Mean-field Behaviour of Neural Tangent Kernel for Deep Neural Networks

Soufiane Hayou 1  Arnaud Doucet 1  Judith Rousseau 1

Abstract

Recent work by Jacot et al. (2018) has shown that training a neural network of any kind, with gradient descent in parameter space, is strongly related to kernel gradient descent in function space with respect to the Neural Tangent Kernel (NTK). Lee et al. (2019) built on this result by establishing that the output of a neural network trained using gradient descent can be approximated by a linear model for wide networks. In parallel, a recent line of studies (Schoenholz et al., 2017; Hayou et al., 2019) has suggested that a special initialization, known as the Edge of Chaos, improves training. In this paper, we connect these two concepts by quantifying the impact of the initialization and the activation function on the NTK when the network depth becomes large. In particular, we show that the performance of wide deep neural networks cannot be explained by the NTK regime. We also leverage our theoretical results to derive a learning rate passband where training is possible.

1. Introduction

Deep neural networks (DNN) have achieved state of the art results on numerous tasks. Hence, there is a multitude of works trying to theoretically explain their remarkable performance; see, e.g., (Du et al., 2018; Nguyen and Hein, 2018; Zhang et al., 2017; Zou et al., 2018). Recently, Jacot et al. (2018) introduced the Neural Tangent Kernel (NTK) that characterises DNN training in the so-called Lazy training regime (or NTK regime). In this regime, the whole training procedure is reduced to a first order Taylor expansion of the output function near its initialization value. It was shown in (Lee et al., 2019), that such a simple model could lead to surprisingly good performance. However, most experiments with NTK regime are performed on shallow neural networks and have not covered DNNs. In this paper, we cover this topic by showing the limitations of the NTK regime for DNNs and how it differs from the actual training of DNNs with Stochastic Gradient Descent.

Neural Tangent Kernel. Jacot et al. (2018) showed that training a neural network (NN) with GD (Gradient Descent) in parameter space is equivalent to a GD in a function space with respect to the NTK. Du et al. (2019) used a similar approach to prove that full batch GD converges to global minima for shallow neural networks, and Karakida et al. (2018) linked the Fisher information matrix to the NTK, studying its spectral distribution for infinite width NN. The infinite width limit for different architectures was studied by Yang (2019), who introduced a tensor formalism that can express the NN computations. Lee et al. (2019) studied a linear approximation of the full batch GD dynamics based on the NTK, and gave a method to approximate the NTK for different architectures. Finally, Arora et al. (2019) proposed an efficient algorithm to compute the NTK for convolutional architectures (Convolutional NTK). In all of these papers, the authors only studied the effect of the infinite width limit (NTK regime) with relatively shallow networks.

Information propagation. In parallel, information propagation in wide DNNs has been studied in (Hayou et al., 2019; Lee et al., 2018; Schoenholz et al., 2017; Yang and Schoenholz, 2017a). These works provide an analysis of the signal propagation at the initial step as a function of the initialization hyper-parameters (i.e. variances of the initial random weights and biases). They identify a set of hyper-parameters known as the Edge of Chaos (EOC) and activation functions ensuring a deep propagation of the information carried by the input. This ensures that the network output still has some information about the input. In this paper, we prove that the Edge of Chaos initialization has also some benefits on the NTK.

NTK training and SGD training. Stochastic Gradient Descent (SGD) has been successfully used in training deep networks. Recently, with the introduction of the Neural Tangent Kernel in (Jacot et al., 2018), Lee et al. (2019) suggested a different approach to training overparameterized neural networks. The idea originates from the conjecture that in overparameterized models, a local minima exists near initialization weights. Thus, using a first order Taylor expansion near initialization, the model is reduced to a simple linear model, and the linear model is trained instead.
of the original network. Hereafter, we refer to this training procedure as the NTK training and the trained model as the NTK regime. We clarify this in section 2.

Contributions. The aim of this paper is to study the large depth limit of NTK. Our contributions are

- We prove that the NTK regime is always trivial in the limit of large depth. However, the convergence rate to this trivial regime is controlled by the initialization hyper-parameters.
- We prove that only an EOC initialization provides a sub-exponential convergence rate to this trivial regime, while other initializations yield an exponential rate. For the same depth, the NTK regime is thus ‘less’ trivial for an EOC. This allows training deep models using NTK training.
- For ResNets, we also have convergence to a trivial NTK regime but this always occurs at a polynomial rate, irrespective of the initialization. To further slow down the NTK convergence rate, we introduce scaling factors to the ResNet blocks, which allows NTK training of deep ResNets.
- We leverage our theoretical results on the asymptotic behaviour of the NTK to show the existence of a learning rate passband for SGD training where training is possible.

