form

$$f_{\text{aNBK}}(\mathbf{x}) = (\Phi^L(\mathbf{x}))^{\top} \left[\Phi^L + \lambda_L \frac{\mathbf{I}}{\beta} \right]^{-1} \mathbf{y}, \quad (5)$$

where $\Phi^L(\mathbf{x})_{\mu} = \Phi^L(\mathbf{x}, \mathbf{x}_{\mu})$ (for derivation, see Appendix A.2).

In Appendix A, we show that the kernel Φ^L is given by a solution to a min-max optimization problem that involves the data-adaptive kernel $\Phi^L \in \mathbb{R}^{P \times P}$, intermediate layer adaptive-kernels $\Phi^{\ell} \in \mathbb{R}^{P \times P}$ and dual adaptive-kernel variables $\hat{\Phi}^{\ell} \in \mathbb{R}^{P \times P}$. Here, we present this min-max problem for the squared loss and linear readout for simplicity, see Appendix A for the full expressions. First, we define the action:

$$S(\{\Phi^{\ell}, \hat{\Phi}^{\ell}\}) = -\frac{1}{2} \sum_{\ell=1}^L \text{Tr} \Phi^{\ell} \hat{\Phi}^{\ell} + \frac{\gamma_0^2}{2} \mathbf{y}^{\top} \left(\frac{\mathbf{I}}{\beta} + \frac{\Phi^L}{\lambda_L} \right)^{-1} \mathbf{y} - \sum_{\ell=1}^{L-1} \ln \mathcal{Z}_{\ell}[\Phi^{\ell-1}, \hat{\Phi}^{\ell}]. \quad (6)$$

where the functions $\mathcal{Z}_{\ell}[\Phi^{\ell-1}, \hat{\Phi}^{\ell}]$ are defined as

$$\mathcal{Z}_{\ell}[\Phi^{\ell-1}, \hat{\Phi}^{\ell}] = \int d\mathbf{h}^{\ell} \exp \left(-\frac{\lambda_{\ell-1}}{2} (\mathbf{h}^{\ell})^{\top} (\Phi^{\ell-1})^{-1} \mathbf{h}^{\ell} \right) \exp \left(-\frac{1}{2} \phi(\mathbf{h}^{\ell})^{\top} \hat{\Phi}^{\ell} \phi(\mathbf{h}^{\ell}) \right), \quad (7)$$

with base case $\Phi_{\mu\nu}^0 \equiv \frac{1}{D} \mathbf{x}_{\mu} \cdot \mathbf{x}_{\nu}$. Then, the saddle point that dominates the distribution is

$$\{\Phi_{*}^{\ell}, \hat{\Phi}_{*}^{\ell}\}_{\ell=1}^L = \arg \min_{\{\Phi^{\ell}\}} \max_{\{\hat{\Phi}^{\ell}\}} S(\{\Phi^{\ell}, \hat{\Phi}^{\ell}\}). \quad (8)$$

Derivation sketch. Taking $t \rightarrow \infty$ at fixed temperature $\beta = \frac{1}{T}$ and finite width N in Eq. (3) converges to a stationary distribution (Kardar, 2007; Welling & Teh, 2011;

Mingard et al., 2020; Naveh et al., 2021) over the trainable parameters θ given the dataset $\mathcal{D}$, which can be interpreted as a Bayesian posterior with log-likelihood $-\beta \gamma^2 \mathcal{L}(\theta)$ and a Gaussian prior of scale $\lambda_{\ell}^{-1/2}$

$$p(\theta|\mathcal{D}) = \frac{1}{Z} \exp \left[-\beta \gamma^2 \mathcal{L}(\theta) - \sum_{\ell=0}^L \frac{\lambda_{\ell}}{2} \|\theta^{\ell}\|^2 \right]. \quad (9)$$

The distribution of Eq. (9) can be studied in the overparameterized thermodynamic limit where the width at each layer $N \rightarrow \infty$ while $P = \Theta_N(1)$. When $\beta \rightarrow \infty$, the posterior is dominated by the set of global minimizers of the loss for the training data (i.e. solutions θ that minimize $\mathcal{L}(\theta)$). We are interested in the partition function Z of Eq. (9), which is function of the order parameters $\{\Phi^{\ell}, \hat{\Phi}^{\ell}\}_{\ell=1}^L$, since it has the form $Z \propto \int \prod_{\ell=1}^L d\Phi^{\ell} d\hat{\Phi}^{\ell} \exp(-N S(\Phi^{\ell}, \hat{\Phi}^{\ell}))$ where $S(\Phi^{\ell}, \hat{\Phi}^{\ell})$ is the Bayesian action given in Equation (6). When N is large, $p(\theta|\mathcal{D})$ is exponentially dominated by the saddle points of S , and looking for the values of $\{\Phi^{\ell}, \hat{\Phi}^{\ell}\}_{\ell=1}^L$ which makes S locally stationary means solving the min-max optimization problem in Algorithm 1. In the lazy learning limit $\gamma_0 \rightarrow 0$, the dual kernels vanish $\hat{\Phi}^{\ell} \rightarrow 0$ and we recover Gaussian preactivation densities, consistent with the NNGP theory (Lee et al., 2018). A full account of this derivation is given in Appendix A.1. $\square$

We present a numerical algorithm to calculate the aNBK Regression Predictor from data in Alg. 1.

Algorithm 1 aNBK Regression Predictor

Input: Dataset $\mathcal{D} = \{\mathbf{x}_{\mu}, y_{\mu}\}_{\mu=1}^P$ with covariance $\Phi_{\mu\nu}^0 = \frac{1}{D} \mathbf{x}_{\mu} \cdot \mathbf{x}_{\nu}$, hyperparameters $\{\gamma_0, \beta, \lambda_{\ell}\}$, step size δ .

Output: Predictor $f(\mathbf{x})$ for any test point $\mathbf{x} \in \{P_{\text{test}}\}$.

Generate: Initial guesses for $\{\Phi^{\ell}, \hat{\Phi}^{\ell}\}_{\ell=1}^L$:

$\Phi^{\ell} = \langle \phi(\mathbf{h}^{\ell}) \phi(\mathbf{h}^{\ell})^{\top} \rangle_{\mathbf{h}^{\ell} \sim \mathcal{N}(0, \Phi^{\ell-1})}$, and $\hat{\Phi}^{\ell} = 0 \quad \forall \ell$.

while Kernels do not converge **do**

 Define action the action S of Equation (6) as differentiable function of $\{\Phi^{\ell}, \hat{\Phi}^{\ell}\}$, using importance sampling to estimate the functions $\mathcal{Z}_{\ell}$.

 Solve the inner optimization problem

$$\max_{\hat{\Phi}^{\ell}} S(\{\Phi^{\ell}, \hat{\Phi}^{\ell}\})$$

 with gradient ascent

 Perform gradient updates on feature kernels

$$\Phi^{\ell} \leftarrow \Phi^{\ell} - \delta \frac{\partial}{\partial \Phi^{\ell}} S(\{\Phi^{\ell}, \hat{\Phi}^{\ell}\})$$

Compute: For a test point $\mathbf{x}$, $\Phi^{\ell}(\mathbf{x}) = \langle \phi(\mathbf{h}^{\ell}(\mathbf{x})) \phi(\mathbf{h}^{\ell}) \rangle_{p(\mathbf{h}, \mathbf{x})}$ with importance sampling and $p(\mathbf{h}, \mathbf{x})$ from Eq. (7) (see A.2).

return $f_{\text{aNBK}}(\mathbf{x})$ as in Eq. (5)

3.2. aNTK

The second order of limits we study is when $\beta \rightarrow \infty$ in Eq. (3). We are interested in the infinite time limit $t \rightarrow \infty$ of the dynamics when $N \rightarrow \infty$, which leads to a predictor

$$f_{\text{aNTK}}(\mathbf{x}) = \sigma \left(\frac{1}{\kappa \lambda_L} \sum_{\mu=1}^P \Delta_{\mu} K^{\text{aNTK}}(\mathbf{x}_{\mu}, \mathbf{x}) \right), \quad (10)$$

where again $\Delta_{\mu} = -\frac{\partial \mathcal{L}}{\partial s_{\mu}}$, s_{μ} the output pre-activation and $K_{\mu\nu}^{\text{aNTK}} = \lim_{t \rightarrow \infty} \frac{\partial f_{\mu}(t)}{\partial \theta} \cdot \frac{\partial f_{\nu}(t)}{\partial \theta} = \lim_{t \rightarrow \infty} \sum_{\ell} G_{\mu\nu}^{\ell+1}(t, t) \Phi_{\mu\nu}^{\ell}(t, t)$ is the *adaptive Neural Tangent Kernel*. The gradient kernel $G_{\mu\nu}^{\ell}(t, t) = \frac{1}{N} \mathbf{g}_{\mu}^{\ell}(t) \cdot \mathbf{g}_{\nu}^{\ell}(t)$ represents the inner products of gradient vectors $\mathbf{g}_{\mu}^{\ell}(t) \equiv N \gamma_0 \frac{\partial s_{\mu}(t)}{\partial \mathbf{h}_{\mu}^{\ell}(t)}$ which are usually computed with back-propagation.

For a squared loss and a linear readout, a homogenous activation function $\phi(\cdot)$, the final predictor has the form

$$f_{\text{aNTK}}(\mathbf{x}_{*}) = \mathbf{k}^{\text{aNTK}}(\mathbf{x}_{*})^{\top} [\mathbf{K}^{\text{aNTK}} + \lambda_L \kappa \mathbf{I}]^{-1} \mathbf{y}. \quad (11)$$

The factor κ appears in the expressions from the weight decay contribution to the dynamics (3), since we consider a κ degree homogeneous network as specified in 2.

In the infinite width limit $N \rightarrow \infty$, the neurons in each hidden layer become independent and the Φ, G kernels can be computed as averages (Bordelon & Pehlevan, 2022)

$$\begin{aligned} \Phi_{\mu\nu}^{\ell}(t, t) &= \langle \phi(h_{\mu}^{\ell}(t)) \phi(h_{\nu}^{\ell}(t)) \rangle \\ G_{\mu\nu}^{\ell}(t, t) &= \langle g_{\mu}^{\ell}(t) g_{\nu}^{\ell}(t) \rangle \end{aligned} \quad (12)$$

where $\langle \cdot \rangle$ denotes the averages over a stochastic process for pairs $\{h_{\mu}^{\ell}, z_{\mu}^{\ell}\}_{\mu=1}^P$ which obey the dynamics

$$\begin{aligned} h_{\mu}^{\ell}(t) &= e^{-\lambda t} \xi_{\mu}^{\ell}(t) \\ &+ \gamma_0 \int_0^t dt' e^{-\lambda(t-t')} \sum_{\nu} \Delta_{\nu}(t') g_{\nu}^{\ell}(t') \Phi_{\mu\nu}^{\ell-1}(t, t') \\ z_{\mu}^{\ell}(t) &= e^{-\lambda t} \psi_{\mu}^{\ell}(t) \\ &+ \gamma_0 \int_0^t dt' e^{-\lambda(t-t')} \sum_{\nu} \Delta_{\nu}(t') \phi(h_{\nu}^{\ell}(t')) G_{\mu\nu}^{\ell+1}(t, t') \\ g_{\mu}^{\ell}(t) &= \dot{\phi}(h_{\mu}^{\ell}(t)) z_{\mu}^{\ell}(t) \end{aligned} \quad (13)$$

where $\xi_{\mu}^{\ell}(t), \psi_{\mu}^{\ell}(t)$ are stochastic processes inherited from the initial conditions, which become suppressed at large times. These equations are one way (but not the only way) to converge to a set of fixed point condition for the final features and final predictor (see Appendix F). Alg. 2 provides pseudocode for calculating the predictor in this setting.

Derivation sketch. A derivation to obtain the DMFT dynamics as in Eq. (13) can be found in (Bordelon & Pehlevan, 2022). Here, we want to stress the difference between

the $\lambda = 0$ (discussed in (Bordelon & Pehlevan, 2022)) and $\lambda > 0$ cases, the last of which leads to the kernel predictor as in Eq. (10). In the case of gradient flow with weight decay, since the predictor dynamics can be written by the chain rule $\frac{df_{\mu}}{dt} = \frac{df_{\mu}}{ds_{\mu}} \frac{ds_{\mu}}{dt}$, we can just track the dynamics of the output pre-activation

$$\frac{ds_{\mu}}{dt} = \sum_{\alpha=1}^P K_{\mu\alpha}^{\text{aNTK}}(t, t) \Delta_{\alpha}(t) - \lambda_L \kappa s_{\mu} \quad (14)$$

if we suppose $\sigma'(s_{\mu}) \neq 0$ ³. How to arrive at this formula can be found in Appendix E. Here, at the fixed point the output pre-activation is $s_{\mu} = \frac{1}{\lambda_L \kappa} \sum_{\mu} \Delta_{\mu} K^{\text{aNTK}}(t, t)$, which recovers the kernel predictor of Eq. (10). $\square$

In principle, for any given value of λ , in order to have an estimate of $\mathbf{K}^{\text{aNTK}}$ at convergence, one has to simulate the stochastic non-Markovian field dynamics as shown in Algorithm 2. When $\lambda > 0$, the contribution from initial conditions $\xi_{\mu}^{\ell}(t), \psi_{\mu}^{\ell}(t)$ (see Eq. (13)), is exponentially suppressed at large time, while the second term of Eq. (13) contributes the most only when the system has reached convergence. This is not true if we switch-off the regularization λ , in which case the contribution from $\xi_{\mu}^{\ell}(t), \psi_{\mu}^{\ell}(t)$ persist late in training, since without the weight decay term, the initial conditions prevent the dynamics from converging to a fixed kernel predictor.

In Fig. 1 we clearly demonstrate this, by comparing the predictor of a two-layer MLP trained on a subset of CIFAR10 with the theoretical predictor of Eq. (11). For the first case when $\lambda = 0$, the network predictor at convergence is not the kernel predictor aNTK. Instead, when $\lambda > 0$, the network dynamics is well-described by Eq. (11). We refer to Appendix E.1 for the case of CNNs.

Algorithm 2 aNTK Regression Predictor

Input: Data $\Phi^0, \mathbf{y}$ and hyperparameters $\{\gamma_0, \lambda_{\ell}\}$.

Output: Predictor $f_{\text{aNTK}}(\mathbf{x})$ for any test point $\mathbf{x} \in \{P_{\text{test}}\}$.

Generate: Initial guesses for $\{\Phi^{\ell}, \mathbf{G}^{\ell}\}_{\ell=1}^L$: $\Phi^0, \mathbf{G}^{L+1} = \mathbf{11}^{\top}$.

Draw $\mathcal{S}$ samples for random fields at initialization $\xi_{\mu,s}^{\ell}(t) = \frac{1}{\sqrt{N}} W^{\ell}(0) \phi(h_{\mu,s}^{\ell-1}(t))$ and $\psi_{\mu,s}^{\ell}(t) = \frac{1}{\sqrt{N}} W^{\ell}(0) g_{\mu,s}^{\ell+1}(t)$

while *Kernels do not converge* $\forall \ell \in \{L\}, \forall s \in \{\mathcal{S}\}$ **do**
 Implement the non-Markovian dynamics of Eq. (13)
 Compute new $\{\Phi^{\ell}, \mathbf{G}^{\ell}\}$

Compute $K_{\mu\nu}^{\text{aNTK}} = \lim_{t \rightarrow \infty} \sum_{\ell=0}^L G_{\mu\nu}^{\ell+1}(t, t) \Phi_{\mu\nu}^{\ell}(t, t)$

return $f_{\text{aNTK}}(\mathbf{x})$ as in Eq. (11)

³As specified in the Appendix E, we restrict the readout activations to those with $\sigma'(s_{\mu}) \neq 0$, otherwise, the gradient signal does not backpropagate through the network, preventing convergence to a kernel predictor.

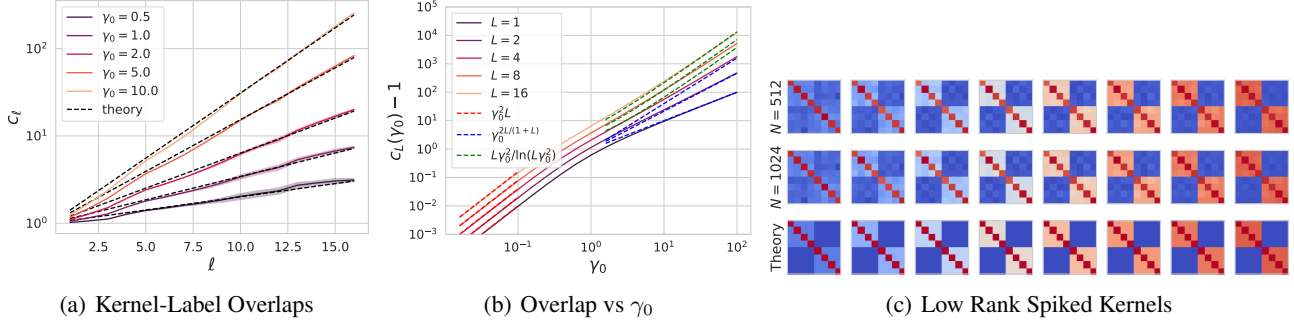


Figure 2. Linear networks with whitened data are determined by a set of kernel-label overlap matrices. (a) The overlap variables c_ℓ increase exponentially with ℓ with rate that depends on γ . Solid lines taken from Langevin dynamics on $N = 1024$ network. (b) The alignment of the final layer c_L as a function of γ_0 and L exhibits three distinct scaling regimes. (c) Examples of learned kernels (at each layer ℓ) in depth $\ell \in \{L = 8\}$, $\gamma_0 = 4.0$ and finite width N networks compared to the $N \rightarrow \infty$ theory.

