A self consistent theory of Gaussian Processes captures feature learning effects in finite CNNs

Anonymous Author(s)
Affiliation
Address
email

Abstract

Deep neural networks (DNNs) in the infinite width/channel limit have received much attention recently, as they provide a clear analytical window to deep learning via mappings to Gaussian Processes (GPs). Despite its theoretical appeal, this viewpoint lacks a crucial ingredient of deep learning in finite DNNs, laying at the heart of their success — *feature learning*. Here we consider DNNs trained with noisy gradient descent on a large training set and derive a self consistent Gaussian Process theory accounting for strong finite-DNN and feature learning effects. Applying this to a toy model of a two-layer linear convolutional neural network (CNN) shows good agreement with experiments. We further identify, both analytical and numerically, a sharp transition between a feature learning regime and a lazy learning regime in this model. Strong finite-DNN effects are also derived for a non-linear two-layer fully connected network. Our self consistent theory provides a rich and versatile analytical framework for studying feature learning and other non-lazy effects in finite DNNs.

1 Introduction

The correspondence between Gaussian Processes (GPs) and deep neural networks (DNNs) has been instrumental in advancing our understanding of these complex algorithms. Early results related randomly initialized strongly over-parameterized DNNs with GP priors [31, 22, 28]. More recent results considered training using gradient flow (or noisy gradients), where DNNs map to Bayesian inference on GPs governed by the neural tangent kernel [19, 23] (or the NNGP kernel [30]). These correspondences carry over to a wide variety of architectures, going beyond fully connected networks (FCNs) to convolutional neural networks (CNNs) [3, 32], recurrent neural networks (RNNs) [1] and even attention networks [18]. They provide us with closed analytical expressions for the outputs of strongly over-parameterized trained DNNs, which have been used to make accurate predictions for DNN learning curves [10, 8, 7].

Despite their theoretical appeal, GPs are unable to capture *feature learning* [40, 39], which is a well-observed key property of trained DNNs. Indeed, it was noticed [19] that as the width tends to infinity, the neural tangent kernel (NTK) tends to a constant kernel that does not evolve during training and weights in hidden layers change infinitesimally from their initialization values. This regime of training was thus dubbed *lazy training* [9]. Other studies showed that for CNNs trained on image classification tasks, the feature learning regime generally tends to outperform the lazy regime [14, 13, 21]. Clearly, working in the feature learning regime is also crucial for performing *transfer learning* [38, 39].

It is therefore desirable to have a theoretical approach to deep learning which enjoys the generality and analytical power of GPs while capturing feature learning effects in finite DNNs. Here we make several contributions towards this goal:

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

1. We show that the mean predictor of a finite DNN trained on a large data set with noisy gradients, weight decay and MSE loss, can be obtained from GP regression on a shifted target (§3). Central to our approach is a non-linear self consistent equation involving the higher cumulants of the finite DNN (at initialization) which predicts this target shift.
2. Using this machinery on a toy model of a two-layer linear CNN in a teacher-student setting, we derive explicit analytical predictions which are in very good agreement with experiments even well away from the GP/lazy-learning regime (large number of channels, C) thus accounting for *strong* finite-DNN corrections (§4). Similar strong correction to GPs, yielding qualitative improvements in performance, are demonstrated for the quadratic two-layer fully connected model of Ref. [26].
3. We show how our framework can be used to study statistical properties of weights in hidden layers. In particular, in the CNN toy model, we identify, both analytically and numerically, a sharp transition between a feature learning phase and a lazy learning phase (§4.4). We define the feature learning phase as the regime where the features of the teacher network leave a clear signature in the spectrum of the student's hidden weights posterior covariance matrix. In essence, this phase transition is analogous to the transition associated with the recovery of a low-rank signal matrix from a noisy matrix taken from the Wishart ensemble, when varying the strength of the low-rank component [5].

1.1 Additional related work

Several previous papers derived leading order finite-DNN corrections to the GP results [30, 37, 12]. While these results are extendable to any order in perturbation theory, such high order expansions have not been studied much, perhaps due to their complexity. In contrast, we develop an analytically tractable non-perturbative approach which we find crucial for obtaining non-negligible feature learning and associated performance enhancement effects.

Previous works [14, 13] studied how the behavior of infinite DNNs depends on the scaling of the top layer weights with its width. In [38] it is shown that the standard and NTK parameterizations of a neural network do not admit an infinite-width limit that can learn features, and instead suggest an alternative parameterization which can learn features in this limit. While unifying various viewpoints on infinite DNNs, this approach does not immediately lend itself to analytical analysis of the kind proposed here.

Several works [34, 25, 15, 2] show that finite width models can generalize either better or worse than their infinite width counterparts, and provide examples where the relative performance depends on the optimization details, the DNN architecture and the statistics of the data. Here we demonstrate analytically that finite DNNs outperform their GP counterparts when the latter have a prior that lacks some constraint found in the data (e.g. positive-definiteness [26], weight sharing [32]).

Deep linear networks (FCNs and CNNs) similar to our toy example have been studied in the literature [4, 35, 20, 16]. These studies use different approaches and assumptions and do not discuss the target shift mechanism which applies also for non-linear CNNs. In addition, their analytical results hinge strongly on linearity whereas our approach could be useful whenever several leading cumulants of the DNN output are known.

A concurrent work [41] derived exact expressions for the output priors of finite FCNs induced by Gaussian priors over their weights. However, these results only apply to the limited case of a *prior* over a *single* training point and only for a FCN. In contrast, our approach applies to the setting of a large training set, it is not restricted to FCNs and yields results for the posterior predictions, not the prior. Focusing on deep linear fully connected DNNs, recent work [24] derived analytical finite-width renormalization results for the GP kernel, by sequentially integrating out the weights of the DNN, starting from the output layer and working backwards towards the input. Our analytical approach, its scope, and the models studied here differ substantially from that work.