Table 1 summarizes the behaviour of NTK and SGD training for different depths and initialization schemes of an FFNN on the MNIST dataset. We show if the model learns or not, i.e. if the model test accuracy is significantly bigger than 10%, which is the accuracy of the trivial random classifier. The results displayed in the table show that for shallow FFNN (L = 3), the model learns to classify with both NTK training and SGD training for any initialization scheme. For a medium depth network (L = 30), NTK training and SGD training both succeed in training the model with an initialization on the EOC, while they both fail with other initializations. It has been observed that with SGD, an EOC initialization in beneficial for the training of deep neural networks (Hayou et al., 2019; Schoenholz et al., 2017). Our results show that the EOC initialization is also beneficial for NTK training (Section 2). However, for a deeper network with L = 300, the NTK training fails for any initialization, while SGD training succeeds in training the model with EOC initialization. This confirms the limitations of the NTK training for DNNs. However, although the large depth NTK regime is trivial, we leverage this asymptotic analysis to infer a theoretical upper bound on the learning (section 4). We illustrate our theoretical results through extensive simulations. All the proofs are detailed in the appendix.

2. Neural Networks and Neural Tangent Kernel

2.1. Setup and notations

Consider a neural network model consisting of L layers of widths \((n_l)_{1 \leq l \leq L}\), \(n_0 = d\), and let \(\theta = (\theta^l)_{1 \leq l \leq L}\) be the flattened vector of weights and bias indexed by the layer’s index, and \(p\) be the dimension of \(\theta\). The output \(f\) of the neural network is given by some mapping \(s: \mathbb{R}^{n_L} \rightarrow \mathbb{R}^p\) of the last layer \(y^L(x); o\) being the dimension of the output (e.g. number of classes for a classification problem). For any input \(x \in \mathbb{R}^d\), we thus have \(f(x, \theta) = s(y^L(x)) \in \mathbb{R}^o\). We train the model, \(\theta\) changes with time \(t\), and we denote by \(\theta_t\) the value of \(\theta\) at time \(t\) and \(f_t(x) = f(x, \theta_t)\). Let \(D = (x_i, z_i)_{1 \leq i \leq N}\) be the data set, and let \(\mathcal{X} = (x_i)_{1 \leq i \leq N}, \mathcal{Z} = (z_j)_{1 \leq j \leq o}\) be the matrices of input and output respectively, with dimension \(d \times N\) and \(o \times N\). We assume that there is no colinearity in the input dataset \(\mathcal{X}\), i.e. there is no two inputs \(x, x' \in \mathcal{X}\) such that \(x' = \alpha x\) for some \(\alpha \in \mathbb{R}\). We also assume that there exists a compact set \(E \subset \mathbb{R}^d\) such that \(\mathcal{X} \subset E\).

The NTK \(K^L_\theta\) is defined as the \(o \times o\) dimensional kernel satisfying for all \(x, x' \in \mathbb{R}^d\)

\[
K^L_\theta(x, x') = \nabla_\theta f(x, \theta) \nabla_{\theta'} f(x', \theta)^T = \sum_{l=1}^{L} \nabla_{\theta^l} f(x, \theta_l) \nabla_{\theta^l} f(x', \theta_l)^T \in \mathbb{R}^{o \times o}.
\]

- The NTK regime (Infinite width): In the case of an FFNN, Jacot et al. (2018) proved that, with GD, the kernel \(K^L_\theta\) converges to \(K^L\), which depends only on \(L\) (depth) for all \(t < T\) when \(n_1, n_2, ..., n_L \to \infty\) sequentially, where \(T\) is an upper bound on the training time. The infinite width limit of the training dynamics with a quadratic loss is given by the linear model

\[
f_t(\mathcal{X}) = e^{-\frac{t}{T}} \hat{K}^L f_0(\mathcal{X}) + (I - e^{-\frac{t}{T}} \hat{K}^L)\mathcal{Z},
\]

where \(\hat{K}^L = K^L(\mathcal{X}, \mathcal{X})\). For any input \(x \in \mathbb{R}^d\), we have

\[
f_t(x) = f_0(x) + \gamma(x, \mathcal{X})(I - e^{-\frac{t}{T}} \hat{K}^L)(\mathcal{Z} - f_0(\mathcal{X})),
\]

where \(\gamma(x, \mathcal{X}) = K^L(x, \mathcal{X})(\hat{K}^L)^{-1}\). Hereafter, we refer
to \(f_t\) by the "NTK regime solution" or simply the "NTK regime" when there is no confusion. For other loss functions such as the cross-entropy loss, (Lee et al., 2019) used some approximation to obtain the NTK regime. These approximations are implemented in Python library (Novak et al., 2020).