4. Infinitely-wide feature learning deep linear networks

Deep linear networks ($\phi(h) \equiv h$) provide a simpler framework for analysis than their nonlinear counterparts (Saxe et al., 2014; Advani & Saxe, 2017; Arora et al., 2019a; Aitchison, 2020; Li & Sompolinsky, 2021; Jacot et al., 2022; Zavatone-Veth et al., 2022a), yet they still converge to non-trivial feature aligned solutions. In deep linear networks, preactivations remain Gaussian at each layer when $P = \Theta_N(1)$ for the Central Limit Theorem (CLT), and this greatly simplifies the saddle point equations to algebraic formulas which close in terms of the kernels for both aNBK and aNTK theories. This means that we can solve for the adaptive kernels in both cases without any refined sampling strategy. The deep linear case of gradient flow dynamics can be found in (Bordelon & Pehlevan, 2022). Here, we report the solution to the saddle point equations that define the kernels in the feature-learning Bayesian setting, specializing to the regression problem (the generic loss case can be found in Appendix A)

$$\begin{aligned}
 \Phi^\ell - \frac{\Phi^{\ell-1}}{\lambda_{\ell-1}} \left(\mathbf{I} + \frac{\Phi^{\ell-1}}{\lambda_{\ell-1}} \hat{\Phi}^\ell \right)^{-1} &= 0 \quad \forall \ell = 1, \dots, L \\
 \hat{\Phi}^\ell - \frac{\hat{\Phi}^{\ell+1}}{\lambda_\ell} \left(\mathbf{I} + \frac{\Phi^\ell}{\lambda_\ell} \hat{\Phi}^{\ell+1} \right)^{-1} &= 0 \quad \forall \ell = 1, \dots, L-1 \\
 \hat{\Phi}^L + \frac{\gamma_0^2}{\lambda_L} \left(\frac{\mathbf{I}}{\beta} + \frac{\Phi^L}{\lambda_L} \right)^{-1} \mathbf{y} \mathbf{y}^\top \left(\frac{\mathbf{I}}{\beta} + \frac{\Phi^L}{\lambda_L} \right)^{-1} &= 0.
 \end{aligned} \tag{15}$$

Here, as a consistency check, it is easy to see that in the lazy limit $\gamma_0 \rightarrow 0$ the dual kernels $\hat{\Phi}^\ell = 0 \quad \forall \ell \in \{L\}$, and as a consequence all the kernels Φ^ℓ will stay equal to the data covariance matrix Φ^0 , consistent with lazy learning. However, for the rich regime where $\gamma_0 > 0$ the $\hat{\Phi}^\ell$ kernels do not vanish and alter the fixed point kernels Φ^ℓ with target dependent information in the form of low rank spikes (by definition of $\hat{\Phi}^L$).

To illustrate this spike effect in the learned kernels in the rich limit, we specialize to whitened input data $\mathbf{K}^x = \mathbf{I}$. We show in Appendix A.3 that the equations get simplified even further, since now the kernels Φ^ℓ only grows in the rank-one $\mathbf{y} \mathbf{y}^\top$ direction. By defining some set of scalar variables $\{c_\ell, \hat{c}_\ell\}_{\ell=1}^L$, which are the overlaps with the label direction $\mathbf{y}^\top \Phi^\ell \mathbf{y} = c_\ell$ and $\mathbf{y}^\top \hat{\Phi}^\ell \mathbf{y} = \hat{c}_\ell$ we find

$$c_\ell = \left(1 + \frac{\gamma_0^2 c_L}{(\beta^{-1} + c_L)^2} \right)^\ell \quad \forall \ell \in \{L\} \tag{16}$$

which means that there is an exponential dependence of the overlap on the layer index ℓ (full derivation can be found in the Appendix) as Fig. 2(a) shows. From Eq. (16) we derive the scalings for lazy, large depth, and large feature strength limits. For the last layer overlap c_L these are

$$\begin{aligned}
 c_L &\sim 1 + L\gamma_0^2 & \gamma_0^2 L &\rightarrow 0 \\
 c_L &\sim \gamma_0^{2L/(L+1)} & \gamma_0 &\rightarrow \infty, L \text{ fixed} \\
 c_L &\sim \frac{L\gamma_0^2}{\ln(L\gamma_0^2)} & L &\rightarrow \infty, \gamma_0 \text{ fixed}
 \end{aligned} \tag{17}$$

which closely match the theory in their respective regimes plotted in Fig. 2(b). In Fig. 2(c) we show examples of learned kernels for a $L = 8$ network and $\gamma_0 = 4.0$ matching the finite width $N = 1028$ network trained with Langevin dynamics.

5. Numerical Results

Two-layer MLPs. In Fig. 3(a) we compare test losses of lazy vs feature learning kernels for a two-layer MLP trained on a P subset of two classes of CIFAR10 in a regression task. The green curve is the performance of NNGPK, Orange is the aNTK, red is the aNBK. There is a gap in performance between the lazy predictors and the adaptive feature learning predictors. However, when sample size P is small, feature

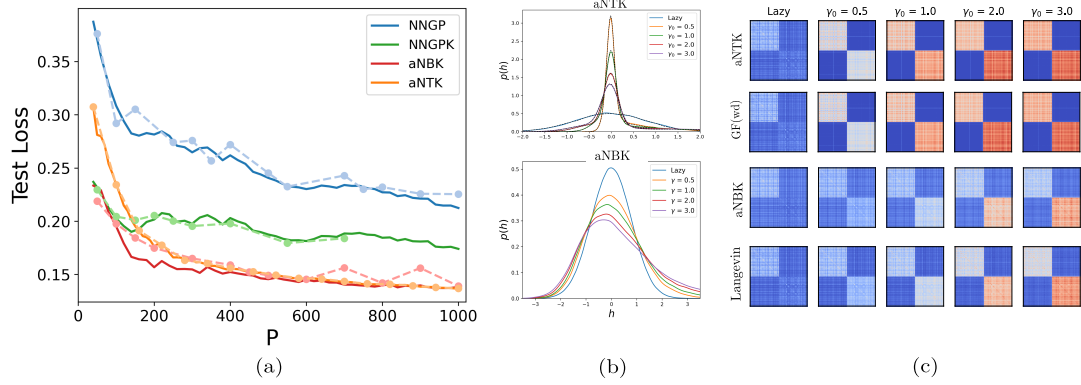


Figure 3. Feature learning theories outperform lazy predictors for a two-layer MLP trained with Squared Loss (SL) on two classes of CIFAR10 (airplane vs automobile). (a) Test losses as a function of sample size P . Solid lines refer to theories, dashed lines to numerical simulations on a $N = 5000$ network. Blue is the NNGP lazy predictor; green is the deterministic NNGPK kernel predictor; orange is aNTK with feature learning strength $\gamma_0 = 0.3$; red is aNBK predictor with the same γ_0 . (b) Non-Gaussian pre-activation densities as a function of γ_0 for (top) aNTK and (bottom) aNBK. Black dashed lines are theoretical predictions. (c) Learned feature kernels of the adaptive theories closely match their relative finite width N network trainings and evolve with γ_0 .

learning in a data-limited scenario can let the model to overfit on test points and lazy learning can be beneficial in a small window of P .

In this plot, we also include the Neural Network Gaussian Process in blue (Neal, 1995; Lee et al., 2018; de G. Matthews et al., 2018), where for a number of patterns P the solution space is sampled from the posterior of Eq. (9) by taking $\gamma = \Theta_N(1)$. Here the mean predictor is equivalent to the NNGPK predictor, however there is also a variance term, which comes from the fact that we are averaging over all possible random weights.

In the rich scenarios, we derive the preactivation distributions $p(\mathbf{h})$ as a function of γ_0 (Fig. 3(b) top and bottom) at convergence. At initialization, $p(\mathbf{h})$ follows $\mathcal{N}(0, \Phi^0)$. However, as learning proceeds and features are learned, the densities accumulate non-Gaussian contributions which are identified by our theories (e.g. Eq. (7)). Fig. 3(c) shows that there is a clustering of $P = 100$ data points by category in the feature space defined by the adaptive kernels.

Deep MLPs. In Fig. 4 we show simulations of a Bayesian $L = 5$ Tanh MLP and compare to our infinite-width predictors. In order to study how feature learning propagates through depth, we plot the theoretical adaptive kernels of the aNBK theory versus the empirical kernels of Langevin dynamics at thermalization as a function of the layer index ℓ . Coherently with the deep linear case in 4, the last layer feature kernel aligns first with the labels, since the clustering of features by class is more evident in this case. There is a small difference between theory and experiments because solving the min-max problem when $L > 1$ makes the convergence of 1 harder, given the dependency of the

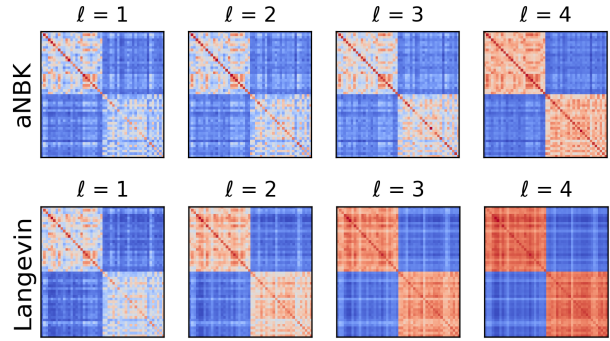


Figure 4. Theory vs empirical kernels at each layer for a 5HL MLP with $\phi(h) \equiv \tanh(h)$ learning $P = 50$ patterns of MNIST with $\mathbf{y} = \{\pm 1\}^P$ labels. Alignments $\mathcal{A}(\Phi^\ell, \Phi_{\text{exp}}^\ell) = \frac{\text{Tr}(\Phi^\ell \Phi_{\text{exp}}^\ell)}{\|\Phi^\ell\| \|\Phi_{\text{exp}}^\ell\|}$ between theory and empirical kernels are $\mathcal{A}(\Phi^1, \Phi_{\text{exp}}^1) = 97\%$, $\mathcal{A}(\Phi^2, \Phi_{\text{exp}}^2) = 81\%$, $\mathcal{A}(\Phi^3, \Phi_{\text{exp}}^3) = 73\%$ and $\mathcal{A}(\Phi^4, \Phi_{\text{exp}}^4) = 89\%$.

non-Gaussian single-site density of Eq. (7) from the layer index ℓ .

5.1. CNNs

DMFT for infinite-width CNNs under gradient-flow was previously derived in (Bordelon & Pehlevan, 2022), which we solve numerically here for the first time. In Fig. 5 we show comparisons of DMFT kernel predictors at convergence for a two-layer MLP and a two-layer CNN with kernel size $k = 8$ and stride 8. Black dashed curves, which are the theories, closely match the full colored lines, which are the network predictors on $N = 1028$ width networks. CNN outperform the MLP at large sample size P for the same γ_0 .

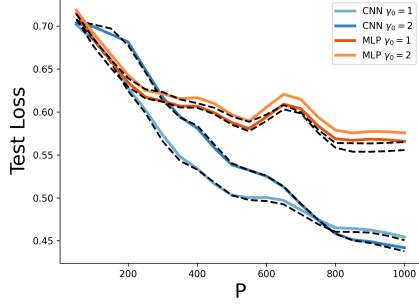


Figure 5. Test Loss as a function of sample size P for DMFT theories at convergence: two-layer MLP vs two-layer CNN trained on P animate/inanimate data on CIFAR10. Dashed lines are for theory, full-colored curves for empirical kernels.

See Appendix E.1 and Figure 5 for simulations of adaptive convolutional kernels derived from the feature-learning setting.

6. Discussion

In this paper, we develop a theory of non-parametric feature kernel predictors for MLP and CNN architectures in μP /mean-field parameterization. By analyzing gradient flow dynamics with weight decay and/or white noise, we identify two distinct infinite-width adaptive kernel predictors: aNBK, representing a Bayesian description of DNNs, and aNTK, corresponding to the fixed points of gradient flow with weight decay (Bordelon & Pehlevan, 2022). Unlike static NNGP (Neal, 1995; Lee et al., 2018) and NTK (Jacot et al., 2020) predictors, our kernels adapt to data. The feature learning strength, controlled by γ_0 , recovers lazy training when $\gamma_0 \rightarrow 0$ and enables richer representations for $\gamma_0 > 0$ (Bordelon et al., 2024). We study their impact across architectures and benchmark tasks on real datasets.

We also analyze infinitely wide deep linear networks in the feature-learning regime, where the saddle point equations simplify. Assuming a white data covariance matrix, the order parameters reduce to scalar overlaps c_ℓ between kernels and labels at each layer ℓ , exhibiting an exponential dependence on ℓ . This implies that for fixed γ_0 , deep linear networks align last-layer kernels first and propagate alignment backward. We derive scaling laws for these overlaps in the lazy, large-width, and large-depth regimes.

Our numerical results show that our adaptive kernels outperform NNGP and NTK at large sample size and match the performance of a trained NN in the feature-learning regime even in moderate widths (e.g. $N = 5000$). Our theory predicts non-Gaussian pre-activation densities at convergence and data-clustered feature kernels, whose alignment with label covariance increases with γ_0 .

Future work could focus on reducing solver computational costs by developing more efficient optimization techniques.

Acknowledgements

We thank Enrico M. Malatesta and Hugo Cui for useful and engaging discussions. B.B. is supported by a Google PhD Fellowship and NSF CAREER Award IIS-2239780. C.L. is supported by DARPA Award DIAL-FP-038, and The William F. Milton Fund from Harvard University. C.P. is supported by NSF grant DMS-2134157, NSF CAREER Award IIS-2239780, DARPA Award DIAL-FP-038, a Sloan Research Fellowship, and The William F. Milton Fund from Harvard University. 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.

Impact Statement

This paper presents work whose goal is to advance the theoretical understanding of deep neural networks. There are many computational/algorithmic consequences of our work, none which we feel must be specifically highlighted here.

References

- Advani, M. S. and Saxe, A. M. High-dimensional dynamics of generalization error in neural networks, 2017. URL <https://arxiv.org/abs/1710.03667>.
- Aitchison, L. Why bigger is not always better: on finite and infinite neural networks, 2020. URL <https://arxiv.org/abs/1910.08013>.
- Aiudi, R., Pacelli, R., Vezzani, A., Burioni, R., and Rotonondo, P. Local kernel renormalization as a mechanism for feature learning in overparametrized convolutional neural networks. *Nature Communications*, 16, 2023. URL <https://api.semanticscholar.org/CorpusID:260125263>.
- Arora, S., Cohen, N., Golowich, N., and Hu, W. A convergence analysis of gradient descent for deep linear neural networks, 2019a. URL <https://arxiv.org/abs/1810.02281>.
- Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R., and Wang, R. On exact computation with an infinitely wide neural net, 2019b. URL <https://arxiv.org/abs/1904.11955>.
- Arous, G. B., Dembo, A., and Guionnet, A. Cugliandolo-kurchan equations for dynamics of spin-glasses, 2004. URL <https://arxiv.org/abs/math/0409273>.

- Atanasov, A., Bordelon, B., and Pehlevan, C. Neural networks as kernel learners: The silent alignment effect, 2021. URL <https://arxiv.org/abs/2111.00034>.
- Baglioni, P., Pacelli, R., Aiudi, R., Di Renzo, F., Vezzani, A., Burioni, R., and Rotondo, P. Predictive power of a bayesian effective action for fully connected one hidden layer neural networks in the proportional limit. *Phys. Rev. Lett.*, 133:027301, Jul 2024. doi: 10.1103/PhysRevLett.133.027301. URL <https://link.aps.org/doi/10.1103/PhysRevLett.133.027301>.
- Basseti, F., Gherardi, M., Ingrosso, A., Pastore, M., and Rotondo, P. Feature learning in finite-width bayesian deep linear networks with multiple outputs and convolutional layers, 2024. URL <https://arxiv.org/abs/2406.03260>.
- Bordelon, B. and Pehlevan, C. Self-consistent dynamical field theory of kernel evolution in wide neural networks, 2022. URL <https://arxiv.org/abs/2205.09653>.
- Bordelon, B. and Pehlevan, C. Dynamics of finite width kernel and prediction fluctuations in mean field neural networks. *Advances in Neural Information Processing Systems*, 36, 2024.
- Bordelon, B., Atanasov, A., and Pehlevan, C. How feature learning can improve neural scaling laws, 2024. URL <https://arxiv.org/abs/2409.17858>.
- Bradbury, J., Frostig, R., Hawkins, P., Johnson, M. J., Leary, C., Maclaurin, D., Necula, G., Paszke, A., VanderPlas, J., Wanderman-Milne, S., and Zhang, Q. JAX: composable transformations of Python+NumPy programs, 2018. URL <http://github.com/google/jax>.
- Chizat, L., Oyallon, E., and Bach, F. On lazy training in differentiable programming, 2020. URL <https://arxiv.org/abs/1812.07956>.
- Cho, Y. and Saul, L. Kernel methods for deep learning. In Bengio, Y., Schuurmans, D., Lafferty, J., Williams, C., and Culotta, A. (eds.), *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009a. URL https://proceedings.neurips.cc/paper_files/paper/2009/file/5751ec3e9a4feab575962e78e006250d-Paper.pdf.
- Cho, Y. and Saul, L. Kernel methods for deep learning. *Advances in neural information processing systems*, 22, 2009b.
- Cui, H., Krzakala, F., and Zdeborová, L. Bayes-optimal learning of deep random networks of extensive-width. In *International Conference on Machine Learning*, pp. 6468–6521. PMLR, 2023.
- De Dominicis, C. Dynamics as a substitute for replicas in systems with quenched random impurities. *Phys. Rev. B*, 18:4913–4919, Nov 1978. doi: 10.1103/PhysRevB.18.4913. URL <https://link.aps.org/doi/10.1103/PhysRevB.18.4913>.
- de G. Matthews, A. G., Rowland, M., Hron, J., Turner, R. E., and Ghahramani, Z. Gaussian process behaviour in wide deep neural networks, 2018. URL <https://arxiv.org/abs/1804.11271>.
- Domingos, P. Every model learned by gradient descent is approximately a kernel machine. *arXiv preprint arXiv:2012.00152*, 2020.
- Fischer, K., Lindner, J., Dahmen, D., Ringel, Z., Krämer, M., and Helias, M. Critical feature learning in deep neural networks, 2024. URL <https://arxiv.org/abs/2405.10761>.
- Geiger, M., Spigler, S., Jacot, A., and Wyart, M. Disentangling feature and lazy training in deep neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(11):113301, November 2020. ISSN 1742-5468. doi: 10.1088/1742-5468/abc4de. URL <http://dx.doi.org/10.1088/1742-5468/abc4de>.
- Hanin, B. and Zlokapa, A. Bayesian interpolation with deep linear networks. *Proceedings of the National Academy of Sciences*, 120(23):e2301345120, 2023.
- Hestness, J., Narang, S., Ardalani, N., Diamos, G., Jun, H., Kianinejad, H., Patwary, M. M. A., Yang, Y., and Zhou, Y. Deep learning scaling is predictable, empirically. *arXiv preprint arXiv:1712.00409*, 2017.
- Hoffmann, J., Borgeaud, S., Mensch, A., Buchatskaya, E., Cai, T., Rutherford, E., Casas, D. d. L., Hendricks, L. A., Welbl, J., Clark, A., et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.
- Ingrosso, A., Pacelli, R., Rotondo, P., and Gerace, F. Statistical mechanics of transfer learning in fully connected networks in the proportional limit. *Physical Review Letters*, 134(17):177301, 2025.
- Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks, 2020. URL <https://arxiv.org/abs/1806.07572>.