2 Preliminaries

We consider a fixed set of n training inputs $\{\mathbf{x}_\mu\}_{\mu=1}^n \subset \mathbb{R}^d$ and a single test point $\mathbf{x}_*$ over which we wish to model the distribution of the outputs of a DNN. We consider a generic DNN architecture where for simplicity we assume a scalar output $f(\mathbf{x}) \in \mathbb{R}$. The learnable parameters of the DNN

that determine its output, are collected into a single vector θ. We pack the outputs evaluated on the training set and on the test point into a vector $\vec{f} \equiv (f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n), f(\mathbf{x}_{n+1})) \in \mathbb{R}^{n+1}$, where we denoted the test point as $\mathbf{x}_* = \mathbf{x}_{n+1}$. We train the DNNs using *full-batch* gradient decent with weight decay and external white Gaussian noise. The discrete dynamics of the parameters are thus

$$\theta_{t+1} - \theta_t = -\left(\gamma\theta_t + \nabla_\theta \mathcal{L}\left(f_\theta\right)\right)\eta + 2\sigma\sqrt{\eta}\xi_t \tag{1}$$

where θ_t is the vector of all network parameters at time step t, γ is the strength of the weight decay, $\mathcal{L}(f_\theta)$ is the loss as a function of the DNN output f_θ (where we have emphasized the dependence on the parameters θ), σ is the magnitude of noise, η is the learning rate and $\xi_t \sim \mathcal{N}(0, I)$. As $\eta \to 0$ this discrete-time dynamics converge to the continuous-time Langevin equation given by $\dot{\theta}(t) = -\nabla_\theta\left(\frac{\gamma}{2}||\theta(t)||^2 + \mathcal{L}(f_\theta)\right) + 2\sigma\xi(t)$ with $\langle \xi_i(t)\xi_j(t')\rangle = \delta_{ij}\delta(t-t')$, so that as $t \to \infty$ the DNN parameters θ will be sampled from the equilibrium Gibbs distribution $P(\theta)$.

As shown in [30], the parameter distribution $P(\theta)$ induces a posterior distribution over the trained DNN outputs $P(\vec{f})$ with the following partition function:

$$Z\left(\vec{J}\right) = \int d\vec{f} P_0\left(\vec{f}\right) \exp\left(-\frac{1}{2\sigma^2}\mathcal{L}\left(\{f_\mu\}_{\mu=1}^n, \{g_\mu\}_{\mu=1}^n\right) + \sum_{\mu=1}^{n+1} J_\mu f_\mu\right) \tag{2}$$

Here $P_0(\vec{f})$ is the prior generated by the finite-DNN with θ drawn from $\mathcal{N}(0, 2\sigma^2/\gamma)$ where the weight decay γ may be layer-dependent, $\{g_\mu\}_{\mu=1}^n$ are the training targets and $\vec{J}$ are source terms used to calculate the statistics of f. We keep the loss function $\mathcal{L}$ arbitrary at this point, committing to a specific choice in the next section. As standard [17], to calculate the posterior mean at any of the training points or the test point $\mathbf{x}_{n+1}$ from this partition function one uses

$$\forall \mu \in \{1, \ldots, n+1\}: \qquad \langle f_\mu \rangle = \partial_{J_\mu} \log Z\left(\vec{J}\right)\Big|_{\vec{J}=\vec{0}} \tag{3}$$

3 A self consistent theory for the posterior mean and covariance

In this section we show that for a large training set, the posterior mean predictor (Eq. 3) amounts to GP regression on a shifted target ($g_\mu \to g_\mu - \Delta g_\mu$). This shift to the target (Δg_μ) is determined by solving certain self-consistent equations involving the cumulants of the prior $P_0(\vec{f})$. For concreteness, we focus here on the MSE loss $\mathcal{L} = \sum_{\mu=1}^n (f_\mu - g_\mu)^2$ and comment on extensions to other losses, e.g. the cross entropy, in App. C. To this end, consider first the prior of the output of a finite DNN. Using standard manipulations (see App. A), it can be expressed as follows

$$P_0\left(\vec{f}\right) \propto \int_{\mathbb{R}^{n+1}} d\vec{t} \exp\left(-\sum_{\mu=1}^{n+1} it_\mu f_\mu + \sum_{r=2}^{\infty} \frac{1}{r!} \sum_{\mu_1,\ldots,\mu_r=1}^{n+1} \kappa_{\mu_1,\ldots,\mu_r} it_{\mu_1}\cdots it_{\mu_r}\right) \tag{4}$$

where $\kappa_{\mu_1,\ldots,\mu_r}$ is the r'th multivariate cumulant of $P_0(\vec{f})$ [29]. The second term in the exponent is the cumulant generating function ($\mathcal{C}$) corresponding to P_0. As discussed in App. B and Ref. [30], for standard initialization protocols the r'th cumulant will scale as $1/C^{(r/2-1)}$, where C controls the over-parameterization. The second ($r = 2$) cumulant which is C-independent, describes the NNGP kernel of the finite DNN and is denoted by $K(\mathbf{x}_{\mu_1}, \mathbf{x}_{\mu_2}) = \kappa_{\mu_1,\mu_2}$.

Consider first the case of $C \to \infty$ [22, 28, 31] where all $r > 2$ cumulants vanish. Here one can explicitly perform the integration in Eq. 4 to obtain the standard GP prior $P_0\left(\vec{f}\right) \propto \exp\left(-\frac{1}{2}\sum_{\mu_1,\mu_2=1}^{n+1} \kappa_{\mu_1,\mu_2} f_{\mu_1} f_{\mu_2}\right)$. Plugging this prior into Eq. 2 with MSE loss, one recovers standard GP regression formulas [33]. In particular, the predictive mean at $\mathbf{x}_*$ is: $\langle f(\mathbf{x}_*)\rangle = \sum_{\mu,\nu=1}^n K_\mu^* \tilde{K}_{\mu\nu}^{-1} g_\nu$ where $K_\mu^* = K(\mathbf{x}_*, \mathbf{x}_\mu)$ and $\tilde{K}_{\mu\nu} = K(\mathbf{x}_\mu, \mathbf{x}_\nu) + \sigma^2\delta_{\mu\nu}$. Another set of quantities we shall find useful are the *discrepancies in GP prediction*, which for the training set read

$$\boxed{\forall \mu \in \{1, \ldots, n\}: \quad \langle \hat{\delta} g_\mu \rangle \equiv g_\mu - \langle f(\mathbf{x}_\mu)\rangle = g_\mu - \sum_{\nu,\nu=1}^{n} K_{\mu\nu'} \tilde{K}_{\nu,\nu'}^{-1} g_\nu} \tag{5}$$

Saddle point approximation for the mean predictor. For a DNN with finite C, the prior $P_0(\vec{f})$ will no longer be Gaussian and cumulants with $r > 2$ would contribute. This renders the partition function in Eq. 2 intractable and so some approximation is needed to make progress. To this end we note that f can be integrated out (see App. A.1) to yield a partition function of the form

$$Z\left(\vec{J}\right) \propto \int_{\mathbb{R}^n} dt_1 \cdots dt_n e^{-\mathcal{S}(\vec{t},\vec{J})} \tag{6}$$

where $\mathcal{S}(\vec{t},\vec{J})$ is the action whose exact form is given in Eq. A.14. Interestingly, the it_μ variables appearing above are closely related to the discrepancies $\hat{\delta}g_\mu$, in particular $\langle it_\mu \rangle = \langle \hat{\delta} g_\mu \rangle / \sigma^2$.