- **Role of the NTK in NTK training:** As it has been observed in Du et al. (2019), the convergence speed of \(f_t\) to \(f_{\infty}\) (infinite training time) is given by the smallest eigenvalue of \(K^L\). If the NTK becomes singular in the large depth limit, then the NTK training fails.

- **Generalization in the NTK regime:** From equation (2), the term \(\gamma\) plays a crucial role in the generalization capacity of the linear model. More precisely, different works (Du et al., 2019; Arora et al., 2019) showed that the inverse NTK plays a crucial role in the generalization error of wide shallow NN. Cao and Gu (2019) proved that training a FeedForward NN of (fixed) depth \(L\) with SGD gives a generalization bound of the form \(O(L\sqrt{zT(K^L)^{-1}z/N})\) in the limit of infinite width, where \(z\) is the training label. Moreover, equation (2) shows that the Reproducing Kernel Hilbert Space (RKHS) generated by the NTK \(K^L\) controls the generalization function. To see this, let \(t \in (0, T)\), from equation (2), we can deduce that there exist coefficients \(a_1, \ldots, a_N \in \mathbb{R}\) such that for all \(x \in \mathbb{R}^d\), \(f_t(x) - f_0(x) = \sum_{i=1}^{N} a_i K^L(x, x)\), showing that the ‘training residual’ \(f_t - f_0\) belongs to the RKHS of the NTK. In other words, the NTK controls whether the network would learn anything beyond initialization with NTK training (linearized regime).

3. Asymptotic Neural Tangent Kernel

In this section, we study the behaviour of \(K^L\) as \(L\) goes to \(\infty\). We prove that the limiting \(K^L\) is trivial so that the NTK cannot explain the generalization power of DNNs. However, with EOC initialization, this convergence is slow, which makes it possible to use NTK training for medium depth neural networks \((L = 30)\). However, since the limiting NTK is trivial, NTK training necessarily fails for large depth neural networks.

3.1. NTK parameterization and the Edge of Chaos

Let \(\phi\) be the activation function. We consider the following architectures:

- **FeedForward Fully-Connected Neural Network (FFNN)** Consider an FFNN of depth \(L\), widths \((n_l)_{1 \leq l \leq L}\), weights \(w^l\) and bias \(b^l\). For some input \(x \in \mathbb{R}^d\), the forward propagation using the NTK parameterization is given by

\[
y^1_l(x) = \frac{\sigma_w}{\sqrt{d}} \sum_{j=1}^{d} w^l_{ij} x_j + \sigma_b b^l_i, \\
y^l_i(x) = \frac{\sigma_w}{\sqrt{n_{l-1}}} \sum_{j=1}^{n_{l-1}} w^l_{ij} \phi(y^{l-1}_j(x)) + \sigma_b b^l_i, \quad l \geq 2.
\]  

(3)

- **Convolutional Neural Network (CNN)** Consider a 1D convolutional neural network of depth \(L\), denoting by \([m : n]\) the set of integers \([m, m + 1, \ldots, n]\) for \(n \leq m\), the forward propagation is given by

\[
y^1_{i,\alpha}(x) = \frac{\sigma_w}{\sqrt{n_1}} \sum_{j=1}^{n_0} \sum_{\beta \in kerr_l} w^1_{i,j,\beta} x_{j,\alpha,\beta} + \sigma_b b^1_i, \\
y^l_{i,\alpha}(x) = \frac{\sigma_w}{\sqrt{n_l}} \sum_{j=1}^{n_{l-1}} \sum_{\beta \in kerr_l} w^l_{i,j,\beta} \phi(y^{l-1}_{j,\alpha,\beta}(x)) + \sigma_b b^l_i,
\]

(4)

where \(i \in [1 : n_1]\) is the channel number, \(\alpha \in [0 : M - 1]\) is the neuron location in the channel, \(n_l\) is the number of channels in the \(l\)th layer, and \(M\) is the number of neurons in each channel, \(kerr_l = [-k : k]\) is a filter with size \(2k + 1\) and \(\nu_l = n_{l-1}(2k + 1)\). Here, \(w^l \in \mathbb{R}^{n_l \times n_{l-1} \times (2k+1)}\).