- Jacot, A., Ged, F., Şimşek, B., Hongler, C., and Gabriel, F. Saddle-to-saddle dynamics in deep linear networks: Small initialization training, symmetry, and sparsity, 2022. URL <https://arxiv.org/abs/2106.15933>.
- Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J., and Amodei, D. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.
- Kardar, M. *Statistical physics of particles*. Cambridge University Press, 2007.
- Lee, J., Bahri, Y., Novak, R., Schoenholz, S. S., Pennington, J., and Sohl-Dickstein, J. Deep neural networks as gaussian processes, 2018. URL <https://arxiv.org/abs/1711.00165>.
- Lee, J., Xiao, L., Schoenholz, S. S., Bahri, Y., Novak, R., Sohl-Dickstein, J., and Pennington, J. Wide neural networks of any depth evolve as linear models under gradient descent *. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(12):124002, December 2020. ISSN 1742-5468. doi: 10.1088/1742-5468/abc62b. URL <http://dx.doi.org/10.1088/1742-5468/abc62b>.
- Lewkowycz, A. and Gur-Ari, G. On the training dynamics of deep networks with l_2 regularization. *Advances in Neural Information Processing Systems*, 33:4790–4799, 2020.
- Li, Q. and Sompolinsky, H. Statistical mechanics of deep linear neural networks: The backpropagating kernel renormalization. *Physical Review X*, 11(3), September 2021. ISSN 2160-3308. doi: 10.1103/physrevx.11.031059. URL <http://dx.doi.org/10.1103/PhysRevX.11.031059>.
- Li, Q. and Sompolinsky, H. Globally gated deep linear networks. *Advances in Neural Information Processing Systems*, 35:34789–34801, 2022.
- Mallinar, N., Beaglehole, D., Zhu, L., Radhakrishnan, A., Pandit, P., and Belkin, M. Emergence in non-neural models: grokking modular arithmetic via average gradient outer product. *arXiv preprint arXiv:2407.20199*, 2024.
- Martin, P. C., Siggia, E. D., and Rose, H. A. Statistical dynamics of classical systems. *Phys. Rev. A*, 8:423–437, Jul 1973. doi: 10.1103/PhysRevA.8.423. URL <https://link.aps.org/doi/10.1103/PhysRevA.8.423>.
- Mei, S., Montanari, A., and Nguyen, P.-M. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33):E7665–E7671, 2018. doi: 10.1073/pnas.1806579115. URL <https://www.pnas.org/doi/abs/10.1073/pnas.1806579115>.
- Mingard, C., Valle-Pérez, G., Skalse, J., and Louis, A. A. Is sgd a bayesian sampler? well, almost, 2020. URL <https://arxiv.org/abs/2006.15191>.
- Naveh, G. and Ringel, Z. A self consistent theory of gaussian processes captures feature learning effects in finite cnns. *Advances in Neural Information Processing Systems*, 34:21352–21364, 2021.
- Naveh, G., Ben David, O., Sompolinsky, H., and Ringel, Z. Predicting the outputs of finite deep neural networks trained with noisy gradients. *Physical Review E*, 104(6), December 2021. ISSN 2470-0053. doi: 10.1103/physrev.104.064301. URL <http://dx.doi.org/10.1103/PhysRevE.104.064301>.
- Neal, R. M. *Bayesian learning for neural networks*. PhD thesis, CAN, 1995. AAINN02676.
- Novak, R., Bahri, Y., Abolafia, D. A., Pennington, J., and Sohl-Dickstein, J. Sensitivity and generalization in neural networks: an empirical study, 2018. URL <https://arxiv.org/abs/1802.08760>.
- Pacelli, R., Ariosto, S., Pastore, M., Ginelli, F., Gherardi, M., and Rotondo, P. A statistical mechanics framework for bayesian deep neural networks beyond the infinite-width limit. *Nature Machine Intelligence*, 5(12): 1497–1507, December 2023. ISSN 2522-5839. doi: 10.1038/s42256-023-00767-6. URL <http://dx.doi.org/10.1038/s42256-023-00767-6>.
- Radhakrishnan, A., Beaglehole, D., Pandit, P., and Belkin, M. Mechanism of feature learning in deep fully connected networks and kernel machines that recursively learn features. *arXiv preprint arXiv:2212.13881*, 2022.
- Rasmussen, C. E. and Williams, C. K. I. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.
- Roberts, D. A., Yaida, S., and Hanin, B. *The principles of deep learning theory*, volume 46. Cambridge University Press Cambridge, MA, USA, 2022.
- Rotskoff, G. and Vanden-Eijnden, E. Trainability and accuracy of artificial neural networks: An interacting particle system approach. *Communications on Pure and Applied Mathematics*, 75(9):1889–1935, 2022.
- Rubin, N., Ringel, Z., Seroussi, I., and Helias, M. A unified approach to feature learning in bayesian neural networks. In *High-dimensional Learning Dynamics 2024: The Emergence of Structure and Reasoning*, 2024a. URL <https://openreview.net/forum?id=ZmOSJ2MV2R>.

- Rubin, N., Seroussi, I., and Ringel, Z. Grokking as a first order phase transition in two layer networks. In *The Twelfth International Conference on Learning Representations*, 2024b. URL <https://openreview.net/forum?id=3ROGsTX3IR>.
- Saxe, A. M., McClelland, J. L., and Ganguli, S. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, 2014. URL <https://arxiv.org/abs/1312.6120>.
- Scholkopf, B. and Smola, A. J. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001. ISBN 0262194759.
- Seroussi, I., Naveh, G., and Ringel, Z. Separation of scales and a thermodynamic description of feature learning in some cnns. *Nature Communications*, 14(1):908, 2023.
- Shan, H., Li, Q., and Sompolinsky, H. Order parameters and phase transitions of continual learning in deep neural networks. *arXiv preprint arXiv:2407.10315*, 2024.
- Sompolinsky, H. and Zippelius, A. Dynamic theory of the spin-glass phase. *Phys. Rev. Lett.*, 47: 359–362, Aug 1981. doi: 10.1103/PhysRevLett.47.359. URL <https://link.aps.org/doi/10.1103/PhysRevLett.47.359>.
- van Meegen, A. and Sompolinsky, H. Coding schemes in neural networks learning classification tasks, 2024. URL <https://arxiv.org/abs/2406.16689>.
- Vyas, N., Bansal, Y., and Nakkiran, P. Limitations of the ntk for understanding generalization in deep learning. *arXiv preprint arXiv:2206.10012*, 2022.
- Vyas, N., Atanasov, A., Bordelon, B., Morwani, D., Sainathan, S., and Pehlevan, C. Feature-learning networks are consistent across widths at realistic scales, 2023. URL <https://arxiv.org/abs/2305.18411>.
- Wei, A., Hu, W., and Steinhardt, J. More than a toy: Random matrix models predict how real-world neural representations generalize, 2022. URL <https://arxiv.org/abs/2203.06176>.
- Welling, M. and Teh, Y. W. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th International Conference on International Conference on Machine Learning, ICML’11*, pp. 681–688, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195.
- Yang, A. X., Robeyns, M., Milsom, E., Anson, B., Schoots, N., and Aitchison, L. A theory of representation learning gives a deep generalisation of kernel methods. In *International Conference on Machine Learning*, pp. 39380–39415. PMLR, 2023.
- Yang, G. and Hu, E. J. Feature learning in infinite-width neural networks, 2022. URL <https://arxiv.org/abs/2011.14522>.
- Yang, G., Hu, E. J., Babuschkin, I., Sidor, S., Liu, X., Farhi, D., Ryder, N., Pachocki, J., Chen, W., and Gao, J. Tensor programs v: Tuning large neural networks via zero-shot hyperparameter transfer, 2022. URL <https://arxiv.org/abs/2203.03466>.
- Zavatone-Veth, J. A., Canatar, A., Ruben, B. S., and Pehlevan, C. Asymptotics of representation learning in finite bayesian neural networks*. *Journal of Statistical Mechanics: Theory and Experiment*, 2022(11): 114008, November 2022a. ISSN 1742-5468. doi: 10.1088/1742-5468/ac98a6. URL <http://dx.doi.org/10.1088/1742-5468/ac98a6>.
- Zavatone-Veth, J. A., Tong, W. L., and Pehlevan, C. Contrasting random and learned features in deep bayesian linear regression. *Physical Review E*, 105(6):064118, 2022b.

A. Multi-layer deep Bayesian MLPs

As mentioned in the main text, we would like to study feature learning when the solution space is sampled from a posterior that is a Gibbs distribution with a likelihood $\mathcal{L}(\theta; \mathcal{D})$ and a Gaussian prior $\frac{\lambda}{2} \|\theta\|^2$ and by assuming width $N_\ell = N \forall \ell \in \{L\}$. We can do this computation for generic loss and non-linear activation functions as Eq. (2) shows. The representer theorem for Bayesian network is indeed independent on the activation choice. Here, the posterior takes the form

$$p(\theta|\mathcal{D}) = \frac{1}{Z} \exp \left[-\beta \gamma^2 \mathcal{L}(\theta; \mathcal{D}) - \sum_{\ell=0}^L \frac{\lambda_\ell}{2} \|\theta^\ell\|^2 \right]. \quad (18)$$

being $\theta = \text{Vec}\{\mathbf{W}^{(0)}, \dots, \mathbf{w}^{(L)}\}$ the collection of weights, $\mathcal{D} = \{\mathbf{x}_\mu, y_\mu\}_{\mu=1}^P$ the dataset with P patterns, and $\beta = \frac{1}{T}$ the temperature inverse. Again, we are interested in the infinitely overparameterized limit where $N \rightarrow \infty, P = \mathcal{O}_N(1)$. Here, when $\beta \rightarrow \infty$ the posterior becomes the uniform distribution over the set of global minimizers $\theta^* \in \arg \min_{\theta} \mathcal{L}(\theta)$. In

this setting, one needs to rescale the loss function $\mathcal{L} \rightarrow \gamma^2 \mathcal{L}$ with $\gamma = \gamma_0 \sqrt{N}$ in order to avoid for the Gaussian prior to dominate over the likelihood when $N \rightarrow \infty$, suppressing any interaction with the learning task $\mathcal{D}$. From the normalization factor in Eq. (18), the partition function reads

$$Z = \int \prod_{\ell=0}^L d\mathbf{W}^\ell e^{-\frac{\beta}{2} \gamma_0^2 N \sum_{\mu} \mathcal{L}(y^\mu, f^\mu) - \sum_{\ell=0}^L \frac{\lambda_\ell}{2} \|\mathbf{W}^\ell\|^2} \quad (19)$$

and since we consider the dataset as fixed, we wish to integrate out the weights and move to a description in the space of representations. This can be done by simply enforcing the definitions of Eq. (20) through the integral representations of some Dirac-delta functions

$$\int \prod_{\mu, \ell} d\mathbf{h}_\mu^{\ell+1} ds_\mu \left(\prod_{\mu, \ell} \delta \left(\mathbf{h}_\mu^{\ell+1} - \frac{1}{\sqrt{N}} \mathbf{W}^\ell \phi(\mathbf{h}_\mu^\ell) \right) \prod_{\mu} \delta \left(s_\mu - \frac{1}{\gamma_0 N} \mathbf{w}^{(L)} \cdot \phi(\mathbf{h}_\mu^L) \right) \right)_{\theta \sim \mathcal{N}(0, \Lambda^{-1})} \quad (20)$$

where $\Lambda = \text{diag}(\lambda_0 \mathbf{I}, \lambda_1 \mathbf{I}, \dots, \lambda_L \mathbf{I})$, getting

$$\begin{aligned} Z &= \int \prod_{\mu} \prod_{\ell=0}^{L-1} \frac{d\mathbf{h}_\mu^{\ell+1} d\hat{\mathbf{h}}_\mu^{\ell+1}}{2\pi} \int \prod_{\mu} \frac{ds_\mu d\hat{s}_\mu}{2\pi(\gamma_0 N)^{-1}} e^{i \sum_{\mu, \ell} \mathbf{h}_\mu^{\ell+1} \cdot \hat{\mathbf{h}}_\mu^{\ell+1} - \frac{1}{2} \sum_{\mu, \nu} \sum_{\ell} (\hat{\mathbf{h}}_\mu^{\ell+1} \cdot \hat{\mathbf{h}}_\nu^{\ell+1}) \left(\frac{\phi(\mathbf{h}_\mu^\ell) \cdot \phi(\mathbf{h}_\nu^\ell)}{N \lambda_\ell} \right) + i \gamma_0 N \sum_{\mu} s_\mu \hat{s}_\mu} \\ &\quad \times e^{-\frac{1}{2} \sum_{\mu, \nu} \hat{s}_\mu \hat{s}_\nu \left(\frac{\phi(\mathbf{h}_\mu^L) \cdot \phi(\mathbf{h}_\nu^L)}{\lambda_L} \right) - \frac{\beta}{2} N \gamma_0^2 \sum_{\mu} \mathcal{L}(y^\mu, \sigma(s^\mu))} \\ &= \int \prod_{\mu\nu} \prod_{\ell=1}^L \frac{d\Phi_{\mu\nu}^\ell d\hat{\Phi}_{\mu\nu}^\ell}{2\pi N^{-1}} \int \prod_{\mu} \frac{ds_\mu d\hat{s}_\mu}{2\pi(\gamma_0 N)^{-1}} e^{\frac{N}{2} \sum_{\mu\nu} \sum_{\ell} \Phi_{\mu\nu}^\ell \hat{\Phi}_{\mu\nu}^\ell - \gamma_0 N \sum_{\mu} s_\mu \hat{s}_\mu + \frac{N}{2} \hat{s}_\mu \frac{\Phi_{\mu\nu}^L}{\lambda_L} \hat{s}_\nu - \frac{\beta}{2} N \gamma_0^2 \sum_{\mu} \mathcal{L}(y_\mu, \sigma(s_\mu))} \\ &\quad \times e^{N \sum_{\ell=0}^{L-1} \ln \mathcal{Z}[\Phi_{\mu\nu}^{\ell+1}, \hat{\Phi}_{\mu\nu}^{\ell+1}]}. \end{aligned} \quad (21)$$

In the last expression, we introduced the adaptive feature kernels as

$$\Phi_{\mu\nu}^\ell = \frac{1}{N} \phi(\mathbf{h}_\mu^\ell) \cdot \phi(\mathbf{h}_\nu^\ell) \quad (22)$$

and, again, we enforced their definitions in Z with some conjugated variables $\hat{\Phi}_{\mu\nu}^\ell$. Both $\{\Phi_{\mu\nu}^\ell, \hat{\Phi}_{\mu\nu}^\ell\}$ will become deterministic quantities in the $N \rightarrow \infty$ limit.

The single-site density in Eq. (21) is given by

$$\begin{aligned} \mathcal{Z}_\ell &= \int \prod_{\mu} \frac{d\mathbf{h}_\mu d\hat{\mathbf{h}}_\mu}{2\pi} e^{i \sum_{\mu} \mathbf{h}_\mu \hat{\mathbf{h}}_\mu - \frac{1}{2} \sum_{\mu\nu} \hat{\mathbf{h}}_\mu \frac{\Phi_{\mu\nu}^{\ell-1}}{\lambda_{\ell-1}} \hat{\mathbf{h}}_\nu - \frac{1}{2} \sum_{\mu\nu} \phi(\mathbf{h}_\mu) \hat{\Phi}_{\mu\nu}^\ell \phi(\mathbf{h}_\nu)} \\ &= \int \frac{\prod_{\mu} d\mathbf{h}_\mu}{\sqrt{2\pi \det \left(\frac{\Phi_{\ell-1}^{\ell-1}}{\lambda_{\ell-1}} \right)}} e^{-\frac{1}{2} \sum_{\mu\nu} \mathbf{h}_\mu \left[\left(\frac{\Phi_{\ell-1}^{\ell-1}}{\lambda_{\ell-1}} \right)^{-1} \right]_{\mu\nu} \mathbf{h}_\nu - \frac{1}{2} \sum_{\mu\nu} \phi(\mathbf{h}_\mu) \hat{\Phi}_{\mu\nu}^\ell \phi(\mathbf{h}_\nu)}. \end{aligned} \quad (23)$$

At each layer, this decouples over the neuron index because we supposed the hidden layers having the same width dimension N for $\ell = 1, \dots, L$, and represents the normalization factor of a non-Gaussian pre-activation density distribution $p(h_\mu^\ell)$ where the non-Gaussian part is proportional to $\hat{\Phi}^\ell$, while the Gaussian contribution has a covariance that is the feature kernel at previous layer $\Phi^{\ell-1}$.