To proceed analytically we adopt the *saddle point (SP) approximation* [11] which, as argued in App. A.4, relies on the fact that the non-linear terms in the action comprise of a sum of many it_μ's. Given that this sum is dominated by collective effects coming from all data points, expanding $\mathcal{S}(\vec{t},\vec{J})$ around the saddle point yields terms with increasingly negative powers of n.

For the training points $\mu \in \{1, \ldots, n\}$, taking the saddle point approximation amounts to setting $\partial_{it_\mu} \mathcal{S}\left(\vec{t},\vec{J}\right)\Big|_{\vec{J}=\vec{0}} = 0$. This yields a set of equations that has precisely the form of Eq. 5, but where the target is shifted as $g_\nu \to g_\nu - \Delta g_\nu$ and the target shift is determined self consistently by

$$\boxed{\Delta g_\nu = \sum_{r=3}^{\infty} \frac{1}{(r-1)!} \sum_{\mu_1,\ldots,\mu_{r-1}=1}^{n} \kappa_{\nu,\mu_1,\ldots,\mu_{r-1}} \left\langle \sigma^{-2} \hat{\delta} g_{\mu_1} \right\rangle \cdots \left\langle \sigma^{-2} \hat{\delta} g_{\mu_{r-1}} \right\rangle} \tag{7}$$

Equation 7 is thus an implicit equation for Δg_ν involving all training points, and it holds for the training set and the test point $\nu \in \{1, \ldots, n+1\}$. Once solved, either analytically or numerically, one calculates the predictions on the test point via

$$\boxed{\langle f_* \rangle = \Delta g_* + \sum_{\mu,\nu=1}^{n} K^*_\mu \tilde{K}^{-1}_{\mu\nu} \left(g_\nu - \Delta g_\nu\right)} \tag{8}$$

Equation 5 with $g_\nu \to g_\nu - \Delta g_\nu$ along with Eqs. 7 and 8 are the first main result of this paper. Viewed as an algorithm, the procedure to predict the finite DNN's output on a test point $\mathbf{x}_*$ is as follows: we shift the target in Eq. 5 as $g \to g - \Delta g$ with Δg as in Eq. 7, arriving at a closed equation for the average discrepancies $\langle \hat{\delta} g_\mu \rangle$ on the training set. For some models, the cumulants $\kappa_{\nu,\mu_2,\ldots,\mu_r}$ can be computed for any order r and it can be possible to sum the entire series, while for other models several leading cumulants might already give a reasonable approximation due to their $1/C^{r/2-1}$ scaling. The resulting coupled non-linear equations can then be solved numerically, to obtain Δg_μ from which predictions on the test point are calculated using Eq. 8.

Notwithstanding, solving such equations analytically is challenging and one of our main goals here is to provide concrete analytical insights. Thus, in §4.2 we propose an additional approximation wherein to leading order we replace all summations over data-points with integrals over the measure from which the data-set is drawn. This approximation, taken in some cases beyond leading order as in Ref. [10], will yield analytically tractable equations which we solve for two simple toy models, one of a linear CNN and the other of a non-linear FCN.

Saddle point plus Gaussian fluctuations for the posterior covariance. The SP approximation can be extended to compute the predictor variance by expanding the action $\mathcal{S}$ to quadratic order in it_μ around the SP value (see App. A.3). Due to the saddle-point being an extremum this leads to $\mathcal{S} \approx \mathcal{S}_{\text{SP}} + \frac{1}{2} t_\mu A^{-1}_{\mu\nu} t_\nu$. This leaves the previous SP approximation for the posterior mean on the training set unaffected (since the mean and maximizer of a Gaussian coincide), but is necessary to get sensible results for the posterior covariance. Empirically, in the toy models we considered in §4 we find that the finite DNN corrections to the variance are much less pronounced than those for the mean. Using the standard Gaussian integration formula, one finds that $A_{\mu\nu}$ is the covariance matrix of it_μ. Performing such an expansion one finds

$$\begin{aligned} A^{-1}_{\mu\nu} &= -\left(\sigma^2 \delta_{\mu\nu} + K_{\mu\nu} + \Delta K_{\mu\nu}\right) \\ \Delta K_{\mu\nu} &= \partial_{it_\mu} \Delta g_\nu \left(it_1, \ldots, it_n\right) \end{aligned} \tag{9}$$

where the it_μ on the r.h.s. are those of the saddle point. This gives an expression for the posterior covariance matrix on the training set:

$$\Sigma_{\mu\nu} = \langle f_\mu f_\nu \rangle - \langle f_\mu \rangle \langle f_\nu \rangle = -\sigma^4 \left[\sigma^2 I + K + \Delta K\right]^{-1}_{\mu\nu} + \sigma^2 \delta_{\mu\nu} \tag{10}$$

where the r.h.s. coincides with the posterior covariance of a GP with a kernel equal to $K + \Delta K$ [33]. The variance on the test point is given by (repeating indices are summed over the training set)

$$\Sigma_{**} = K_{**} - K^*_\mu A^{-1}_{\mu\nu} K^*_\nu + \left\langle \partial^2_{it_*} \tilde{\mathcal{C}}|_{it_*=0} \right\rangle + 2\left(\left\langle \Delta g_* K^*_\mu it_\mu \right\rangle - \langle \Delta g_* \rangle \left\langle K^*_\mu it_\mu \right\rangle\right) + \mathrm{Var}\left(\Delta g_*\right) \tag{11}$$

where here Δg_* is as in Eq. 7 but where the $\left\langle \sigma^{-2} \hat{\delta} g_\mu \right\rangle$'s are replaced the it_μ's that have Gaussian fluctuations, and $\tilde{\mathcal{C}}$ is $\mathcal{C}$ without the second cumulant (see App. A.1). The first two terms in Eq. 11 yield the standard result for the GP posterior covariance matrix on a test point [33], for the case of $\Delta K = 0$ (see Eq. 9). The rest of the terms can be evaluated by the SP plus Gaussian fluctuations approximation, where the details would depend on the model at hand.