We assume periodic boundary conditions, which result in having \(y^l_{i,\alpha} = y^l_{i,\alpha+M} = y^l_{i,\alpha-M}\), and similarly for \(l = 0\), \(x_{i,\alpha+M_\theta} = x_{i,\alpha} = x_{i,\alpha-M_\theta}\). For the sake of simplification, we consider the case of 1D CNN, the generalization to a 2D CNN for \(m \in N\) is straightforward.

Hereafter, for \(x, x' \in \mathbb{R}^d\), we denote by \(x \cdot x'\) the scalar product in \(\mathbb{R}^d\). For \(x, x' \in \mathbb{R}^{n_0 \times (2k+1)}\), let \([x, x']_{\alpha,\alpha'}\) be a convolutional mapping defined by \([x, x']_{\alpha,\alpha'} = \sum_{l=1}^{n_0} \sum_{\beta \in kerr_0} x_{j,\alpha,\beta} x_{j,\alpha',\beta}\).

We initialize the model randomly with \(w_{ij}^l \sim \mathcal{N}(0, 1)\), where \(\mathcal{N}(\mu, \sigma^2)\) denotes the normal distribution of mean \(\mu\) and variance \(\sigma^2\). In the limit of infinite width, the neurons \((y^l_{i,\alpha}(\cdot))_{i,j}\) become Gaussian processes (Neal, 1995; Lee et al., 2018; Matthews et al., 2018; Hayou et al., 2019; Schoenholz et al., 2017); hence, studying their covariance kernel is the natural way to gain insights on their behaviour. Hereafter, we denote by \(q^l(x, x')\) resp. \(q^l_{\alpha,\alpha'}(x, x')\) the covariance between \(y^l_i(x)\) and \(y^l_i(x')\) resp. \(y^l_{i,\alpha}(x)\) and \(y^l_{i,\alpha}(x')\). We define the correlations \(c^l(x, x')\) and \(c^l_{\alpha,\alpha'}(x, x')\) similarly.

For FFNN, we have that

\[
q^1(x, x') = \sigma_w^2 + \frac{\sigma_b^2}{d} x \cdot x',
\]

and similarly for CNN we have

\[
q^1_{\alpha,\alpha'}(x, x') = \sigma_w^2 + \frac{\sigma_b^2}{n_0(2k+1)} [x, x']_{\alpha,\alpha'}.
\]
For $\epsilon \in (0,1)$, we define the set $B_\epsilon$ by:

FFNN : $B_\epsilon = \{(x,x') \in \mathbb{R}^d : c^1(x,x') \leq 1 - \epsilon\}$,

CNN : $B_\epsilon = \{(x,x') \in \mathbb{R}^d : \forall \alpha, \alpha', c^1_{\alpha,\alpha'}(x,x') \leq 1 - \epsilon\}$,

and assume that there exists $\epsilon > 0$, such that for all $x \neq x' \in \mathcal{X}$, $(x,x') \in B_\epsilon$. The infinite width limit refers to infinite number of neurons for Fully Connected layers, and infinite number of channels for Convolutional layers. All results below are derived in this limit.

Jacot et al. (2018) established the following infinite width limit of the NTK of an FFNN when $\sigma_w = 1$. We generalize the result to any $\sigma_w > 0$.

**Lemma 1** (Generalization of Theorem 1 in (Jacot et al., 2018)). Consider an FFNN of the form (3). Then, as $n_1, n_2, ..., n_{l-1} \to \infty$, we have for all $x, x' \in \mathbb{R}^d$, $i, i' \leq n_l$, $K^l_{ii}(x,x') = \delta_{ii} K^l(x,x')$, where $K^l(x,x')$ is given by the recursive formula

$$K^l(x,x') = \hat{q}^l(x,x') K^{l-1}(x,x') + \hat{q}^l(x,x'),$$

where $\hat{q}^l(x,x') = \sigma_x^2 + \sigma_w^2 \mathbb{E}[\phi(y_1^{-1}(x))\phi(y_1^{-1}(x'))]$ and $\hat{q}^l(x,x') = \sigma_x^2 \mathbb{E}[\phi'(y_1^{-1}(x))\phi'(y_1^{-1}(x'))].$

Lemmas 1, 2, 3 and 4 are trivial and follow the same induction approach as in (Jacot et al., 2018). These results can be obtained using the Tensor Program framework of Yang (2020) for example.