In the large $N \rightarrow \infty$ limit, we can collect the partition function of Eq. (21) in the compact form

$$Z = \int \prod_{\mu\nu} \prod_{\ell=1}^L \frac{d\Phi_{\mu\nu}^\ell d\hat{\Phi}_{\mu\nu}^\ell}{2\pi N^{-1}} \int \prod_{\mu} \frac{ds_\mu d\hat{s}_\mu}{2\pi(\gamma_0 N)^{-1}} e^{-NS(\{\Phi^\ell, \hat{\Phi}_{\ell=1}^L\}_{\ell=1}^L)} \quad (24)$$

being $S(\{\Phi^\ell, \hat{\Phi}_{\ell=1}^L\}_{\ell=1}^L)$ the intensive Bayesian action

$$S = -\frac{1}{2} \sum_{\mu\nu} \sum_{\ell} \Phi_{\mu\nu}^\ell \hat{\Phi}_{\mu\nu}^\ell + \gamma_0 \sum_{\mu} s_\mu \hat{s}_\mu - \frac{1}{2} \sum_{\mu\nu} \hat{s}_\mu \frac{\Phi_{\mu\nu}^L}{\lambda_L} \hat{s}_\nu + \gamma_0^2 \frac{\beta}{2} \sum_{\mu} \mathcal{L}(y_\mu, \sigma(s_\mu)) - \sum_{\ell} \ln \mathcal{Z}. \quad (25)$$

At infinite width N , this partition function is exponentially dominated by the saddle points of S . We thus identify the kernels that make S locally stationary ($\delta S = 0$) by the equations

$$\frac{\partial S}{\partial \hat{\Phi}_{\mu\nu}^\ell} = 0 \quad \forall \ell = 1, \dots, L \quad (26a)$$

$$\frac{\partial S}{\partial \Phi_{\mu\nu}^\ell} = 0 \quad \forall \ell = 1, \dots, L-1 \quad (26b)$$

$$\frac{\partial S}{\partial \Phi_{\mu\nu}^L} = 0 \quad (26c)$$

$$\frac{\partial S}{\partial \hat{s}_\mu} = 0 \quad (26d)$$

$$\frac{\partial S}{\partial s_\mu} = 0 \quad (26e)$$

where the last two saddle points fix the output pre-activation given the loss function $\mathcal{L}$. If we explicitly write down the partial derivatives, we get

$$\Phi^\ell = \langle \phi(\mathbf{h}^\ell) \phi(\mathbf{h}^\ell)^\top \rangle \quad \forall \ell \in \{L\} \quad (27a)$$

$$\hat{\Phi}^\ell = (\Phi^\ell)^{-1} - \lambda_\ell (\Phi^\ell)^{-1} \langle \mathbf{h}^{\ell+1} \mathbf{h}^{\ell+1\top} \rangle (\Phi^\ell)^{-1} \quad \forall \ell \in \{L-1\} \quad (27b)$$

$$\hat{\Phi}^L = -\frac{1}{\lambda_L} \hat{\mathbf{s}} \hat{\mathbf{s}}^\top \quad (27c)$$

$$\hat{\mathbf{s}} = -\beta \gamma_0 \frac{\partial \mathcal{L}}{\partial \mathbf{s}} \quad (27d)$$

$$\mathbf{s} = \frac{1}{\gamma_0 \lambda_L} \Phi^L \hat{\mathbf{s}} \quad (27e)$$

from which we get a kernel predictor on a unseen test point $\mathbf{x}$, since

$$f(\mathbf{x}) = \sigma \left(\frac{\beta}{\lambda_L} \sum_{\mu=1}^P \Delta_\mu \Phi^L(\mathbf{x}_\mu, \mathbf{x}) \right) \quad (28)$$

being $\Delta_\nu = -\frac{\partial \mathcal{L}}{\partial s_\nu}$ the pattern error signal for that given loss.

A.1. Regression problem

In this specific case, where the form of the loss function is known $\mathcal{L} = \sum_{\mu=1}^P (y_\mu - f_\mu)^2$ and the readout is linear, i.e. $f(s_\mu) = s_\mu \quad \forall \mu \in \{P\}$, we can integrate over the output pre-activations and its conjugated parameter $\{s_\mu, \hat{s}_\mu\}$. After

integrating, we obtain

$$\begin{aligned}
 Z &= \int \prod_{\mu\nu} \prod_{\ell=1}^L \frac{d\Phi_{\mu\nu}^\ell d\hat{\Phi}_{\mu\nu}^\ell}{2\pi N^{-1}} e^{-NS(\{\Phi_{\mu\nu}^\ell, \hat{\Phi}_{\mu\nu}^\ell\}_{\ell=1}^L)} \\
 &= \int \prod_{\mu\nu} \prod_{\ell=1}^L \frac{d\Phi_{\mu\nu}^\ell d\hat{\Phi}_{\mu\nu}^\ell}{2\pi N^{-1}} e^{-\frac{N}{2} \sum_{\ell} \text{Tr}(\Phi^\ell \hat{\Phi}^\ell) + \frac{N}{2} \mathbf{y}^\top \left(\frac{\mathbf{I}}{\beta} + \frac{\Phi^L}{\lambda_L} \right)^{-1} \mathbf{y} - N \sum_{\ell=0}^{L-1} \ln \mathcal{Z}[\Phi^{\ell-1}, \hat{\Phi}^\ell]}
 \end{aligned} \tag{29}$$

where the important quantity to be extremized in the limit $N \rightarrow \infty$ is the intensive action

$$S(\Phi^\ell, \hat{\Phi}^\ell) = -\frac{1}{2} \sum_{\ell} \text{Tr}(\Phi^\ell \hat{\Phi}^\ell) + \frac{\gamma_0^2}{2} \mathbf{y}^\top \left(\frac{\mathbf{I}}{\beta} + \frac{\Phi^L}{\lambda_L} \right)^{-1} \mathbf{y} - \sum_{\ell=1}^{L-1} \ln \mathcal{Z}[\Phi^{\ell-1}, \hat{\Phi}^\ell]. \tag{30}$$

Here, the saddle points which render the action S locally stationary $\delta S = 0$ with respect to these $2L$ matrix order parameters can be collected as

$$\Phi^\ell = \langle \phi(\mathbf{h}^\ell) \phi(\mathbf{h}^\ell)^\top \rangle \quad \forall \ell = 1, \dots, L \tag{31a}$$

$$\hat{\Phi}^\ell = (\Phi^\ell)^{-1} - \lambda_\ell (\Phi^\ell)^{-1} \langle \mathbf{h}^{\ell+1} \mathbf{h}^{\ell+1\top} \rangle (\Phi^\ell)^{-1} \quad \forall \ell = 1, \dots, L-1 \tag{31b}$$

$$\hat{\Phi}^L = -\frac{\gamma_0^2}{\lambda_L} \left(\frac{\mathbf{I}}{\beta} + \frac{\Phi^L}{\lambda_L} \right)^{-1} \mathbf{y} \mathbf{y}^\top \left(\frac{\mathbf{I}}{\beta} + \frac{\Phi^L}{\lambda_L} \right)^{-1} \tag{31c}$$

being $\Phi_{\mu\nu}^0 = \frac{\mathbf{x}_\mu \cdot \mathbf{x}_\nu}{\lambda_0 D}$ the data matrix covariance in this notation.

Notice that the last layer dual's kernel $\hat{\Phi}_{\mu\nu}^L$ vanishes in the lazy limit $\gamma_0 \rightarrow 0$ and so do all the dual kernels at previous layers $\hat{\Phi}_{\mu\nu}^\ell = 0$, while for non-negligible γ_0 we see that the each hidden layer features are non-Gaussian from Eq. (23). Details on the numerical implementation of the numerical solver for Eqs. (23) can be found in Sec. G.2.

In general, solving the min-max problem for deep networks is hard, because solving $\max_{\hat{\Phi}} S$ (S is the action defined in Algorithm 1) means to find some values for the dual kernels $\{\hat{\Phi}^\ell\}_{\ell=1}^L$ which tilt the non-Gaussian measure in Eq. (7) at each layer ℓ . In Fig. 4 of the main text we solved the theory for Algorithm 1 by initializing the theoretical kernels $\{\Phi, \hat{\Phi}\}_{\ell=1}^L$ with a lazy guess as explained in Sec. G.2. However, an easier strategy for convergence is to initialize the solver Algorithm 1 with the empirical NN kernels $\{\Phi^\ell\}_{\ell=1}^L$ at convergence obtained from Langevin simulations and perturbed with a multiplicative Gaussian noise. In this way, one only needs to solve for the dual variables. This warm start allows faster convergence. This is what we did to produce Fig. 7. Here, we plot the kernel alignment $\mathcal{A}(\Phi^\ell, \mathbf{y} \mathbf{y}^\top) = \frac{\mathbf{y}^\top \Phi^\ell \mathbf{y}}{\|\mathbf{y} \mathbf{y}^\top\| \|\Phi^\ell\|}$, which is the cosine similarity between the kernel Φ^ℓ and the label covariance $\mathbf{y} \mathbf{y}^\top$ at each layer. We show that by increasing feature strength γ_0 , the alignments increase. As expected, the last layer kernel aligns first with the labels as γ_0 increases, followed by the previous ones. Fig. 7(b) shows that the perturbed kernels of the theory retrieve the final NN kernels as locally stable fixed points.

In Fig. 6 we show that the non Gaussian preactivation distribution, depending on the activation function chosen, can develop a multimodal profile, and so a Mexican-hat profile for high values of feature learning strength γ_0 .

A.2. Generalization error

Knowing the posterior distribution makes it easy to compute the test error on a new (unseen) example $(\mathbf{x}_0, y_0)$, which is defined as

$$\epsilon_g(\mathbf{x}_0, y_0) = \langle (y_0 - f_0(\mathbf{x}_0; \theta))^2 \rangle_{\theta \sim p(\theta|\mathcal{D})} \tag{32}$$

where the sampling measure corresponds to Eq. (18). Here, we can include with an index $\mu = 0$ the test pattern contribution, and just compute

$$\begin{aligned}
 \epsilon_g(\mathbf{x}_0, y_0) &= \frac{1}{Z} \int \prod_{\mu\nu=0}^P \prod_{\ell=1}^L \frac{d\tilde{\Phi}_{\mu\nu}^\ell d\hat{\Phi}_{\mu\nu}^\ell}{2\pi N^{-1}} \int \prod_{\mu=0}^P \frac{ds_\mu d\hat{s}_\mu}{2\pi(\gamma_0 N)^{-1}} e^{\sum_{\mu\nu=0}^P \sum_{\ell=1}^L N_\ell \tilde{\Phi}_{\mu\nu}^\ell \hat{\Phi}_{\mu\nu}^\ell + \frac{N_L}{2\lambda_L} \sum_{\mu,\nu=1}^P \Phi_{\mu\nu}^L \hat{s}_\mu \hat{s}_\nu + \frac{N_L}{2\lambda_L} \Phi_{00}^L (\hat{s}_0)^2} \times \\
 &\quad \times e^{\frac{N_L}{\lambda_L} \hat{s}_0 \sum_{\mu} \Phi_{\mu} \hat{s}_\mu - N_L \sum_{\mu=0}^P s_\mu \hat{s}_\mu - \frac{\beta}{2} N \sum_{\mu=1}^P (y^\mu - s^\mu)^2 + N \sum_{\ell=0}^L \ln \tilde{\mathcal{Z}}_\ell} \times (y_0 - s_0)^2
 \end{aligned} \tag{33}$$

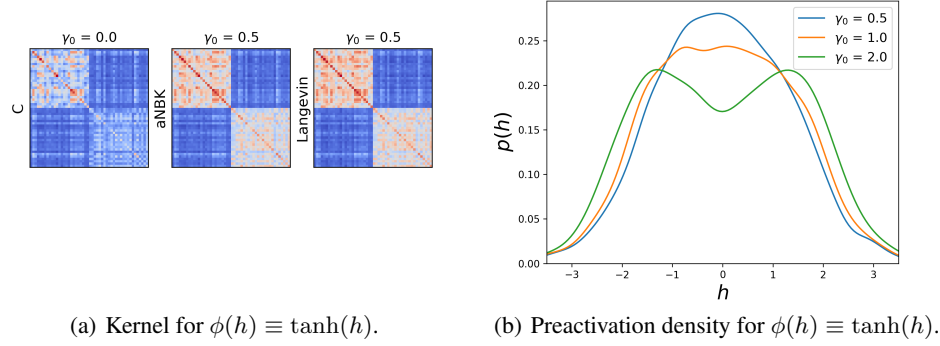


Figure 6. (a) Bayesian theory (aNBK) and empirical (Langevin) adaptive kernels with feature learning strength γ_0 and for $P = 50$ patterns of 0/1 classes of MNIST. C is the Gram matrix of data. (b) The preactivation distribution is in general non-Gaussian at each value of feature strength γ_0 . For activation $\phi(h) \equiv \tanh(h)$ we show that $p(h)$ can develop a Mexican-hat profile.

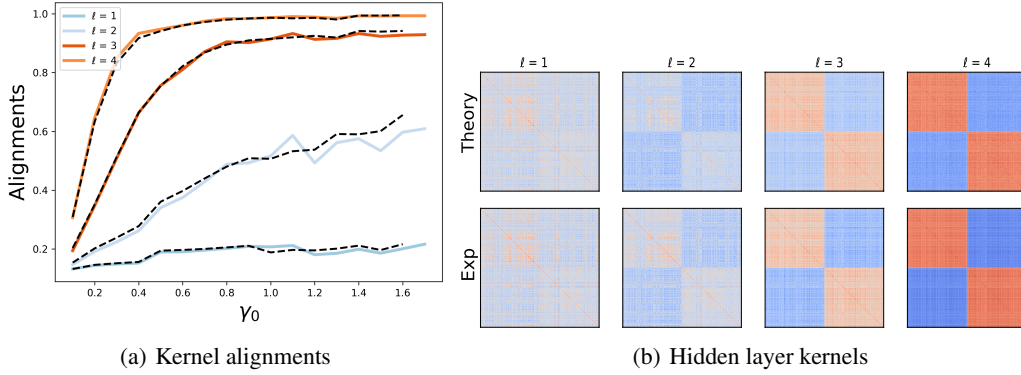


Figure 7. Bayesian $L = 5$ ReLU MLP trained on $P = 1000$ data of CIFAR10. Colored curve are experiments, dashed lines are the predictors calculated from Algorithm 1. (a) Kernel alignments $\mathcal{A}(\Phi^\ell, \mathbf{y}\mathbf{y}^\top)$ at each layer vs feature strength. (b) Theory vs empirical kernels at each layer. Here, theory refers to the intermediate layer kernels calculated via Algorithm 1.

where the single-site action contains now all the possible interactions with the test point at each layer ℓ in the test-test kernel $\Phi_{00}^\ell = \langle \phi(h_0)^2 \rangle$ and the test-train kernel $\Phi_\mu^\ell = \langle \phi(h_0)\phi(h_\mu) \rangle$

$$\begin{aligned} \tilde{Z} = & \int \prod_{\mu=0}^P \frac{dh_\mu d\hat{h}_\mu}{2\pi} e^{i \sum_{\mu=0}^P h_\mu \hat{h}_\mu - \frac{1}{2} \sum_{\mu,\nu=1}^P \hat{h}_\mu \frac{\Phi_{\mu\nu}^{\ell-1}}{\lambda_{\ell-1}} \hat{h}_\nu - \sum_{\mu,\nu=1}^P \phi(h_\mu) \hat{\Phi}_{\mu\nu}^\ell \phi(h_\nu) - \frac{1}{2} \frac{\Phi_{00}^{\ell-1}}{\lambda_{\ell-1}} \hat{h}_0^2 - \hat{\Phi}_{00}^\ell \phi(h_0)^2 - \sum_{\mu=1}^P \frac{\Phi_{\mu}^{\ell-1}}{\lambda_{\ell-1}} \hat{h}_0 \hat{h}_\mu} \\ & \times e^{-\sum_{\mu=1}^P \hat{\Phi}_\mu^\ell \phi(h_0) \phi(h_\mu)}. \end{aligned} \quad (34)$$

Exploiting the saddle point equations, we realize that the dual kernels concerning the test point $\hat{\Phi}_\mu^\ell$, $\hat{\Phi}_{00} = 0$, and that

$$\Phi_{\mu\nu}^\ell = \langle \phi(h_\mu) \phi(h_\nu) \rangle \quad \forall \ell = 1, \dots, L \quad (35a)$$

$$\hat{\Phi}_{\mu\nu}^\ell = \frac{1}{2\lambda_\ell} \langle \hat{h}_\mu^{\ell+1} \hat{h}_\nu^{\ell+1} \rangle \quad \forall \ell = 1, \dots, L-1 \quad (35b)$$

$$\hat{\Phi}_{\mu\nu}^L + \frac{1}{2\lambda_L} \hat{s}_\mu \hat{s}_\nu = 0 \quad (35c)$$

$$\hat{\Phi}_\mu^\ell = \frac{1}{\lambda_\ell} \langle \hat{h}_0^{\ell+1} \hat{h}_\mu^{\ell+1} \rangle \quad \forall \ell = 1, \dots, L-1 \quad (35d)$$

$$\hat{s}_0 = \hat{\Phi}_\mu^\ell = \hat{\Phi}_{00} = 0 \quad (35e)$$

$$s_0 = \frac{1}{\lambda_L} \sum_\mu \Phi_\mu^L \hat{s}_\mu \quad (35f)$$

$$\hat{s}_\mu - \beta(y_\mu - s_\mu) = 0 \quad (35g)$$

$$s_\mu = \frac{1}{\lambda_L} \sum_\nu \Phi_{\mu\nu}^L \hat{s}_\nu. \quad (35h)$$

This allows to rewrite the single site density in a much simpler form, where the non-Gaussian contribution includes just the train points, while the Gaussian part has a $(P+1) \times (P+1)$ covariance matrix $\tilde{\Phi} = \begin{pmatrix} \Phi_{00} & \Phi_\mu^\top \\ \Phi_\mu & \Phi_{\mu\nu} \end{pmatrix}$

$$\tilde{Z} = \int \frac{\prod_\mu dh_\mu}{\sqrt{2\pi \det \left(\frac{\tilde{\Phi}^{\ell-1}}{\lambda_{\ell-1}} \right)}} e^{-\frac{\lambda_{\ell-1}}{2} \sum_{\mu\nu=0}^P h_\mu (\tilde{\Phi}^{\ell-1})_{\mu\nu}^{-1} h_\nu - \frac{1}{2} \sum_{\mu\nu=1}^P \phi(h_\mu) \tilde{\Phi}_{\mu\nu}^\ell \phi(h_\nu)} \quad (36)$$

This means that once we solved for Eq. (31) we can marginalize Eq. (36) to get $p(h_0^\ell | \mathbf{h}^\ell)$ and hence the test-train vector kernel Φ_μ . The test error expression is

$$\epsilon_g(\mathbf{x}_0, y_0) = \left(y_0 - \frac{1}{\lambda_L} \Phi^L{}^\top \left[\frac{\Phi^L}{\lambda_L} + \frac{\mathbf{I}}{\beta} \right]^{-1} \mathbf{y} \right)^2 \quad (37)$$

meaning that the predictor $f(x) = \frac{1}{\lambda_L} \Phi^L(x)^\top \left[\frac{\Phi^L}{\lambda_L} + \frac{\mathbf{I}}{\beta} \right]^{-1} \mathbf{y}$ is again a kernel predictor, with adaptive kernels from Eqs. (31).

An alternative way to obtain the predictor in the specific regression setting from Eq. (28) and when $\sigma(s) = s$ is by solving for $\Delta_\mu \equiv -\frac{\partial L}{\partial s_\mu}$. In this case

$$\Delta_\mu = y_\mu - f_\mu = y_\mu - \frac{\beta}{\lambda} \sum_\nu \Delta_\nu \Phi_{\mu\nu} \quad (38)$$

and by solving this equation for $\Delta \in \mathbb{R}^P$ we obtain

$$\Delta = \left[\mathbf{I} + \frac{\beta}{\lambda} \Phi \right]^{-1} \mathbf{y}. \quad (39)$$

Lastly, combining this with the original Eq. (28), we get

$$f(x) = \Phi(x)^\top \left[\mathbf{I} + \frac{\beta}{\lambda} \Phi \right]^{-1} \mathbf{y} \quad (40)$$

as is it in the main text Eq. (5).