4 Two toy models

4.1 The two layer linear CNN and its properties

Here we define a teacher-student toy model showing several qualitative real-world aspects of feature learning and analyze it via our self-consistent shifted target approach. Concretely, we consider the simplest student CNN $f(\mathbf{x})$, having one hidden layer with linear activation, and a corresponding teacher CNN, $g(\mathbf{x})$

$$f\left(\mathbf{x}\right) = \sum_{i=1}^{N} \sum_{c=1}^{C} a_{i,c} \mathbf{w}_c \cdot \tilde{\mathbf{x}}_i \qquad g\left(\mathbf{x}\right) = \sum_{i=1}^{N} \sum_{c=1}^{C^*} a^*_{i,c} \mathbf{w}^*_c \cdot \tilde{\mathbf{x}}_i \tag{12}$$

This describes a CNN that performs 1-dimensional convolution where the convolutional weights for each channel are $\mathbf{w}_c \in \mathbb{R}^S$. These are dotted with a convolutional window of the input $\tilde{\mathbf{x}}_i = \left(x_{S(i-1)+1}, \ldots, x_{S\cdot i}\right)^\mathsf{T} \in \mathbb{R}^S$ and there are no overlaps between them so that $\mathbf{x} = \left(x_1, \ldots, x_{N\cdot S}\right)^\mathsf{T} = \left(\tilde{\mathbf{x}}_1, \ldots, \tilde{\mathbf{x}}_N\right)^\mathsf{T} \in \mathbb{R}^{N\cdot S}$. Namely, the input dimension is $d = NS$, where N is the number of (non-overlapping) convolutional windows, S is the stride of the conv-kernel and it is also the length of the conv-kernel, hence there is no overlap between the strides. The inputs $\mathbf{x}$ are sampled from $\mathcal{N}(0, I_d)$.

Despite its simplicity, this model distils several key differences between feature learning models and lazy learning or GP models. Due to the lack of pooling layers, the GP associated with the student fails to take advantage of the weight sharing property of the underlying CNN [32]. In fact, here it coincides with a GP of a fully-connected DNN which is quite inappropriate for the task. We thus expect that the finite network will have good performance already for $n = C^*(N + S)$ whereas the GP will need n of order of the dimension (NS) to learn well [10]. Thus, for $N + S \ll NS$ there should be a broad regime in the value of n where the finite network substantially outperforms the corresponding GP. We later show (§4.4) that this performance boost over GP is due to feature learning, as one may expect.

Conveniently, the cumulants of the student DNN of any order can be worked out exactly. Assuming γ and σ^2 of the noisy GD training are chosen such that[1] $a_{i,c} \sim \mathcal{N}\left(0, \sigma_a^2/CN\right)$, $\quad \mathbf{w}_c \sim \mathcal{N}\left(\mathbf{0}, \frac{\sigma_w^2}{S} I_S\right)$ (and similarly for the teacher DNN) the covariance function for the associated GP reads

$$K\left(\mathbf{x}, \mathbf{x}'\right) = \frac{\sigma_a^2 \sigma_w^2}{NS} \sum_{i=1}^{N} \tilde{\mathbf{x}}_i^\mathsf{T} \tilde{\mathbf{x}}'_i = \frac{\sigma_a^2 \sigma_w^2}{NS} \mathbf{x}_i^\mathsf{T} \mathbf{x}'_i \tag{13}$$

Denoting $\lambda := \frac{\sigma_a^2}{N} \frac{\sigma_w^2}{S}$, the even cumulant of arbitrary order $2m$ is (see App. F.1):

$$\kappa_{2m}\left(\mathbf{x}_1, \ldots, \mathbf{x}_{2m}\right) = \frac{\lambda^m}{C^{m-1}} \sum_{i_1, \ldots, i_m = 1}^{N} \left(\bullet_{i_1}, \bullet_{i_2}\right) \cdots \left(\bullet_{i_{m-2}}, \bullet_{i_{m-1}}\right) \left(\bullet_{i_{m-1}}, \bullet_{i_m}\right) \cdots \left[(2m-1)!\right] \tag{14}$$

[1] Generically this requires C dependent and layer dependent weight decay.

while all odd cumulants vanish due to the sign flip symmetry of the last layer. In this notation, we mean that the $\bullet$'s stand for integers in $\{1, \ldots, 2m\}$ and e.g. $(1_{i_1}, 2_{i_2}) \equiv \left(\tilde{\mathbf{x}}^1_{i_1} \cdot \tilde{\mathbf{x}}^2_{i_2}\right)$ and the bracket notation $[(2m-1)!]$ stands for the number of ways to pair the integers $\{1, ..., 2m\}$ into the above form. This result can then be plugged in 7 to obtain the self consistent (saddle point) equations on the training set. See App. A.4 for a convergence criterion for the saddle point, supporting its application here.

4.2 Self consistent equation in the limit of a large training set

In §3 our description of the self consistent equations was for a finite and fixed training set. Further analytical insight can be gained if we consider the limit of a large training set, known in the GP literature as the Equivalent Kernel (EK) limit [33, 36]. For a short review of this topic, see App. D. In essence, in the EK limit we replace the discrete sums over a specific draw of training set, as in Eqs. 5, 7, 8, with integrals over the entire input distribution $\mu(\mathbf{x})$. Given a kernel that admits a spectral decomposition in terms of its eigenvalues and eigenfunctions: $K(\mathbf{x}, \mathbf{x}') = \sum_s \lambda_s \psi_s(\mathbf{x}) \psi_s(\mathbf{x}')$, the standard result for the GP posterior mean at a test point is approximated by [33]

$$\langle f(\mathbf{x}_*) \rangle = \int d\mu(\mathbf{x})\, h(\mathbf{x}_*, \mathbf{x})\, g(\mathbf{x})\,; \qquad h(\mathbf{x}_*, \mathbf{x}) = \sum_s \frac{\lambda_s}{\lambda_s + \sigma^2/n} \psi_s(\mathbf{x}_*)\, \psi_s(\mathbf{x}) \tag{15}$$

This has several advantages, already at the level of GP analysis. From a theoretical point of view, the integral expressions retain the symmetries of the kernel $K(\mathbf{x}, \mathbf{x}')$ unlike the discrete sums that ruin these symmetries. Also, Eq. 15 does not involve computing the inverse matrix $\tilde{K}^{-1}$ which is costly for large matrices.

In the context of our theory, the EK limit allows for a derivation of a simple analytical form for the self consistent equations. As shown in App. E.1 in our toy CNN both Δg and $\hat{\delta g}$ become linear in the target. Thus the self-consistent equations can be reduced to a single equation governing the proportionality factor (α) between $\hat{\delta g}$ and g ($\hat{\delta} g = \alpha g$). Thus starting from the general self consistent equations, 5, 7, 8, taking their EK limit, and plugging in the general cumulant for our toy model (14) we arrive at the following equation for α

$$\alpha = \frac{\sigma^2/n}{\lambda + \sigma^2/n} + (1-q)\frac{\lambda}{\lambda + \sigma^2/n} - q\frac{\lambda}{\lambda + \sigma^2/n}\frac{\lambda^2}{C}\left(\frac{\alpha}{\sigma^2/n}\right)^3 \left[1 - \frac{\lambda}{C}\left(\frac{\alpha}{\sigma^2/n}\right)^2\right]^{-1} \tag{16}$$

Here set for simplicity $\sigma_a^2 = 1 = \sigma_w^2$ we have $\lambda = 1/(NS)$ and also introduced the constant $q \equiv \lambda^{-1}(1 - \hat{\alpha}^{\text{train}}_{\text{GP}})(1 + \sigma^2/n)$ where $\hat{\alpha}_{\text{GP}}$ is computed using the empirical GP predictions: $\hat{\alpha}^{\text{train}}_{\text{GP}} \equiv 1 - \left(\sum_{\mu=1}^n f^{\text{GP}}_\mu g_\mu\right) / \left(\sum_{\mu=1}^n g_\mu^2\right)$ or analytically in the perturbation theory approach developed in [10]. The quantity q has an interpretation as $1/n$ corrections to the EK approximation [10] but here can be considered as a fitting parameter. It is non-negative and is typically $O(1)$; for more details and analytical estimates see App. E.2.

Equation 16 is the second main analytical result of this work. It simplifies the highly non-linear inference problem to a single equation that embodies strong non-linear finite-DNN effect and feature learning (see also §4.4). In practice, to compute α_{test} we numerically solve 16 using q_{train} for the training set to get α_{train}, and then set $\alpha = \alpha_{\text{train}}$ in the r.h.s. of 16 but use $q = q_{\text{test}}$. Equation 16 can also be used to bound α analytically on both the training set and test point, given the reasonable assumption that α changes continuously with C. Indeed, at large C the pole in this equation lays at $\alpha_{\text{pole}} = (\sigma^2/n)(C/\lambda)^{1/2} \gg 1$ whereas $\alpha \approx \alpha_{\text{GP}} < \alpha_{\text{pole}}$. As C diminishes, continuity implies that α must remain smaller than α_{pole}. The latter decays as $\sigma^2\sqrt{CNS}/n$ implying that the amount of data required for good performance scales as $\sqrt{CNS}$ rather than as NS in the GP case.

4.3 Numerical verification

In this section we numerically verify the predictions of the self consistent theory of Sec. §4.2, by training linear shallow student CNNs on a teacher with $C^* = 1$ as in Eq. 12, using noisy gradients as in Eq. 1, and averaging their outputs across noise realizations and across dynamics after reaching equilibrium.

For simplicity we used $N = S$ and $n \in \{62, 200, 650\}$, $S \in \{15, 30, 60\}$ so that $n \propto S^{1.7}$. The latter scaling places us in the poorly performing regime of the associated GP while allowing good performance of the CNN. Indeed, as aforementioned, the GP here requires n of the scale of $\lambda^{-1} = NS = O(S^2)$ for good performance [10], while the CNN requires n of scale of the number of parameters ($C(N + S) = O(S)$).

The results are shown in Fig 1 where we compare the theoretical predictions given by the solutions of the self consistent equation (16) to the empirical values of α obtained by training actual CNNs and averaging their outputs across the ensemble.

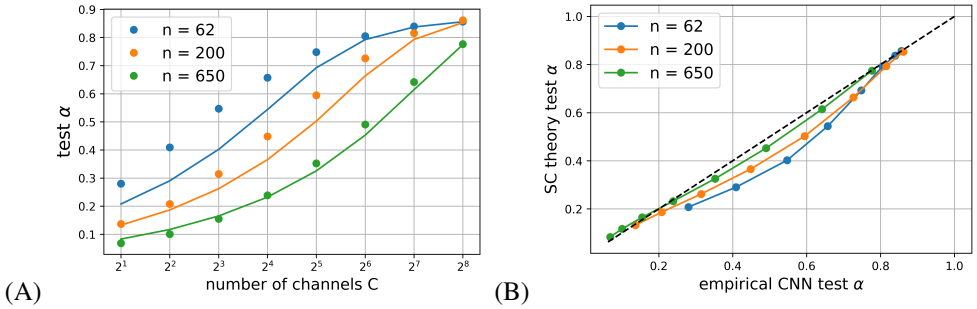

Figure 1: **(A)** The CNNs' cosine distance α, defined by $\langle f \rangle = (1 - \alpha)g$ between the ensemble-averaged prediction $\langle f \rangle$ and ground truth g plotted vs. number of channels C for the test set (for the train set, see App. H.2). As n increases, the solution of the self consistent equation 16 (solid line) yields an increasingly accurate prediction of these empirical values (dots). **(B)** Same data as in (A), presented as empirical α vs. predicted α. As n grows, the two converge to the identity line (dashed black line). Solid lines connecting the dots here are merely for visualization purposes.