A.3. Deep linear case

In the deep linear case $\phi(h^\ell) = h^\ell$, the action of Eq. (30) gets simplified because the single-site density is now Gaussian, and this leads to

$$S(\{\Phi^\ell, \hat{\Phi}^\ell\}) = -\frac{1}{2} \sum_{\ell} \text{Tr}(\Phi^\ell \hat{\Phi}^\ell) + \frac{\gamma_0^2}{2} \mathbf{y}^\top \left(\frac{\mathbf{I}}{\beta} + \frac{\Phi^L}{\lambda_L} \right)^{-1} \mathbf{y} + \frac{1}{2} \sum_{\ell=1}^L \ln \det \left(\mathbf{I} + \frac{\Phi^{\ell-1}}{\lambda_{\ell-1}} \hat{\Phi}^\ell \right) \quad (41)$$

with the following saddle point equations

$$\Phi^\ell - \frac{\Phi^{\ell-1}}{\lambda_{\ell-1}} \left(\mathbf{I} + \frac{\Phi^{\ell-1}}{\lambda_{\ell-1}} \hat{\Phi}^\ell \right)^{-1} = 0 \quad \forall \ell = 1, \dots, L \quad (42a)$$

$$\hat{\Phi}^\ell - \frac{\hat{\Phi}^{\ell+1}}{\lambda_\ell} \left(\mathbf{I} + \frac{\Phi^\ell}{\lambda_\ell} \hat{\Phi}^{\ell+1} \right)^{-1} = 0 \quad \forall \ell = 1, \dots, L-1 \quad (42b)$$

$$\hat{\Phi}^L + \frac{\gamma_0^2}{\lambda_L} \left(\frac{\mathbf{I}}{\beta} + \frac{\Phi^L}{\lambda_L} \right)^{-1} \mathbf{y} \mathbf{y}^\top \left(\frac{\mathbf{I}}{\beta} + \frac{\Phi^L}{\lambda_L} \right)^{-1} = 0. \quad (42c)$$

In principle, this is a closed set of equations, which can be iteratively solved as mentioned in Sec. 4. If we choose input data that are whitened, with $\Phi^0 = \mathbf{K}^x = \mathbf{I}$ and label norm $|\mathbf{y}| = 1$, the equations can be simplified even further, since feature kernels at each layer Φ^ℓ only evolve in the rank-one direction $\mathbf{y} \mathbf{y}^\top$. This allows to define the variables $\{c_\ell, \hat{c}_\ell\}$ which are the overlaps with the label direction $\mathbf{y}$

$$\mathbf{y}^\top \Phi^\ell \mathbf{y} \equiv c^\ell, \quad \mathbf{y}^\top \hat{\Phi}^\ell \mathbf{y} \equiv \hat{c}^\ell. \quad (43)$$

In this setting, the reduced saddle point equations become

$$c_1 = \frac{1}{1 + \hat{c}_1}, \quad c_{\ell+1} = \frac{c_\ell}{1 + c_\ell \hat{c}_{\ell+1}} \quad (44)$$

$$\hat{c}_L = -\frac{\gamma^2}{(\beta^{-1} + c_L)^2}, \quad \hat{c}_\ell = \frac{\hat{c}_{\ell+1}}{1 + c_\ell \hat{c}_{\ell+1}} \quad (45)$$

which are $2L$ scalar equations for the overlaps c_ℓ at each layer. We note the following conservation

$$c_\ell \hat{c}_\ell = c_{\ell+1} \hat{c}_{\ell+1} = -\frac{\gamma^2 c_L}{(\beta^{-1} + c_L)^2} \equiv \chi(c_L) \quad (46)$$

which implies that

$$1 = \frac{c_\ell}{c_{\ell+1} + c_\ell \chi(c_L)} \implies c_{\ell+1} = c_\ell (1 - \chi(c_L)) = (1 - \chi(c_L))^\ell c_1. \quad (47)$$

Since we have $c_1 = \frac{1}{1 + \chi(c_1)} \implies c_1 = 1 - \chi(c_L)$, hence we find

$$c_\ell = (1 - \chi(c_L))^\ell \quad (48)$$

which means that here is an exponential dependence of the overlap on layer index. In practice, we can solve Eq. (48) for the last layer overlap c_L since $\chi(c_L)$ and then move backward in computing all the previous layer overlaps.

We can also extract the following small γ or small L asymptotics. Precisely, when L is fixed and $\gamma_0 \rightarrow 0$ we recover a perturbative feature learning regime

$$c_L \sim 1 + \frac{L\gamma^2}{c_L} \implies c_L \sim 1 + L\gamma^2, \quad \gamma^2 L \rightarrow 0 \quad (49)$$

where correction are $\mathcal{O}(\gamma^2 L)$. Similarly, at large γ with fixed L , which stands for a shallow but very rich regime, the overlaps scale as

$$c = [1 + \gamma^2 c^{-1}]^L \implies c \sim \gamma^{2L/(L+1)} \quad (50)$$

while, alternatively the large L asymptotics for a very deep network have the form

$$c^{1+1/L} = c + \gamma^2 \implies c \ln c \sim L\gamma^2 \implies c \sim \frac{L\gamma^2}{\ln(L\gamma^2)} \quad (51)$$

which we show to be predictive in Fig.3 of the main text.

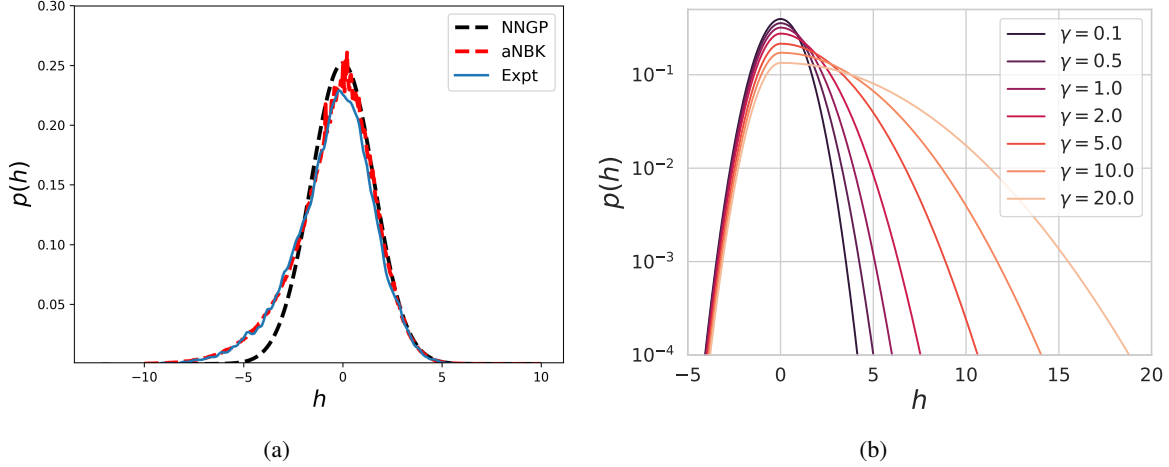


Figure 8. (a) Bayesian two-layer MLP trained on a whitened covariance matrix $\Phi^0 = \mathbf{I}$ on $P = 4$ train points. Feature learning ($\gamma_0 > 0$) leads to a non-Gaussian pre-activation distributions. Black-dashed curve is the lazy NNGP when $\mathbf{h} \sim \mathcal{N}(0, \Phi^0)$; red curve is the aNBK theory when $\mathbf{h}$ is sampled from Eq. 7; blue is the empirical pre-activation distribution of a $N = 5000$ network trained in the rich regime. (b) Non-gaussian pre-activation distribution as a function of feature learning strength γ for a Bayesian 2-layer MLP trained with Squared Error (SE) on 0-1 classes of MNIST dataset. Here sample size $P = 100$.

A.4. Non-gaussian pre-activation density

In the non-linear case, as Eq. (23) shows, when $\gamma_0 > 0$ the pre-activation density at each layer is non-Gaussian, with γ_0 entering in the saddle point equation for the dual kernel at the last layer $\hat{\Phi}^L$ (see Eq. (31)). This means that, once we have $\{\Phi^\ell, \hat{\Phi}^\ell\}_{\ell=1}^L$ from the solver (Alg 1), we can evaluate Eq. (23) with importance sampling and compute $p(h_\mu)$ for a given pattern μ . In Fig. A.4(a) we show that, while in the lazy regime where no feature learning enters, the hidden layer pre-activation of the NNGP predictors are Gaussian, this is not the case in our setting.

A.5. Perturbative approximation

In the $\gamma_0 \rightarrow 0$ limit, we recover the static kernels of NNGP predictor (Neal, 1995; Lee et al., 2018). Corrections to this lazy limit can be extracted at small but finite γ_0 . In order to do so, we can expand each macroscopic variable $q(\gamma_0)$ in power series of γ_0 , such as $q = q^{(0)} + \gamma_0^2 q^{(1)} + \gamma_0^4 q^{(2)} + \dots$, and compute the corrections up to $\mathcal{O}(\gamma_0^2)$. First of all, we notice that at leading order in γ_0^2

$$\hat{\Phi}^L = -\gamma_0^2 (\Phi_0^L + \beta^{-1})^{-1} \mathbf{y} \mathbf{y}^\top (\Phi_0^L + \beta^{-1})^{-1} \quad (52)$$

where we set each $\lambda_\ell = 1$ for clarity of notation. For each dual kernel at previous layer $\ell = 1, \dots, L-1$ we have instead a recursion

$$\frac{1}{2} \hat{\Phi}^\ell = -\frac{\partial}{\partial \Phi^\ell} \ln \mathcal{Z}(\Phi^\ell, \hat{\Phi}^{\ell+1}) \quad (53)$$

where non-perturbatively

$$\frac{1}{2} \hat{\Phi}^\ell = -\frac{1}{\mathcal{Z}} \int d\mathbf{h} \frac{\partial}{\partial \Phi^\ell} \exp \left(-\frac{1}{2} \mathbf{h} (\Phi^\ell)^{-1} \mathbf{h} - \frac{1}{2} \phi(\mathbf{h}) \hat{\Phi}^{\ell+1} \phi(\mathbf{h}) \right) \quad (54)$$

$$= \frac{1}{2} \frac{1}{\left\langle \exp \left(-\frac{1}{2} \phi(\mathbf{h}) \hat{\Phi}^{\ell+1} \phi(\mathbf{h}) \right) \right\rangle_0} \times \left\langle \frac{\partial^2}{\partial \mathbf{h}^2} \exp \left(-\frac{1}{2} \phi(\mathbf{h}) \hat{\Phi}^{\ell+1} \phi(\mathbf{h}) \right) \right\rangle_0 \quad (55)$$

being $\langle \cdot \rangle_0$ the Gaussian average with covariance Φ^ℓ . With a little bit of algebra the numerator can be written as

$$\begin{aligned} \left\langle \frac{\partial^2}{\partial h_\mu \partial h_\nu} \exp \left(-\frac{1}{2} \phi(h) \hat{\Phi} \phi(h) \right) \right\rangle_0 &= - \left\langle \frac{\partial}{\partial h_\nu} \left[\exp \left(-\frac{1}{2} \phi(h) \hat{\Phi} \phi(h) \right) \dot{\phi}(h_\mu) \hat{\Phi}_{\mu\alpha} \phi(h_\alpha) \right] \right\rangle_0 \\ &= \sum_{\alpha\beta} \hat{\Phi}_{\mu\alpha} \hat{\Phi}_{\nu\beta} \left\langle \dot{\phi}(h_\mu) \dot{\phi}(h_\nu) \phi(h_\alpha) \phi(h_\beta) \exp \left(-\frac{1}{2} \phi(h) \hat{\Phi} \phi(h) \right) \right\rangle_0 \\ &\quad - \delta_{\mu\nu} \sum_{\alpha} \hat{\Phi}_{\mu\alpha} \left\langle \ddot{\phi}(h_\mu) \phi(h_\alpha) \exp \left(-\frac{1}{2} \phi(h) \hat{\Phi} \phi(h) \right) \right\rangle_0 \\ &\quad - \left\langle \dot{\phi}(h_\mu) \dot{\phi}(h_\alpha) \exp \left(-\frac{1}{2} \phi(h) \hat{\Phi} \phi(h) \right) \right\rangle_0 \hat{\Phi}_{\mu\alpha}. \end{aligned} \quad (56)$$

Under the leading order approximation, we find the following relationship between successive layers

$$\hat{\Phi}^\ell \sim \frac{1}{2} \left\langle \frac{\partial^2}{\partial \mathbf{h} \partial \mathbf{h}^\top} \phi(\mathbf{h})^\top \hat{\Phi}^{\ell+1} \phi(\mathbf{h}) \right\rangle \quad (57)$$

The entries of this Hessian matrix can be computed in terms of derivatives of the activation function

$$\frac{\partial}{\partial h_\mu} \frac{\partial}{\partial h_\nu} \sum_{\alpha\beta} \phi(h_\alpha) \phi(h_\beta) \hat{\Phi} = \frac{\partial}{\partial h_\mu} \left[\dot{\phi}(h_\nu) \phi(h_\beta) \hat{\Phi}_{\nu\beta} + \dot{\phi}(h_\nu) \phi(h_\alpha) \hat{\Phi}_{\nu\alpha} \right] \quad (58)$$

$$= 2 \left\langle \dot{\phi}(h_\mu) \dot{\phi}(h_\nu) \right\rangle \hat{\Phi}_{\mu\nu} + 2 \delta_{\mu\nu} \left\langle \ddot{\phi}(h_\mu) \sum_{\beta} \phi(h_\beta) \right\rangle \hat{\Phi}_{\mu\beta} \quad (59)$$

all of these Gaussians can be evaluated at the unperturbed NNGP kernels Gaussian densities. Once these $\hat{\Phi}^\ell$ matrices have been computed, the Φ^ℓ matrices can be asymptotically approximated as

$$\Phi^\ell \sim \Phi_0^\ell - \frac{1}{2} \left\langle \phi(\mathbf{h}) \phi(\mathbf{h})^\top \left(\phi(\mathbf{h})^\top \hat{\Phi}^\ell \phi(\mathbf{h}) \right) \right\rangle_0 - \Phi_0^\ell \left\langle \phi(\mathbf{h})^\top \hat{\Phi}^\ell \phi(\mathbf{h}) \right\rangle_0 \quad (60)$$

which agree with the perturbative treatment of (Zavatone-Veth et al., 2022a) obtained in the NTK parameterization which is similar to our small γ_0 expansion (neglecting finite width fluctuations).

B. Expanded Related Works

In this section we provide additional details comparing our contributions with previous works. We break these up into (1) scale-renormalization theories (2) Adaptive Kernel Theories in the NTK scaling (3) and rescaling the likelihood.

B.1. Scale Renormalization Theories

Many theories of proportional limits of Bayesian neural networks predict that the mean predictor has the form

$$\langle f(x) \rangle = Q(\alpha) \sum_{\mu\nu} \Phi^L(x, x_\mu) [Q(\alpha) \Phi^L + \beta^{-1} \mathbf{I}]_{\mu\nu}^{-1} y_\nu = \sum_{\mu\nu} \Phi^L(x, x_\mu) [\Phi^L + \beta^{-1} Q(\alpha)^{-1} \mathbf{I}]_{\mu\nu}^{-1} y_\nu, \quad (61)$$

where $Q(\alpha) \in \mathbb{R}$ is a scalar that is determined self consistently as a function of $\alpha = P/N$ and $\Phi^L(x, x')$ is the final layer's NNGP kernel under the prior (Li & Sompolinsky, 2021; Pacelli et al., 2023). We rewrote this expression in the last line to emphasize that this is equivalent to a data dependent ridge with the original NNGP kernel. In the limit of low temperature $\beta \rightarrow \infty$ (zero regularization), these effects have no impact on the mean predictor. This theory does not capture adaptation in the entries of the kernel.

Extensions of this theory to CNNs, gated linear networks, and transfer learning result in slightly more order parameters which compute gate $\times$ gate overlaps for gated linear networks (Li & Sompolinsky, 2022), or space $\times$ space overlaps for CNNs (Aiudi et al., 2023), or task $\times$ task overlaps for transfer/continual learning (Shan et al., 2024; Ingrosso et al., 2025). While these theories are slightly more flexible, they do not allow for adaptation for the kernels to adapt within each block (gate, spatial location, task) but just reweight the blocks.

B.2. Adaptive Kernel Theories in NTK Scaling

Both Seroussi et al. (2023) and Fischer et al. (2024) derive mean field actions for the kernels and conjugate kernels for Bayesian networks under the posterior. Seroussi et al. (2023) use a variational Gaussian approximation to the preactivation density which finds the best Gaussian density which approximates the true posterior density for h . Fischer et al, explicitly describe how the joint distribution of kernels obeys a large deviation principle at large N (or in the proportional limit $P, N \rightarrow \infty$ with $P/N = \alpha$)

$$\frac{1}{N} \ln p(\{\Phi^\ell\}) \sim - \max_{\{\hat{\Phi}^\ell\}} S(\{\Phi^\ell, \hat{\Phi}^\ell\}) \quad (62)$$

where S is an action. However, in their theory the single-site distribution for hidden layer ℓ have the form

$$p_\ell(h) \propto \exp \left(-\frac{1}{2} \sum_{\mu\nu} h_\mu h_\nu [\Phi^{\ell-1}]_{\mu\nu}^{-1} - \frac{1}{2N} \sum_{\mu\nu} \phi(h_\mu) \phi(h_\nu) \hat{\Phi}_{\mu\nu}^\ell \right), \quad (63)$$

which reveal that the non-Gaussianity is explicitly suppressed with respect to N . As they focus on large width networks $N \gg 1$, they expand the single-site densities at leading order

$$p_\ell(h) \approx \exp \left(-\frac{1}{2} \sum_{\mu\nu} h_\mu h_\nu [\Phi^{\ell-1}]_{\mu\nu}^{-1} \right) \left[1 - \frac{1}{2N} \sum_{\mu\nu} \phi(h_\mu) \phi(h_\nu) \hat{\Phi}_{\mu\nu}^\ell \right] \quad (64)$$

which enables computation of *Gaussian* averages with a perturbed density.