4.4 Feature learning phase transition in the CNN model

At this point there is evidence that our self-consistent shifted target approach works well within the feature learning regime of the toy model. Indeed GP is sub-optimal here, since it does not represent the CNN's weight sharing present in the teacher network. Weight sharing is intimately tied with feature learning in the first layer, since it aggregates the information coming from all convolutional windows to refine a single set of repeating convolution-filters. Empirically, we observed a large performance gap of finite C CNNs over the infinite-C (GP) limit, which was also observed previously in more realistic settings [21, 14, 32]. Taken together with the existence of a clear feature in the teacher, a natural explanation for this performance gap is that feature learning, which is completely absent in GPs, plays a major role in the behavior of finite C CNNs.

To analyze this we wish to track how the feature of the teacher $\mathbf{w}^*$ are reflected in the student network's first layer weights $\mathbf{w}_c$ across training time (after reaching equilibrium) and across training realizations. However, as our formalism deals with *ensembles of DNNs*, computing averages of $\mathbf{w}_c$ with respect to these ensembles would simply give zero. Indeed, the chance of a DNN with specific parameters $\theta = \{a_{i,c}, \mathbf{w}_c\}$ appearing is the same as that of $-\theta$. Consequently, to detect feature learning the first reasonable object to examine is the empirical covariance matrix $\Sigma_W = \frac{S}{C} W W^\mathsf{T}$, where the matrix $W \in \mathbb{R}^{S \times C}$ has $\mathbf{w}_c$ as its c'th column. This Σ_W is invariant under such a change of signs and provides important information on the statistics of $\mathbf{w}_c$.

As shown in App. G, using our field-theory or function-space formulation, we find that to leading order in $1/C$ the ensemble average of the empirical covariance matrix, for a teacher with a single feature $\mathbf{w}^*$, is

$$\langle [\Sigma_W]_{ss'} \rangle = \left(1 + \left(\frac{1}{\lambda} + \frac{n}{\sigma^2}\right)^{-1}\right) \delta_{ss'} + \frac{2}{C} \frac{\lambda}{(\lambda + \sigma^2/n)^2} w_s^* w_{s'}^* + O(1/C^2) \tag{17}$$

A first conclusion that could be drawn here, is that given access to an ensemble of such trained CNNs, feature learning happens for any finite C as a statistical property. We turn to discuss the more common setting where one wishes to use the features learned by a specific randomly chosen CNN from this ensemble.

To this end, we follow Ref. [27] and model Σ_W as a Wishart matrix with a rank-one perturbation. The variance of the matrix and details of the rank one perturbation are then determined by the above equation. Consequently the eigenvalue distribution is expected to follow a spiked Marchenko-Pastur (MP), which was studied extensively in [6]. To test this modeling assumption, for each snapshot of training time (after reaching equilibrium) and noise realization we compute Σ_W's eigenvalues and aggregate these across the ensemble. In Fig. 2 we plot the resulting empirical spectral distribution for varying values of C while keeping S fixed. Note that, differently from the usual spiked-MP model, varying C here changes both the distribution of the MP bulk (which is determined by the ratio S/C) as well as the strength of the low-rank perturbation.

Our main finding is a phase transition between two regimes which becomes sharp as one takes $n, S \to \infty$. In the regime of large C the eigenvalue distribution of Σ_W is indistinguishable from the MP distribution, whereas in the regime of small C an outlier eigenvalue λ_m departs from the support of the bulk MP distribution and the associated top eigenvector has a non-zero overlap with $\mathbf{w}^*$, see Fig. 2. We refer to the latter as the feature-learning regime, since the feature $\mathbf{w}^*$ is manifested in the spectrum of the students weights, whereas the former is the non-feature learning regime. We use the quantity $\mathcal{Q} \equiv \mathbf{w}^{*\mathsf{T}} \Sigma_W \mathbf{w}^*$ as a surrogate for λ_m, as it is valid on both sides of the transition. Having established the correspondence to the MP plus low rank model, we can use the results of [6] to find the exact location of the phase transition, which occurs at the critical value C_{crit} given by

$$C_{\text{crit}} = \frac{4}{S\left(S^{-1} + (\sigma^2/n)\, S\right)^4} \left(1 + \left(S^2 + \frac{n}{\sigma^2}\right)^{-1}\right) + O\left(1 + \left(\frac{1}{\lambda} + \frac{n}{\sigma^2}\right)^{-1}\right) \tag{18}$$

where we assumed for simplicity $N = S$ so that $\lambda = S^{-2}$.

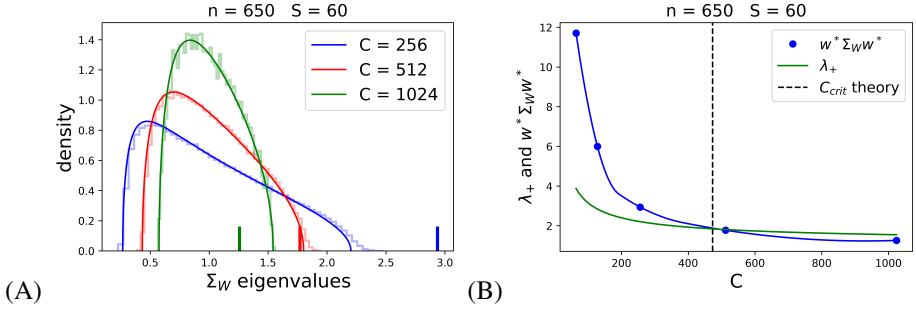

Figure 2: **(A)** Aggregated histograms of Σ_W eigenvalues where $\Sigma_W = \frac{S}{C} W W^{\mathsf{T}}$ is the normalized empirical covariance matrix of the hidden layer weights during training. Different colors indicate varying number of channels, C. Solid smooth lines indicate the corresponding Marchenko-Pastur (MP) distributions with support on $[\lambda_-, \lambda_+]$ where: $\lambda_\pm = \left(1 \pm \sqrt{S/C}\right)^2$. The quantity $\mathcal{Q} \equiv \mathbf{w}^{*\mathsf{T}} \Sigma_W \mathbf{w}^*$, which correlates with the SNR of the feature $\mathbf{w}^*$, is represented by thick short bars. For large C, $\mathcal{Q}$ remains within the MP bulk whereas for small C it pops out. **(B)** The theoretical λ_+ curve and interpolated curve of $\mathcal{Q}$ and intersect very close to the theoretically predicted value given in Eq. 18, here given by $C_{\text{crit}} = 473$ (dashed vertical line).

4.5 Two-layer FCN with average pooling and quadratic activations

Another setting where GPs are expected to under-perform finite-DNNs is the case of quadratic fully connected teacher and student DNNs where the teacher is rank-1, also known as the phase retrieval problem [26]. Here we consider some positive target of the form $g(\mathbf{x}) = (\mathbf{w}_* \cdot \mathbf{x})^2 - \sigma_w^2 ||\mathbf{x}||^2$ where $\mathbf{w}_*, \mathbf{x} \in \mathbb{R}^d$ and a student DNN given by $f(\mathbf{x}) = \sum_{m=1}^{M} (\mathbf{w}_m \cdot \mathbf{x})^2 - \sigma_w^2 ||\mathbf{x}||^2$. We consider training this DNN on n train points $\{\mathbf{x}_\mu\}_{\mu=1}^n$ using noisy GD training with weight decay $\gamma = 2M\sigma^2/\sigma_w^2$.

Similarly to the previous toy model, here too the GP associated with the student at large M (and finite σ^2) overlooks a qualitative feature of the finite DNN — the fact that the first term in $f(\mathbf{x})$ is non-negative. Interestingly, this feature provides a strong performance boost [26] in the $\sigma^2 \to 0$ limit compared to the associated GP. Namely the DNN, even at large M, performs well for $n > 2d$ [26] whereas the associated GP is expected to work well only for $n = O(d^2)$ [10].

We wish to solve for the predictions of this model with our self consistent GP based approach. As shown in App. I, the cumulants of this model can be obtained from the following cumulant generating function

$$\mathcal{C}(t_1, ..., t_{n+1}) = -\frac{M}{2} \operatorname{Tr} \left(\log \left[I - 2M^{-1}\sigma_w^2 \sum_\mu i t_\mu \mathbf{x}_\mu \mathbf{x}_\mu^\intercal \right] \right) - \sum_{\mu=1}^{n+1} i t_\mu \sigma_w^2 ||\mathbf{x}_\mu||^2 \tag{19}$$

The associated GP kernel is given by $K(\mathbf{x}_\mu, \mathbf{x}_\nu) = 2M^{-1}\sigma_w^4 (\mathbf{x}_\mu \cdot \mathbf{x}_\nu)^2$. Following this, the target shift equation, at the saddle point level, appears as

$$\Delta g_\nu = -\sum_\mu K(\mathbf{x}_\nu, \mathbf{x}_\mu) \frac{\hat{\delta} g_\mu}{\sigma^2} + \sigma_w^2 \mathbf{x}_\nu^\intercal \left[I - 2M^{-1}\sigma_w^2 \sum_\mu \frac{\hat{\delta} g_\mu}{\sigma^2} \mathbf{x}_\mu \mathbf{x}_\mu^\intercal \right]^{-1} \mathbf{x}_\nu - \sigma_w^2 ||\mathbf{x}_\nu||^2 \tag{20}$$

In App. I, we solve these equations numerically for $\sigma^2 = 10^{-5}$ and show that our approach captures the correct $n = 2d$ threshold value. An analytic solution of these equations at low σ^2 using EK or other continuum approximations is left for future work (see Refs. [10, 7, 8] for potential approaches). As a first step towards this goal, in App. I we consider the simpler case of $\sigma^2 = 1$ and derive the asymptotics of the learning curves which deviate strongly from those of GP for $M \ll d$.

5 Discussion

In this work we presented a correspondence between ensembles of finite DNNs trained with noisy gradients and GPs trained on a shifted target. The shift in the target can be found by solving a set of self consistent equations for which we give a general form. We found explicit expressions for these equations for the case of a 2-layer linear CNN and a non-linear FCN, and solved them analytically and numerically. For the former model, we performed numerical experiments on CNNs that agree well with our theory both in the GP regime *and well away from it*, i.e. for small number of channels C, thus accounting for strong finite C effects. For the latter model, the numerical solution of these equations capture a remarkable and subtle effect in these DNNs which the GP approach completely overlooks — the $n = 2d$ threshold value.

Considering feature learning in the CNN model, we found that averaging over ensembles of such networks always leads to a form of feature learning. Namely, the teacher always leaves a signature on the statistics of the student's weights. However, feature learning is usually considered at the level of a single DNN instance rather than an ensemble of DNNs. Focusing on this case, we show numerically that the eigenvalues of Σ_W, the student hidden weights covariance matrix, follow a Marchenko–Pastur distribution plus a rank-1 perturbation. We then use our approach to derive the critical number of channels C_{crit} below which the student is in a feature learning regime.

There are many directions for future research. Our toy models where chosen to be as simple as possible in order to demonstrate the essence of our theory on problems where lazy learning grossly under-performs finite-DNNs. Even within this setting, various extensions are interesting to consider such as adding more features to the teacher CNN (e.g. biases or a subset of linear functions which are more favorable), studying linear CNNs with overlapping convolutional windows, or deeper linear CNNs. As for non-linear CNNs, we believe it is possible to find the exact cumulants of any order for a variety of toy CNNs involving, for example, quadratic activation functions. For other cases it may be useful to develop methods for characterizing and approximating the cumulants.

More generally, we advocated here a physics-style methodology using approximations, self-consistency checks, and experimental tests. As DNNs are very complex experimental systems, we believe this mode of research is both appropriate and necessary. Nonetheless we hope the insights gained by our approach would help generate a richer and more relevant set of toy models on which mathematical proofs could be made.

References

[1] S. Alemohammad, Z. Wang, R. Balestriero, and R. Baraniuk. The recurrent neural tangent kernel. *arXiv preprint arXiv:2006.10246*, 2020.

[2] A. Andreassen and E. Dyer. Asymptotics of wide convolutional neural networks. *arXiv preprint arXiv:2008.08675*, 2020.

[3] S. Arora, S. S. Du, W. Hu, Z. Li, R. Salakhutdinov, and R. Wang. On Exact Computation with an Infinitely Wide Neural Net. *arXiv e-prints*, page arXiv:1904.11955, Apr 2019.

[4] P. Baldi and K. Hornik. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks*, 2(1):53–58, 1989.