Some key differences between our approach and this approach is that

- Our theory does not allow for linearization of the preactivation densities unless the richness hyperparameter γ_0 is explicitly made small. We therefore do not rely on linear response theory to solve our equations but rather confront the min-max problem directly.
- Our theory results in *deterministic kernels* rather than random matrices for the kernels $\Phi_{\mu\nu}^\ell$. The kernels arising from a proportional limit are random matrices. Reducing the noise from random initialization results in better performance as we discuss in Appendix C.

B.3. Feature Learning At Infinite Width By Rescaling Likelihood

An alternative approach, in analogy to mean-field/ μ P scaling in the dynamics case (Mei et al., 2018; Chizat et al., 2020; Geiger et al., 2020; Yang & Hu, 2022; Bordelon & Pehlevan, 2022), one could instead reparameterize the network definition so that feature learning persists as $N \rightarrow \infty$ even if P is kept constant. This approach was pursued for Bayesian networks by Yang et al. (2023). However, they do not reparameterize the definition of the network predictor so their saddle point equations for the kernels are not in agreement with ours. To see this concretely in the case of a one hidden layer linear networks. For direct comparison, the posterior kernels satisfy the following saddle point equations in our theory and theirs at zero temperature

$$\begin{cases} \Phi = C^{-1} - \gamma_0^2 \Phi^{-1} \mathbf{y} \mathbf{y}^\top \Phi^{-1} & \text{Ours} \\ 0 = C^{-1} - \Phi^{-1} \mathbf{y} \mathbf{y}^\top \Phi^{-1} & \text{Theirs} \end{cases} \quad (65)$$

where the second equation is provided in Equation 128 of Yang et al. (2023). We can obtain their result from our result by taking an ultra-rich limit $\gamma_0 \rightarrow \infty$ and rescaling the kernel as $\Phi = \gamma_0 \bar{\Phi}$ which gives

$$C^{-1} = \bar{\Phi}^{-1} \mathbf{y} \mathbf{y}^\top \bar{\Phi}^{-1}. \quad (66)$$

We thus see that our scaling allows for more intermediate behaviors between the ultra-rich $\gamma_0 \rightarrow \infty$ and the lazy learning $\gamma_0 \rightarrow 0$ limits. Further, Yang et al. (2023) utilize a different spline-based numerical method to approximate the preactivation density under the posterior, whereas we utilize importance sampling.

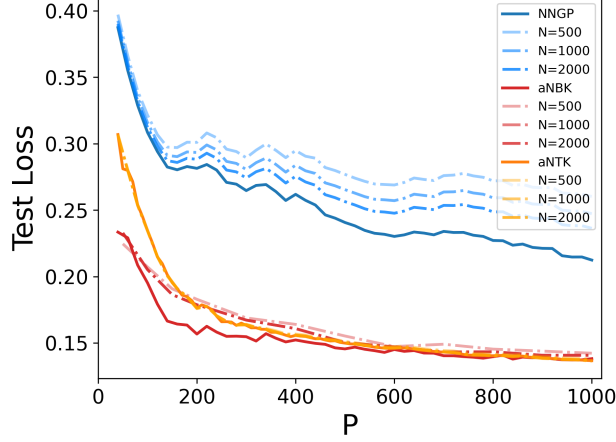


Figure 9. Finite-width effects in NTK and μP parameterizations. *Blue* curves are for different N values, showing that NNGP predictor is not consistent across network widths. We compare predictions of finite width N networks (as in (Pacelli et al., 2023)) to the NNGP infinite width limit. Finite width effects are instead negligible in μP parameterization, where finite N networks trained with Langevin consistently lie on the theory full *orange* and *red* curves.

C. Finite-width effects

In principle, in μP , the kernels $\Phi_{\mu\nu}^\ell$ and predictions f_μ at width N exhibit $\mathcal{O}\left(\frac{1}{\sqrt{N}}\right)$ fluctuations around their limiting values (Bordelon & Pehlevan, 2024). Because these fluctuations generically add variance to the predictor (see (Li & Sompolsky, 2021; Pacelli et al., 2023)), they increase the test loss compared to the large N limit as we show in Figure 9. Here, we compare the finite width *light blue* effects in 1HL fully connected architecture (parameterized with NTK parameterization) at finite width and finite $\alpha = P/N = \Theta_N(1)$, with *orange* and *red* curves, which are the finite width effects in our mean-field parameterization. *Light blue* curves in this plot are the predictor performance as obtained in (Pacelli et al., 2023). *Light orange* and *light red* curves are the finite width effects of our aNTK and aNBK predictors respectively.

When N is comparable to P in either parameterization, the kernels Φ^ℓ should actually be thought of as *random matrices* with significant deformations to their spectra compared to the $N \rightarrow \infty$ limit.

D. Deep Bayesian CNNs

In this section, we describe the Bayesian posterior of a Deep convolutional neural network (CNN) for infinitely many channels. Here, we need to add an index a for each weight W_{ija}^ℓ in order to account for the filter value at spacial displacement a from the filter center at each layer, while $\mathcal{S}^\ell$ is the spatial receptive field at layer ℓ . The $L - 1$ hidden layers of the CNN can be expressed as

$$h_{\mu ia}^1 = \frac{1}{\sqrt{D}} \sum_{j=1}^D \sum_{b \in \mathcal{S}^0} W_{ijb}^0 x_{\mu, j, a+b} \quad (67a)$$

$$h_{\mu ia}^{\ell+1} = \frac{1}{\sqrt{N}} \sum_{j=1}^N \sum_{b \in \mathcal{S}^\ell} W_{ijb}^\ell \phi(h_{\mu, j, a+b}^\ell) \quad (67b)$$

$$f_\mu = \frac{1}{\gamma_0 N} \sum_{i=1}^N \sum_a w_{ia}^L \phi(h_{\mu ia}^L) \quad (67c)$$

From these definitions, the partition function turns out to be

$$Z = \int \prod_{\ell=0}^{L-1} \prod_{ijb} dW_{ijb}^\ell \prod_{ia} dw_{ia}^L e^{-\frac{\beta}{2} \gamma_0 N^2 \sum_\mu (y_\mu - \frac{1}{\gamma_0 N} \sum_{i=1}^N \sum_a w_{ia}^L \phi(h_{\mu ia}^L))^2 + \frac{\lambda}{2} \sum_{\ell=0}^{L-1} \sum_{ijb} (W_{ijb}^\ell)^2 + \frac{\lambda}{2} \sum_{ia} (w_{ia}^L)^2} \quad (68)$$

By imposing the pre-activation definitions with the use of the integral representation of some Dirac delta functions as we did in Eq. (20), we can integrate out the weights contribution and just move in the space of representations, and get

$$Z = \int \prod_{\ell=0}^{L-1} \prod_{\mu ia} \frac{dh_{\mu ia}^{\ell+1} d\hat{h}_{\mu ia}^{\ell+1}}{2\pi} \int \prod_{\mu} \frac{ds_{\mu} d\hat{s}_{\mu}}{2\pi} e^{i \sum_{\ell} \sum_{\mu ia} h_{\mu ia}^{\ell+1} \hat{h}_{\mu ia}^{\ell+1} + i \sum_{\mu} s_{\mu} \hat{s}_{\mu}} \int \prod_{\ell} \prod_{ijb} dW_{ijb}^{\ell} \prod_{ia} dw_{ia} e^{-\frac{\beta}{2} \gamma_0 N^2 \sum_{\mu} (y_{\mu} - s_{\mu})^2} \\ \times e^{\frac{\lambda}{2} \sum_{\ell} \sum_{ijb} (W_{ijb}^{\ell})^2 + \frac{\lambda}{2} \sum_{ia} (w_{ia}^L)^2 - i \sum_{\ell} \sum_{\mu ia} \hat{h}_{\mu ia}^{\ell+1} \left(\frac{1}{\sqrt{N}} \sum_{j=1}^N \sum_{b \in S^{\ell}} W_{ijb}^{\ell} \phi(h_{\mu, j, a+b}^{\ell}) \right) - i \sum_{\mu} \hat{s}_{\mu} \left(\frac{1}{\gamma_0 N} \sum_{i=1}^N \sum_a w_{ia}^L \phi(h_{\mu ia}^L) \right)} \quad (69)$$

$$Z = \int \prod_{\ell=1}^{L-1} \prod_{\mu\nu} \prod_{aa'} \frac{d\Phi_{\mu\nu, aa'}^{\ell} d\hat{\Phi}_{\mu\nu, aa'}^{\ell}}{2\pi} \int \prod_{\mu\nu} \frac{d\Phi_{\mu\nu}^L d\hat{\Phi}_{\mu\nu}^L}{2\pi} e^{N \sum_{\ell=1}^{L-1} \sum_{\mu\nu, aa'} \Phi_{\mu\nu, aa'}^{\ell} \hat{\Phi}_{\mu\nu, aa'}^{\ell} + N \sum_{\mu\nu} \Phi_{\mu\nu}^L \hat{\Phi}_{\mu\nu}^L} \\ \times \int \prod_{\mu} \frac{ds_{\mu} d\hat{s}_{\mu}}{2\pi} e^{+i\gamma_0 N \sum_{\mu} s_{\mu} \hat{s}_{\mu} - \frac{\beta}{2} \gamma_0 N^2 \sum_{\mu} (y_{\mu} - s_{\mu})^2 - \frac{1}{2} \sum_{\mu\nu} \hat{s}_{\mu} \hat{s}_{\nu} \frac{1}{\lambda} \Phi_{\mu\nu}^L} \\ \times \left[\int \prod_{\ell=1}^L \prod_{\mu a} \frac{dh_{\mu a}^{\ell} d\hat{h}_{\mu a}^{\ell}}{2\pi} e^{\sum_{\ell=1}^L \sum_{\mu a} h_{\mu a}^{\ell} \hat{h}_{\mu a}^{\ell} - \frac{1}{2} \sum_{\ell=1}^L \sum_{\mu\nu} \sum_{aa'} \hat{h}_{\mu a}^{\ell} \hat{h}_{\nu a'}^{\ell} \frac{\Phi_{\mu\nu, aa'}^{\ell-1}}{\lambda}} e^{-\sum_{\ell=1}^{L-1} \sum_{\mu\nu, aa'} \hat{\Phi}_{\mu\nu, aa'}^{\ell} \left(\sum_b \phi(h_{\mu, a+b}^{\ell}) \phi(h_{\nu, a'+b}^{\ell}) \right)} \right. \\ \left. e^{-\sum_{\mu\nu} \hat{\Phi}_{\mu\nu}^L \left(\sum_a \phi(h_{\mu a}^L) \phi(h_{\nu a}^L) \right)} \right]^N \\ = \int \prod_{\ell=1}^{L-1} \prod_{\mu\nu, aa'} \frac{d\Phi_{\mu\nu, aa'}^{\ell} d\hat{\Phi}_{\mu\nu, aa'}^{\ell}}{2\pi} \int \prod_{\mu\nu} \frac{d\Phi_{\mu\nu}^L d\hat{\Phi}_{\mu\nu}^L}{2\pi} e^{N \sum_{\ell=1}^{L-1} \sum_{\mu\nu, aa'} \Phi_{\mu\nu, aa'}^{\ell} \hat{\Phi}_{\mu\nu, aa'}^{\ell} + N \sum_{\mu\nu} \Phi_{\mu\nu}^L \hat{\Phi}_{\mu\nu}^L} \\ \times e^{-\gamma_0^2 \frac{N}{2} \sum_{\mu\nu} y^{\mu} \left(\frac{1}{\beta} + \frac{\Phi_{\mu\nu}^L}{\lambda} \right)^{-1} y^{\nu} + N \ln \mathcal{Z}} \quad (70)$$

With saddle point equations

$$\Phi_{\mu\nu, aa'}^{\ell} = \left\langle \sum_b \phi(h_{\mu, a+b}^{\ell}) \phi(h_{\nu, a'+b}^{\ell}) \right\rangle \quad \forall \ell = 1, \dots, L-1 \quad (71a)$$

$$\hat{\Phi}_{\mu\nu, aa'}^{\ell} = \frac{1}{2\lambda} \langle \hat{h}_{\mu a}^{\ell} \hat{h}_{\nu a'}^{\ell} \rangle \quad \forall \ell = 1, \dots, L-1 \quad (71b)$$

$$\Phi_{\mu\nu}^L = \left\langle \sum_a \phi(h_{\mu a}^L) \phi(h_{\nu a}^L) \right\rangle \quad (71c)$$

$$\hat{\Phi}_{\mu\nu}^L = -\frac{\gamma_0^2}{2\lambda} \sum_{\alpha\beta} \left[\frac{\mathbb{I}}{\beta} + \frac{\Phi}{\lambda} \right]_{\mu\alpha}^{-1} y_{\alpha} y_{\beta} \left[\frac{\mathbb{I}}{\beta} + \frac{\Phi}{\lambda} \right]_{\beta\nu}^{-1} \quad (71d)$$

E. DMFT review

In this section, we briefly recall the dynamical mean field theory derivation of a gradient flow dynamics for a multi-layer fully connected neural network. As clarified in the main text, we are interested in a fully connected feedforward network with L layers, each of width N , defined as

$$f_{\mu} = \frac{1}{\gamma} h_{\mu}^{L+1} = \frac{1}{\gamma\sqrt{N}} \mathbf{w}^{(L)} \cdot \phi(\mathbf{h}_{\mu}^L), \quad \mathbf{h}_{\mu}^{\ell+1} = \frac{1}{\sqrt{N}} \mathbf{W}^{\ell} \phi(\mathbf{h}_{\mu}^{\ell}), \quad \mathbf{h}_{\mu}^1 = \frac{1}{\sqrt{D}} \mathbf{W}^{(0)} \mathbf{x}_{\mu} \quad (72)$$

where each trainable parameter $\mathbf{W}^{\ell}$ is initialized as a Gaussian random variable $W_{ij}^{\ell} \sim \mathcal{N}(0, 1)$ with unit variance. Here, the gradient updates for the weights $\mathbf{W}^{\ell}(t)$ and the output function f_{μ} are given by

$$\frac{d\mathbf{W}^{\ell}(t)}{dt} = \frac{\gamma}{N} \sum_{\mu} \Delta_{\mu}(t) \mathbf{g}_{\mu}^{\ell+1}(t) \phi(\mathbf{h}_{\mu}^{\ell}(t))^{\top} - \lambda \mathbf{W}^{\ell}(t), \quad \frac{df_{\mu}}{dt} = \sum_{\alpha=1}^P K_{\mu\alpha}^{\text{NTK}}(t, t) \Delta_{\alpha}(t) - \lambda \kappa f_{\mu} \quad (73)$$

where $\Delta_\mu(t) = -\frac{\partial \mathcal{L}}{\partial f_\mu(t)}$ represents the pattern error signal for a pattern μ , and $\mathbf{g}_\mu^\ell(t) = \sqrt{N} \frac{\partial h_\mu^{\ell+1}(t)}{\partial \mathbf{h}_\mu^\ell(t)}$ captures the backpropagated gradients flowing from the downstream layers. Instead, the term $\phi(\mathbf{h}_\mu^\ell(t))$ involves the activations of the ℓ -th layer and reflects the forward pass contribution to the weight update.

As specified in the main text, for the representer theorem to be valid in this case, we restrict to κ degree homogeneous network, whose output scales as $f(a\theta) = a^\kappa f(\theta)$. Here, the predictor dynamics is governed by the driving force of the error signal propagated through the *adaptive Neural Tangent Kernel* $K_{\mu\nu}^{\text{aNTK}}(t, t') = \frac{\partial h_\mu^{\ell+1}(t)}{\partial \theta} \cdot \frac{\partial h_\nu^{\ell+1}(t')}{\partial \theta}$. This quantifies the interaction between parameter gradients for outputs of pattern pairs (μ, ν) at times (t, t') . The homogeneity factor κ appears in the second term from the weight decay contribution. Some quantities of interest to define are the forward and gradient kernels at each layer

$$\Phi_{\mu\nu}^\ell(t, t') = \frac{1}{N} \phi(\mathbf{h}_\mu^\ell(t)) \cdot \phi(\mathbf{h}_\nu^\ell(t')), \quad G_{\mu\nu}^\ell(t, t') = \frac{1}{N} \mathbf{g}_\mu^\ell(t) \cdot \mathbf{g}_\nu^\ell(t') \quad (74)$$

which allows to rewrite $K_{\mu\nu}^{\text{aNTK}}(t, t') = \sum_{\ell=0}^L G_{\mu\nu}^{\ell+1}(t, t') \Phi_{\mu\nu}^\ell(t, t')$. If we take care of the initial conditions over the weights, in the DMFT limit where $N, \gamma \rightarrow \infty$ with $\gamma = \gamma_0 \sqrt{N}$, we can determine the final kernels by solving the field dynamics

$$\begin{aligned} h_\mu^\ell(t) &= e^{-\lambda t} \xi_\mu^\ell(t) + \gamma_0 \int_0^t dt' e^{-\lambda(t-t')} \sum_\nu \Delta_\nu(t') g_\nu^\ell(t') \Phi_{\mu\nu}^{\ell-1}(t, t') \\ z_\mu^\ell(t) &= e^{-\lambda t} \psi_\mu^\ell(t) + \gamma_0 \int_0^t dt' e^{-\lambda(t-t')} \sum_\nu \Delta_\nu(t') \phi(\mathbf{h}_\nu^\ell(t')) G_{\mu\nu}^{\ell+1}(t, t'). \end{aligned} \quad (75)$$

In Eq. (75), both pre-activations $h_\mu^\ell(t)$ and pre-gradient signals $z_\mu^\ell = \frac{1}{\sqrt{N}} W^\ell g_\mu^{\ell+1}$ decouple over the neuron index and factorize over the layer index, and the contribution from initial conditions $\xi_\mu^\ell(t) = \frac{1}{\sqrt{N}} W^\ell(0) \phi(h_\mu^{\ell-1})(t)$ and $\psi_\mu^\ell(t) = \frac{1}{\sqrt{N}} W^\ell(0) g_\mu^{\ell+1}(t)$ is exponentially suppressed at large time t . Simulating a stochastic process like Eq. (75) requires keeping track of the entire history trajectory, and computing at each step produces of kernel matrices that have dimension $PT \times PT$. This scales cubically in both sample and time dimensions $\mathcal{O}_N(P^3 T^3)$, allowing in principle to solve for the field dynamics when $P, T = \mathcal{O}_N(1)$. A sketch of an iterative algorithm procedure can be found in Algorithm 1, where given an initial guess on $\{\Phi^\ell, G^\ell\}_{\ell=1}^L$ one can compute K^{aNTK} and solve for the predictor dynamics of Eq. (72) once drawn a number $\mathcal{S}$ of samples $\{\xi_{\mu,n}^\ell(t)\}_{n=1}^{\mathcal{S}} \sim \mathcal{N}(0, \Phi^{\ell-1})$, $\{\psi_{\mu,n}^\ell(t)\}_{n=1}^{\mathcal{S}} \sim \mathcal{N}(0, G^{\ell+1})$ and solved Eq. (75) for each $\{h_{\mu,n}^\ell(t), z_{\mu,n}^\ell(t)\}_{n=1}^{\mathcal{S}}$. A sketch of the solver can also be found in the main text Algorithm 2. As can be noticed in Figures 10, solving for the field dynamics gives a good agreement with simulations. Precisely, once knowing the predictor, one can study how both train and test losses updates during the training dynamics and for different values of γ . Fig. 10(b) (top left panel) shows that the *lazy* learning regime when $\gamma = 0$ does not allow the network to interpolate on the training data and leads to a higher test loss compared to the rich cases with $\gamma > 0$. Increasing γ has also the effect of speed up learning, while for that given sample size value ($P = 200$ here) there exists an optimal degree of feature learning (i.e. γ) concerning the test loss. The specifics of the learning task can be found in the figure caption. Fig. 10 also shows the non-Gaussianity of the pre-activation and pre-gradient distributions once the system has thermalized, at the end of training. As mentioned above, because of the weight initialization as $W_{ij}^\ell \sim \mathcal{N}(0, 1)$, both $\{h, z\}$ are Gaussian distributed when the training starts. Then feature learning has the effects of accumulating non-Gaussian contributions as the training proceeds.

E.1. DMFT for Convolutional Networks

In Fig. 11 we show how our DMFT theory can be extended to a two-layer CNN which is trained on CIFAR10 images on a subset of $P = 100$ (*left*) or $P = 1000$ (*right*) points. When data sample is small, for any value of γ_0 there is an optimal early stopping time for the best Test Loss performance.

The theory to get the predictor is an easy extension to the multi-layer fully-connected setting and can be found in (Bordelon & Pehlevan, 2022). However, we recall here for the sake of clarity how the field dynamic equations get modified in this setting. Again, in our notation a is the spatial displacement from the center of the filter at each layer where $b \in \mathcal{S}^\ell$ is the spatial relative field at layer ℓ as in Eq. (D). The pre-activation definitions still remain the same as it is in Appendix D. In the same way, the gradient signal are now defined as

$$\mathbf{g}_{\mu,a}^\ell = \gamma_0 N \sum_b \frac{\partial f}{\partial \mathbf{h}_{\mu,b}^{\ell+1}} \cdot \frac{\partial \mathbf{h}_{\mu,b}^{\ell+1}}{\partial \mathbf{h}_{\mu,a}^\ell}. \quad (76)$$

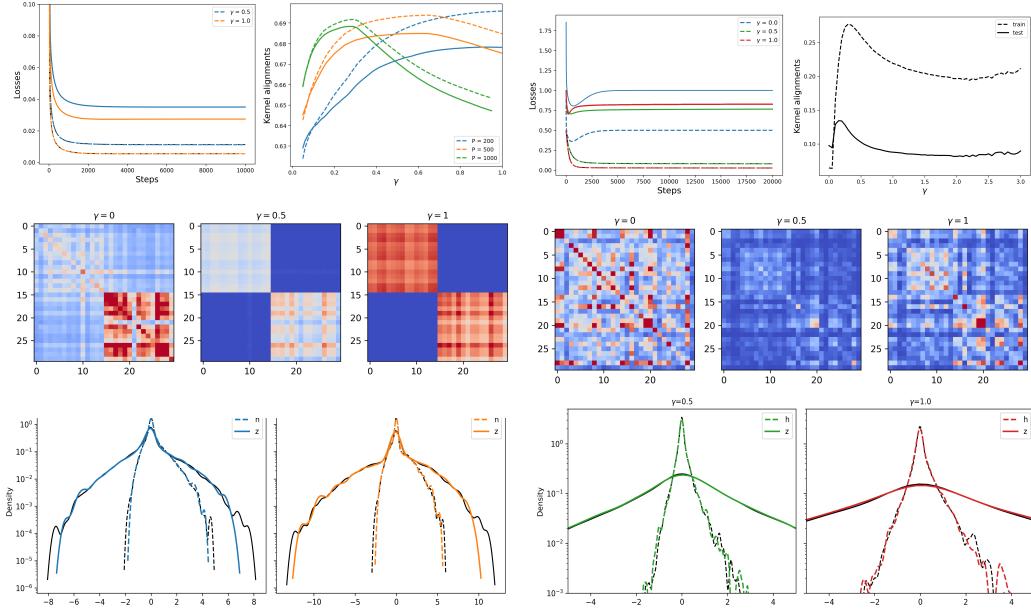


Figure 10. The training dynamics of two layer ReLU MLPs trained with weight decay. The richly trained networks achieve lower training and test errors at equal levels of regularization. Our theory can reproduce the final preactivation and pregradient densities in each setting.

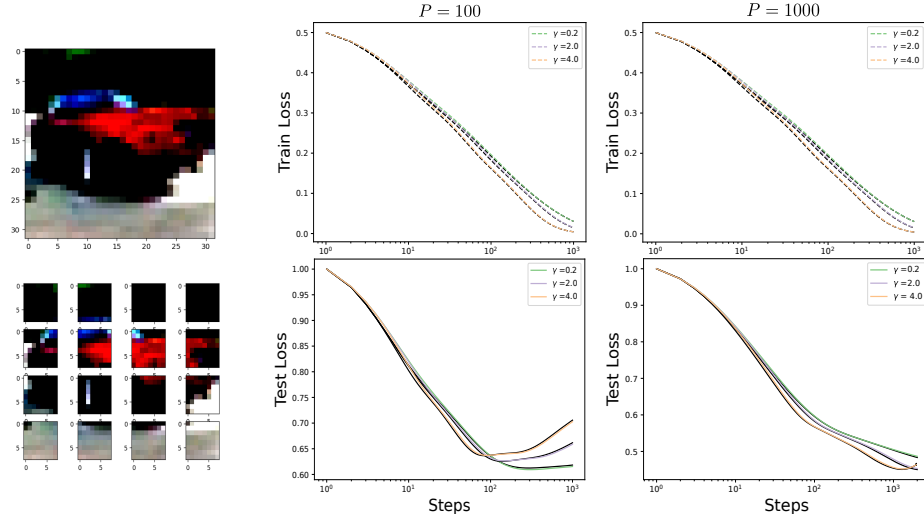


Figure 11. Training dynamics for a two layer CNN for varying richness γ_0 on CIFAR-10 images. The images are turned into patches before computing cross-spatial correlations in the data. The training dynamics for infinite width networks (black) is compared to training finite width networks. For small training set sizes, richer training can result in faster overfitting, while this effect is less severe when there is more data.

while the weight dynamics per filter is

$$\frac{d}{dt} \mathbf{W}_b^\ell(t) = \frac{\gamma_0}{\sqrt{N}} \sum_{\mu,a} \Delta_\mu \mathbf{g}_{\mu,a}^{\ell+1} \phi(\mathbf{h}_{\mu,a+b})^\top - \lambda \mathbf{W}_b^\ell(t). \quad (77)$$

Given that, the stochastic dynamics for the pre-activation and pre-gradient signals (similar to Eq. (75)) becomes

$$\begin{aligned} \mathbf{h}_{\mu,a}^{\ell+1}(t) &= e^{-\lambda t} \boldsymbol{\xi}_{\mu,a}^{\ell+1}(t) + \gamma_0 \int_0^t dt' e^{-\lambda(t-t')} \sum_{\nu,b,c} \Delta_\nu(t') \Phi_{\mu\nu,a+b,a+c}^\ell(t, t') \mathbf{g}_{\nu,c}^{\ell+1}(t') \\ \mathbf{z}_{\mu,a}^\ell(t) &= e^{-\lambda t} \boldsymbol{\psi}_{\mu,a}^\ell(t) + \gamma_0 \int_0^t dt' e^{-\lambda(t-t')} \sum_{\nu,b,c} \Delta_\nu(t') G_{\mu\nu,a-b,c-b}^{\ell+1}(t, t') \phi(\mathbf{h}_{\nu,c}^\ell) \end{aligned} \quad (78)$$

where again $\boldsymbol{\xi}_{\mu,a}^{\ell+1}(t) = \frac{1}{\sqrt{N}} \mathbf{W}^\ell(0) \phi(\mathbf{h}_{\mu,a}^\ell(t))$ and $\boldsymbol{\psi}_{\mu,a}^\ell(t) = \frac{1}{\sqrt{N}} \mathbf{W}^\ell(0) g_{\mu,a}^{\ell+1}(t)$ from the initial conditions. At large time t , as it is for the fully connected dynamics, the contribution from initial condition get suppressed and the fixed point predictor is a kernel predictor, being the feature and gradient kernels now

$$\Phi_{\mu a, \nu b}^\ell(t, t') = \frac{1}{N} \phi(\mathbf{h}_{\mu a}^\ell(t)) \cdot \phi(\mathbf{h}_{\nu b}^\ell(t')), \quad G_{\mu a, \nu b}^\ell(t, t') = \frac{1}{N} \mathbf{g}_{\mu a}^\ell(t) \cdot \mathbf{g}_{\nu b}^\ell(t'). \quad (79)$$

The Neural Tangent Kernel is instead $K_{\mu\nu}^{\text{aNTK}}(t, t) = \sum_\ell \sum_{ab} \Phi_{\mu a, \nu b}^\ell(t, t) G_{\mu a, \nu b}^{\ell+1}(t, t)$ and the predictor again at convergence $f(\mathbf{x}) = \frac{1}{\kappa \lambda_L} \sum_\nu \Delta_\nu K^{\text{aNTK}}(\mathbf{x}, \mathbf{x}_\nu)$.

F. Fixed point structure of GD with Weight Decay

In what follows, we will be interested in the infinite time limit of a dynamics such that in Eq. (75). Prior work on the dynamics of L2 regularization in the kernel regime revealed that training a wide network for infinite time leads to collapse of the features and network predictor to zero (Lewkowycz & Gur-Ari, 2020). However, if one instead adopts a μ P scaling, then it is *possible* to have a non-trivial fixed point at infinite width as the feature learning updates and regularization updates are of the same order (Bordelon & Pehlevan, 2022). This is because from Eq. , we realize that in the setting where $\lambda > 0$, not only the initial contribution of the fields dynamics are suppressed at large time t , but also the second terms contribute the most when the system has equilibrated, leading to a predictor which is a kernel predictor

$$f(\mathbf{x}_*) = \mathbf{k}(\mathbf{x}_*)^\top [\mathbf{K} + \lambda \kappa \mathbf{I}]^{-1} \mathbf{y} \quad (80)$$

Because of the simple interpretation of DNNs in this regime, we wish to say something about the fixed point structure of the field dynamics, which are

$$h_\mu^\ell = \frac{\gamma_0}{\lambda} \sum_\nu \Delta_\nu \Phi_{\mu\nu}^{\ell-1} \dot{\phi}(h_\nu) z_\nu, \quad z_\mu^\ell = \frac{\gamma_0}{\lambda} \sum_\nu \Delta_\nu \phi(h_\nu) G_{\mu\nu}^{\ell+1}. \quad (81)$$

In what follows, we specialize to simple solvable cases to gain intuition for these fixed point constraints. In general, these constraints imply a set of possible joint densities over h, z as well as determine the final feature and gradient kernels and the predictor.

F.1. Two Layer MLP with whitened data

Let's consider a single data point with a white covariance matrix $K^x = 1$, label $y = 1$ and a transfer function $\phi(x) = \text{ReLU}(x)$. In this setting, the dynamics of training for the pre-activation and pre-gradient signals are

$$\frac{d}{dt} h(t) = \gamma_0 \Delta(t) g(t) - \lambda h(t), \quad \frac{d}{dt} z(t) = \gamma_0 \Delta(t) \phi(h(t)) - \lambda z(t). \quad (82)$$

At the fixed point, the following conditions are satisfied

$$h = \frac{\gamma_0}{\lambda} \Delta \dot{\phi}(h) z, \quad z = \frac{\gamma_0}{\lambda} \Delta \phi(h). \quad (83)$$

In principle, there are infinitely many solutions to these equations, and combining them gives the following constraint on h

$$h = \frac{\gamma_0^2}{\lambda^2} \Delta^2 \dot{\phi}(h) \phi(h) = \frac{\gamma_0^2}{\lambda^2} \Delta^2 \phi(h). \quad (84)$$

Since we know that here $\phi(h) = \max(0, h)$, this means that the following two constraints on the pre-activation density must be satisfied

$$\forall h < 0 \quad p(h) = 0, \quad \Delta = \frac{\lambda}{\gamma}. \quad (85)$$

Lastly, we have the equation that fixes the value of the pattern error signal Δ through the predictor definition, which is

$$\Delta = 1 - \frac{1}{\gamma_0} \langle z \phi(h) \rangle = 1 - \gamma_0^{-1} \langle h^2 \rangle \quad (86)$$

implying that the second moment of h must give

$$\langle h^2 \rangle = \gamma_0 - \lambda. \quad (87)$$

The solution here is correct if $\gamma_0 > \lambda$. Otherwise $\langle h^2 \rangle = 0$ and $p(h) = \delta(h)$. We can verify that this is true by comparing the pre-activation density of a two-layer MLP trained until interpolation with the theoretical predictions of the fixed points. As shown in Fig. 12, there is a sharp phase transition in the limiting density right when $\gamma_0 = \lambda$. Precisely, when $\gamma_0 < \lambda$, the effect of feature learning here is to kill the left-side of the distribution and adjusting the variance of $h > 0$ such that the constraint of Eq. (87) is satisfied. Another way of saying this is that in the infinite time limit $t \rightarrow \infty$, the $\{\gamma_0, \lambda_0\} \rightarrow 0$ limits do not commute. In the first case of Fig. 12, when $\lim_{\gamma_0 \rightarrow 0, \lambda \rightarrow 0}$ we get a stable non-Gaussian behavior for the $p(h)$. In the second case of Fig. 12 we see a collapse when $\lim_{\lambda, \gamma \rightarrow 0}$ and nothing is learned by the network. In the same way, in the limit where we fix $t = \mathcal{O}_N(1)$ and study the ridge-less limit of a lazy network ($\gamma_0 = 0$), we recover the Neural Tangent Kernel predictor, and consequently the Gaussian pre-activation density at initialization.

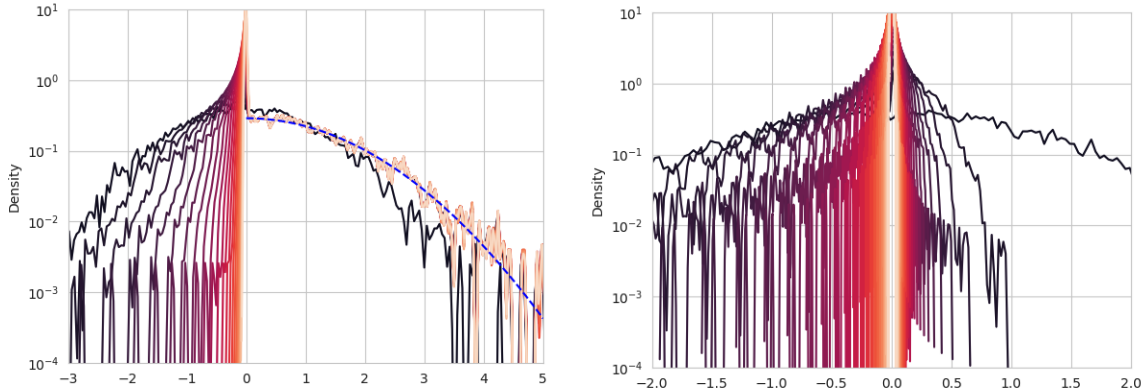


Figure 12. Pre-activation densities of a two-layer MLP trained with GD and weight decay at different times. Ligther colors represent the end of training. Dashed blue line is the theoretical prediction from the fixed point in the infinite time limit.

F.2. Linear case

In principle, the constraints one gets by looking at fixed points of Eq. (81) fix the first two moments of the pre-activation density distribution, but are not enough to determine the full marginal $p(h)$. Indeed, this remains history dependent and one in principle has to track the entire update dynamics in order to get the full description. One possible way of understanding this is by looking at the simplest, linear case. Here, we initialize the weights of a two-layer MLP to be Laplace distributed. Here, we expect the distribution $p(h)$ to be Gaussian distributed if we train with weight decay until interpolation. Fig. 13 shows that training with weight decay to the fixed point does not recover a Gaussian single site density. But it does have the properties demanded by the saddle point equations, which are again $\Delta = \lambda/\gamma_0$; $\langle h^2 \rangle = \gamma_0 - \lambda$.

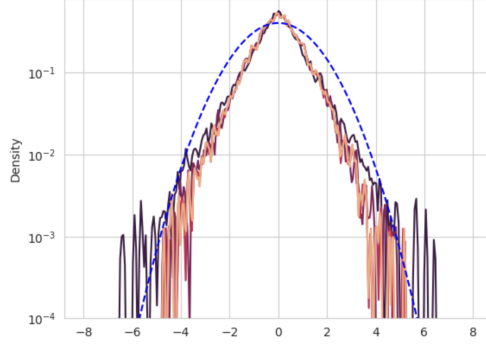


Figure 13. Pre-activation density of a two-layer MLP trained with $P = 1$ with a white covariance matrix. Dark colors correspond to early time training, being the weights initialized as Laplace distributed at $t = 0$. Light colors coincide with the end of training, when the system has thermalized. Blue dashed line is the theory prediction from the fixed point equations.