[5] F. Benaych-Georges and R. R. Nadakuditi. The eigenvalues and eigenvectors of finite, low rank perturbations of large random matrices. *Advances in Mathematics*, 227(1):494–521, 2011.

[6] F. Benaych-Georges and R. R. Nadakuditi. The singular values and vectors of low rank perturbations of large rectangular random matrices. *Journal of Multivariate Analysis*, 111:120–135, 2012.

[7] B. Bordelon, A. Canatar, and C. Pehlevan. Spectrum dependent learning curves in kernel regression and wide neural networks, 2020.

[8] A. Canatar, B. Bordelon, and C. Pehlevan. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature Communications*, 12(1), May 2021.

[9] L. Chizat, E. Oyallon, and F. Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems*, pages 2937–2947, 2019.

[10] O. Cohen, O. Malka, and Z. Ringel. Learning Curves for Deep Neural Networks: A Gaussian Field Theory Perspective. *arXiv e-prints*, page arXiv:1906.05301, Jun 2019.

[11] H. E. Daniels. Saddlepoint Approximations in Statistics. *The Annals of Mathematical Statistics*, 25(4):631 – 650, 1954.

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

[13] M. Geiger, L. Petrini, and M. Wyart. Landscape and training regimes in deep learning. *Physics Reports*, 2021.

[14] M. Geiger, S. Spigler, A. Jacot, and M. Wyart. Disentangling feature and lazy training in deep neural networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2020(11):113301, 2020.

[15] B. Ghorbani, S. Mei, T. Misiakiewicz, and A. Montanari. When do neural networks outperform kernel methods? *arXiv preprint arXiv:2006.13409*, 2020.

[16] S. Gunasekar, J. Lee, D. Soudry, and N. Srebro. Implicit bias of gradient descent on linear convolutional networks. *arXiv preprint arXiv:1806.00468*, 2018.

[17] M. Helias and D. Dahmen. Statistical field theory for neural networks. *arXiv preprint arXiv:1901.10416*, 2019.

[18] J. Hron, Y. Bahri, J. Sohl-Dickstein, and R. Novak. Infinite attention: Nngp and ntk for deep attention networks. In *International Conference on Machine Learning*, pages 4376–4386. PMLR, 2020.

[19] A. Jacot, F. Gabriel, and C. Hongler. Neural Tangent Kernel: Convergence and Generalization in Neural Networks. *arXiv e-prints*, page arXiv:1806.07572, Jun 2018.

[20] A. K. Lampinen and S. Ganguli. An analytic theory of generalization dynamics and transfer learning in deep linear networks. *arXiv preprint arXiv:1809.10374*, 2018.

[21] J. Lee, S. S. Schoenholz, J. Pennington, B. Adlam, L. Xiao, R. Novak, and J. Sohl-Dickstein. Finite versus infinite neural networks: an empirical study. *arXiv preprint arXiv:2007.15801*, 2020.

[22] J. Lee, J. Sohl-dickstein, J. Pennington, R. Novak, S. Schoenholz, and Y. Bahri. Deep neural networks as gaussian processes. In *International Conference on Learning Representations*, 2018.

[23] J. Lee, L. Xiao, S. S. Schoenholz, Y. Bahri, R. Novak, J. Sohl-Dickstein, and J. Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. *arXiv preprint arXiv:1902.06720*, 2019.

[24] Q. Li and H. Sompolinsky. Statistical mechanics of deep linear neural networks: The back-propagating renormalization group. *arXiv preprint arXiv:2012.04030*, 2020.

[25] E. Malach, P. Kamath, E. Abbe, and N. Srebro. Quantifying the benefit of using differentiable learning over tangent kernels. *arXiv preprint arXiv:2103.01210*, 2021.

[26] S. S. Mannelli, E. Vanden-Eijnden, and L. Zdeborová. Optimization and generalization of shallow neural networks with quadratic activation functions. *arXiv preprint arXiv:2006.15459*, 2020.

[27] C. H. Martin and M. W. Mahoney. Implicit Self-Regularization in Deep Neural Networks: Evidence from Random Matrix Theory and Implications for Learning. *arXiv e-prints*, page arXiv:1810.01075, Oct. 2018.

[28] A. G. d. G. Matthews, M. Rowland, J. Hron, R. E. Turner, and Z. Ghahramani. Gaussian process behaviour in wide deep neural networks. *arXiv preprint arXiv:1804.11271*, 2018.

[29] P. Mccullagh. *Tensor Methods in Statistics*. Dover Books on Mathematics, 2017.

[30] G. Naveh, O. Ben-David, H. Sompolinsky, and Z. Ringel. Predicting the outputs of finite networks trained with noisy gradients. *arXiv preprint arXiv:2004.01190*, 2020.

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

[32] R. Novak, L. Xiao, J. Lee, Y. Bahri, G. Yang, D. A. Abolafia, J. Pennington, and J. Sohl-Dickstein. Bayesian Deep Convolutional Networks with Many Channels are Gaussian Processes. *arXiv e-prints*, page arXiv:1810.05148, Oct. 2018.

[33] C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005.

[34] M. Refinetti, S. Goldt, F. Krzakala, and L. Zdeborová. Classifying high-dimensional gaussian mixtures: Where kernel methods fail and neural networks succeed. *arXiv preprint arXiv:2102.11742*, 2021.

[35] A. M. Saxe, J. L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. *arXiv preprint arXiv:1312.6120*, 2013.

[36] P. Sollich and C. K. Williams. Understanding gaussian process regression using the equivalent kernel. In *International Workshop on Deterministic and Statistical Methods in Machine Learning*, pages 211–228. Springer, 2004.

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

[38] G. Yang and E. J. Hu. Feature learning in infinite-width neural networks. *arXiv preprint arXiv:2011.14522*, 2020.

[39] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson. How transferable are features in deep neural networks? In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[40] D. Yu, M. Seltzer, J. Li, J.-T. Huang, and F. Seide. Feature learning in deep neural networks - studies on speech recognition. In *International Conference on Learning Representations*, May 2013.

[41] J. A. Zavatone-Veth and C. Pehlevan. Exact priors of finite neural networks. *arXiv preprint arXiv:2104.11734*, 2021.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [No] We do not identify any such impact
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (b) Did you include complete proofs of all theoretical results? [Yes]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] due to a technical issue they are not included in the main text but will be included in the supplementary
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]