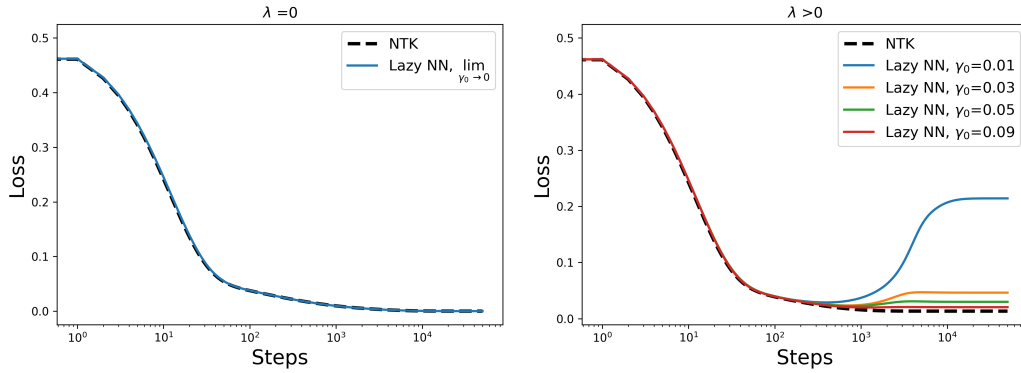


Figure 14. Weight decay in the lazy training regime can cause a model to “unlearn” and reduce its output after the features start to decay. Provided γ_0 is sufficiently large compared to λ , however, the final predictor will still be nontrivial, unlike the zero predictor obtained in NTK parameterization (Lewkowycz & Gur-Ari, 2020).

G. Algorithmic implementation and numerical details

In this section, we provide more details regarding the numerical methods used for solving both the adaptive kernel theories and for training the finite width N neural networks.

G.1. Data pre-processing

As clarified in the main text, we focus on regression problems on two classes (respectively labeled as $-1/1$) for both MNIST and CIFAR10 datasets. For MLPs architectures, the input patterns have 28×28 pixels except when comparing the MLP test prediction with CNN as we did in Figure 5. Indeed, for all the experiments except the one in Figure 5, we resize CIFAR10 images to 28×28 pixels by reducing their spatial resolution and then converting them to grayscale. In each case, we normalize the pixel values to have a consistent scale, and then we flatten the images as a $D = 784$ input vectors. For CNNs on CIFAR10, we preserve the spatial structure of the 32×32 pixel images rather than flattening them. Specifically, after normalization, each image is partitioned into non-overlapping patches of size 8×8 , resulting in a multi-dimensional array that maintains the local neighborhood relationships. This representation enables the convolutional layer to apply a $K = 8 \times 8$ spatial filter directly, thereby effectively capturing localized features relevant to the regression tasks.

G.2. Theory solver: Min-Max optimization for Bayesian DNNs

In this section, we provide more detail on the solver for the aNBK algorithm. We use automatic differentiation (via JAX) to compute gradients of the action $S(\{\Phi^\ell, \hat{\Phi}^\ell\})$ (Equation 6) with respect to the order parameters $\{\Phi^\ell, \hat{\Phi}^\ell\}$ (Bradbury et al.,

2018). A key challenge for this is to estimate the single site moment generating functions $\mathcal{Z}_\ell$ as a differentiable function of both variables Φ^ℓ and $\hat{\Phi}^\ell$. To do so, we use importance sampling, by recognizing that, conditional on $\Phi^{\ell-1}$, the $\mathcal{Z}_\ell$ can be expressed as an average of a nonlinear function with respect to a Gaussian with covariance $\Phi^{\ell-1}$

$$\begin{aligned}\mathcal{Z}_\ell &= \left\langle \exp \left(-\frac{1}{2} \phi(\mathbf{h})^\top \hat{\Phi}^\ell \phi(\mathbf{h}) \right) \right\rangle_{\mathbf{h} \sim \mathcal{N}(0, \Phi^{\ell-1})} \\ &\approx \frac{1}{B} \sum_{k=1}^B \exp \left(-\frac{1}{2} \phi(\mathbf{h}_k)^\top \hat{\Phi}^\ell \phi(\mathbf{h}_k) \right).\end{aligned}\quad (88)$$

In our code, in practice, each of the vectors $\mathbf{h}_k$ are i.i.d. draws from $\mathcal{N}(0, \Phi^\ell)$, by setting $\lambda_\ell = 1 \forall \ell \in \{L\}$.

At each step of the iteration scheme for the min-max solver, we resample a new batch of B vectors $\mathbf{h}_k$ and use these to estimate $\mathcal{Z}_\ell$, which provides fresh samples at each iteration of the algorithm.

Our solver proceeds by alternately optimizing over the conjugate kernels $\{\hat{\Phi}^\ell\}_{\ell=1}^L$ (via gradient ascent in the inner loop) and the kernels $\{\Phi^\ell\}_{\ell=1}^L$ (via gradient descent in the outer loop). In practice, the inner loop runs for $t_{\text{inner}} \sim 200$ steps gradient steps, after which the outer loop updates $\{\Phi^\ell\}_{\ell=1}^L$ given $\{\hat{\Phi}^\ell\}_{\ell=1}^L$ for a maximum number $t_{\text{outer}} \sim 20000$ of iterations. The code implement this scheme as follows:

- **Kernel Initialization:** A function `init_kernels` computes the initial *lazy* guesses for kernels $\{\Phi^\ell, \hat{\Phi}^\ell\}_{\ell=1}^L$, given a data covariance matrix $\Phi^0 = \frac{1}{D} \mathbf{x} \mathbf{x}^\top \in \mathbb{R}^{P \times P}$. At layer $\ell = 1$, this involves generating a `samples = 20000` number of Gaussian vectors $\mathbf{h}^1 \in \mathbb{R}^{P \times \text{samples}}$ (by drawing from $\mathcal{N}(0, \Phi^0)$ after computing the square-root of Φ^0) and then computing an empirical kernel via $\Phi^1 = \frac{1}{\text{samples}} \phi(\mathbf{h}^1) \phi(\mathbf{h}^1)^\top \in \mathbb{R}^{P \times P}$. The same iterative procedure is applied at each layer, given $\Phi^\ell = \frac{1}{\text{samples}} \phi(\mathbf{h}^{\ell-1}) \phi(\mathbf{h}^{\ell-1})^\top$. For the conjugated kernels, $\hat{\Phi}^\ell = \mathbf{0}^{P \times P} \forall \ell \in \{L\}$.
- **Single-Site Estimation:** A function `single_site` computes the log-normalized moment generating function by taking an average over the batch of B samples as described above: $\log \mathcal{Z}_\ell = \log \left(\frac{1}{B} \sum_{k=1}^B \exp \left(-\frac{1}{2} \phi(\mathbf{h}_k)^\top \hat{\Phi}^\ell \phi(\mathbf{h}_k) \right) \right)$.
- **Action Function:** The overall action $S(\{\Phi^\ell, \hat{\Phi}^\ell\}_{\ell=1}^L)$ is computed through a function `action` as in Equation (6).
- **Gradient Updates:** The gradients of $S(\{\Phi^\ell, \hat{\Phi}^\ell\}_{\ell=1}^L)$ with respect to Φ^ℓ and $\hat{\Phi}^\ell$ are computed using JAX's automatic differentiation and are used to update the corresponding variables via gradient ascent/descent (GD) as detailed above. Two separate learning rates (or update steps), `up_step_Phi` $\sim 1e-5$ and `up_step_hatPhi` $\sim 1e-4$, are used to control the outer and inner updates, respectively.

G.3. Theory solver: DMFT dynamics

In this section, we detail the implementation of our DMFT dynamics solver, which is used to simulate the gradient flow dynamics with weight decay (given a regularization parameter λ at each layer $\ell \in \{L\}$). Minor variations to take into account spatial positions are implemented for the CNN case. This solver is based on the DMFT derivation described in Section E and captures the evolution of the pre-activation fields h_μ and gradient signals z_μ via Monte Carlo sampling. For simplicity, we restrict to the case $L = 1$ (i.e., two layers architecture).

The implementation proceeds as follows:

- **Fields initialization:** an initial set of `samples = 20000` Gaussian pre-activation fields $\mathbf{h}(0) \in \mathbb{R}^{P \times \text{samples}}$ is generated by drawing Monte Carlo samples from $\mathcal{N}(0, \mathbf{K}^x)$, where $\mathbf{K}^x = \Phi^{\ell=0} = \frac{1}{D} \mathbf{x} \mathbf{x}^\top \in \mathbb{R}^{P \times P}$ is the Gram matrix of the P data. In the same way, an auxiliary array of length `samples = 20000`, $\mathbf{z}(0) \in \mathbb{R}^{\text{samples}}$ is generated by sampling Gaussian random variables $z_s(0) \sim \mathcal{N}(0, 1)$. From that, the gradient field $\mathbf{g}(0) \in \mathbb{R}^{P \times \text{samples}}$ is computed as $\mathbf{g}(0) = \dot{\phi}(\mathbf{h}(0)) \odot \mathbf{z}(0)$ (Equation (13)).
- **Initial kernels and error signal:** Having $\mathbf{h}(0) \in \mathbb{R}^{P \times \text{samples}}$, the feature kernel is evaluated as $\Phi^{\ell=1}(0) = \frac{1}{\text{samples}} \phi(\mathbf{h}(0)) \phi(\mathbf{h}(0))^\top \in \mathbb{R}^{P \times P}$, being $\phi(\cdot) = \max(0, \cdot)$ the ReLU activation function. The initial gradient kernels are instead: $\mathbf{G}^{\ell=1}(0) = \frac{1}{\text{samples}} \mathbf{g}(0) \mathbf{g}(0)^\top \in \mathbb{R}^{P \times P}$, $\mathbf{G}^{\ell=2} = \mathbf{I} \in \mathbb{R}^{P \times P} \forall t \in \{T\}$.

According to Section E, the initial error signal for P patterns $\Delta(0) \in \mathbb{R}^P$ is computed as $\Delta(0) = \mathbf{y} - \frac{1}{\gamma_0 \text{samples}} \phi(\mathbf{h}(0)) \mathbf{z}(0)$ because we focus on MSE loss $\mathcal{L} = \frac{1}{2} \sum_{\mu=1}^P (\Delta_{\mu})^2$.

- **Field dynamics:** at each step for $T = 20000$ steps, the fields and the error signals are updated according to the dynamics

$$\mathbf{h}(t+1) = \mathbf{h}(t) + \eta \gamma_0 (\mathbf{g}(t) \odot \Delta(t)) \Phi^0(t) - \lambda \mathbf{h}(t) \quad (89a)$$

$$\mathbf{z}(t+1) = \mathbf{z}(t) + \eta \gamma_0 (\phi(\mathbf{h}(t)) \odot \Delta(t)) - \lambda \mathbf{z}(t) \quad (89b)$$

$$\Delta(t) = \mathbf{y} - \frac{1}{\gamma_0 \text{samples}} \phi(\mathbf{h}(t)) \mathbf{z}(t). \quad (89c)$$

- **Loss:** Training loss $\mathcal{L}(t)$ are computed as the mean squared error over the corresponding $\Delta(t)$, since $\mathcal{L}(t) = \frac{1}{2} \sum_{\mu=1}^P \Delta_{\mu}(t)^2$.
- **Return:** The function `solve_dynamics_reg` returns: list of losses over iterations $\mathcal{L}(t)$, the target vector $\mathbf{y}$, final states $\{\mathbf{h}(T), \mathbf{z}(T)\}$ for the fields and the final kernels $\{\Phi^{\ell=1}(T), \mathbf{G}^{\ell=1}(T)\}$. From that, the aNTK kernel is computed as $\mathbf{K}^{\text{aNTK}}(T) = \text{Tr}(\Phi^{\ell=0}(T) \mathbf{G}^{\ell=1}(T)) + \Phi^{\ell=1}(T)$.

In our theory solver, learning rate $\eta = 1 \times 10^{-3}$, samples = 20000, and we span over P, γ_0 values.

G.4. MLP Experiments

For designing the model, we construct a deep multilayer perceptron (MLP) in JAX. The architecture is built as a sequence of weight matrices with input dimension $D = 784$ (except for Figure 5 where the input dimension is $D = 1024$) and hidden layer width $N = 1024$ for a number $L + 1$ of hidden layers, each randomly initialized from a normal distribution such that $W_{ij}^{\ell} \sim \mathcal{N}(0, 1)$. Each hidden layer is followed by a ReLU non-linearity after being normalized according to the mean-field scaling as in Equation (2). The last layer returns a scalar output that serves as the regression prediction.

For training, we either employ a variant of deep Langevin dynamics, or a gradient descent with weight decay dynamics according to Equation (3). The MLP is trained to minimize a regularized mean squared error (MSE) loss (rescaled by the factor $\frac{1}{2} \gamma_0^2 N$) in both cases. For Langevin training, the optimizer we use - `optax.noisy_sgd` - is designed to inject Gaussian white noise into the gradient updates at each iteration. This noise is drawn from a zero-mean Gaussian distribution whose variance is controlled by both the learning rate η and the inverse temperature parameter β^{-1} as in Equation (3). This white noise plays a critical role in approximating the posterior distribution (Equation (9)) over the network parameters.

For both Langevin and gradient descent dynamics, we use a weight decay contribution proportional to λ as it is for (3) (in all our experiments we use $\lambda = 1$ for each layer, except when we compare with CNN test loss, where $\lambda = 1 \times 10^{-2}$). For Langevin, we average the $T = 20000$ steps fluctuations every 1000 steps after $t > 5000$ epochs. We use a learning rate $\eta = 5 \times 10^{-4}$ and an inverse temperature $\beta = 50$. For gradient descent, we train until convergence for $T = 20000$ epochs and we use a learning rate $\eta = 1 \times 10^{-3}$. Both the experiments are performed by varying the sample size P and the feature learning strength γ_0 .

G.5. CNN Experiments

We implement our CNN using Flax's Linen module in JAX. The model architecture and training strategy are designed to capture localized features from the animate/inanimate regression tasks on CIFAR10 images. The CNN consists of a single convolutional layer with a kernel size of 8×8 and stride equal to the kernel size. This choice effectively splits each 32×32 input image (with 3 color channels) into non-overlapping patches. We set the number of filters to $N = 1024$, and the convolution weights are initialized using a normal distribution with unit variance. To ensure that the variance is appropriately scaled, the convolutional output is divided by $\sqrt{3 \times 8^2}$. A ReLU activation is applied afterwards. This activated output is then flattened and fed to a dense layer that outputs a single scalar prediction; the dense layer's weights are further scaled by a factor of $1/(\gamma_0 N)$ to maintain the mean field scaling as in Equation (2).

The CNN is trained using a gradient-based optimization strategy over $T = 20000$ epochs, over a MSE loss. Before each parameter update, the gradients are scaled by a factor of $\frac{1}{2}\gamma_0^2 N$ and regularized with a weight decay term proportional to λ as it is in the dynamics of Equation (3). The key hyperparameters for training the CNN are chosen as follows: learning rate $\eta = 1 \times 10^{-3}$, regularization $\lambda = 1 \times 10^{-2}$, and the experiments are performed by varying the number of training sample P and the feature learning strength γ_0 .

G.6. Computational overhead

Time for Kernels Algorithm (1): the most computationally expensive part is the Monte Carlo estimation of the non-Gaussian single-site density. At each inner step, we sample a batch of B Gaussian vectors (as clarified in G.2) and estimate the single site density function, an operation costing roughly $O(BP^2)$ per inner update (since it involves $P \times P$ matrix operations). Repeating this for t_{inner} inner steps gives an overhead of approximately $O(t_{\text{inner}}BP^2)$ per outer update.

In contrast, when training a finite-width network via full Langevin dynamics, the end-to-end cost scales as $O(TN^2P)$, where $N = 1024$ is the network width. For comparable T (we use $T = t_{\text{outer}} = 20000$ for both Langevin and theory solver), this is typically much lower than the overhead of the full theory solver, $O(t_{\text{outer}}t_{\text{inner}}BP^2)$, especially when the sample size P is large.

For deep linear networks—where the single-site density is Gaussian—the overhead of Algorithm (1) is significantly reduced because the conjugate kernels can be solved explicitly (see Eq. (15)), leading to a cost of $O(P^3)$ per step.

When comparing to DMFT-based analysis (Algorithm (2)), the full dynamics require handling $PT \times PT$ kernels, leading to a time complexity of $O(T^3P^3)$ for the complete dynamics. In our work, Algorithm (1) already represents an improvement over Algorithm (2). In Appendix E.1, we propose an $O(TP^3)$ version of Algorithm (2) by leveraging weight decay to directly solve the fixed-point equations.

To get an order of magnitude, in most of our simulations $t_{\text{outer}} = 10^4$, $t_{\text{inner}} = 2 \times 10^2$, and $T = 2 \times 10^4$, $B = 2 \times 10^4$ and sample size at most $P = 10^3$.

Time for output predictions Generating the final predictions using Algorithm (1) requires approximately $O(t_{\text{outer}}t_{\text{inner}}BP^2)$ operations, as the procedure involves repeated sampling and kernel evaluations over the training dynamics. By comparison, Algorithm (2) incurs a heavier cost of $O(T^3P^3)$, while a simple kernel regression with a known kernel only requires $O(P^3)$ time. End-to-end training of a finite-width network scales as $O(TN^2P)$ (for both Langevin and gradient descent with weight decay dynamics), which is significantly more efficient in practice when P is large.

Memory for kernels For Algorithm (1), the memory footprint is primarily dictated by the storage of the kernels $\{\Phi^\ell, \hat{\Phi}^\ell\}_{\ell=1}^L$, each requiring roughly $O(P^2)$ space. In contrast, Algorithm (2) have a memory cost $O(P^2T^2)$. Meanwhile, finite-width network training typically demands a memory proportional to $O(N^2)$.

H. Glossary

Here we provide more explanation of our choice of terminology for the various kernel predictors. We use the letter K at the end of a name to indicate that this predictor is a kernel method with *no variance from random weight prior or initialization*. We use the prefix letter “a” to indicate an *adaptive kernel method* where the feature kernels adapt to the structure of the learning task.

- NNGP: the Gaussian process with $\Theta_N(1)$ variance for the network outputs under both the prior and the posterior. The mean of this process is a kernel method with the matrix Φ^ℓ in the lazy limit.
- NNGPK: the mean kernel predictor for lazy training with the initial final kernel neglecting. This corresponds to $N \rightarrow \infty$ first followed by $\gamma_0 \rightarrow 0$ in our parameterization. In this limit, there is no variance of the predictor under either prior or posterior.
- NTK: a kernel method for the initial neural tangent kernel at infinite width without any randomness or variability in the predictor from initialization. This is the predictor obtained in the NTK parameterization if the initial output of the model is subtracted off (ie if a centering operation is performed where $f(\theta, x) \rightarrow f(\theta, x) - f(\theta_0, x)$).

- aNBK: the adapted Bayesian kernel method in our scaling limit. This corresponds to regression with the adapted Φ^L kernel.
- aNTK: the adapted NTK kernel in our feature learning scaling limit. This corresponds to regression with the adapted K kernel.