**Lemma 2** (Infinite width dynamics of the NTK of a CNN). Consider a CNN of the form (4), then we have that for all $x, x' \in \mathbb{R}^d$, $i, i' \leq n_1$ and $\alpha, \alpha' \in [0 : M - 1]$

$$K^1_{(i,\alpha), (i',\alpha')}(x,x') = \delta_{ii'} \left( \frac{\sigma_x^2}{n_0(2k + 1)(x,x')_{\alpha,\alpha'} + \sigma_w^2} \right).$$

For $l \geq 2$, as $n_1, n_2, ..., n_{l-1} \to \infty$ recursively, we have for all $i, i' \leq n_1$, $\alpha, \alpha' \in [0 : M - 1]$, $K^l_{(i,\alpha), (i',\alpha')}(x,x') = \delta_{ii'} K^l_{\alpha,\alpha'}(x,x')$, where $K^l_{\alpha,\alpha'}$ is given by the recursion

$$K^l_{\alpha,\alpha'} = \frac{1}{2k + 1} \sum_{\beta \in \text{ker}l} \Psi^{-1}_{\alpha+\beta,\alpha'+\beta},$$

where $\Psi^{-1}_{\alpha+\beta,\alpha'+\beta} = \hat{q}^l_{\alpha+\beta,\alpha'+\beta} K^{l-1}_{\alpha+\beta,\alpha'+\beta} + \hat{q}^l_{\alpha,\alpha'}$, and $\hat{q}^l_{\alpha,\alpha'}$ resp. $\hat{q}^l_{\alpha,\alpha'}$ is defined as $q^l$, resp. $q^l$ in Lemma 1, with $y_1^{-1}(x)$, $y_1^{-1}(x')$ in place of $y_1^{-1}(x)$, $y_1^{-1}(x').$

The NTK of a CNN differs from that of an FFNN in the sense that it is an average over the NTK values of the previous layer. This is due to the fact that neurons in the same channel are not independent at initialization.

Using the above recursive formulas for the NTK, we can develop its mean-field theory to better understand its dynamics as $L$ goes to infinity. To alleviate notations, we hereafter use the notation $K^L$ for the NTK of both FFNN and CNN. For FFNN, it represents $K^L_1$ given by Lemma 1, whereas for CNN, it represents $K^L_{\alpha,\alpha'}$ given in lemma 2 for any $\alpha, \alpha'$, i.e. all results that follow are true for any $\alpha, \alpha'$. We start by reviewing the Edge of Chaos theory.

**Edge of Chaos (EOC):** For some input $x$, we denote by $q^l(x)$ the variance of $y^l(x)$. The convergence of $q^l(x)$ as $l$ increases is studied in (Lee et al., 2018), (Schoenholz et al., 2017), and (Hayou et al., 2019). Under general regularity conditions, it is proved that $q^l(x)$ converges to a point $q(\sigma_b, \sigma_w) > 0$ independent of $x$ as $l \to \infty$. The asymptotic behaviour of the correlation $c^l(x,x')$ between $y^l(x)$ and $y^l(x')$ for any two inputs $x$ and $x'$ is also driven by $(\sigma_b, \sigma_w)$: Schoenholz et al. (2017) show that if $\sigma_w^2 \mathbb{E}[\phi'(\sqrt{q(\sigma_b, \sigma_w)}Z)]^2 < 1$, where $Z \sim \mathcal{N}(0,1)$ then $c^l(x,x')$ converges to $1$ exponentially quickly, and the authors call this phase the ordered phase. However, if $\sigma_w^2 \mathbb{E}[\phi'(\sqrt{q(\sigma_b, \sigma_w)}Z)]^2 > 1$ then $c^l(x,x')$ converges to $c < 1$, which is then referred to as the chaotic phase. The authors define the EOC as the set of parameters $(\sigma_b, \sigma_w)$, such that $\sigma_w^2 \mathbb{E}[\phi'(\sqrt{q(\sigma_b, \sigma_w)}Z)]^2 = 1$. The behaviour of $c^l(x,x')$ on the EOC is studied in (Hayou et al., 2019) where it is proved to converge to $1$ at a polynomial rate (see Section 2 of the Supplementary). The exact rate depends on the smoothness of the activation function.

The following proposition establishes that any initialization on the Ordered or Chaotic phase, leads to a trivial limiting NTK as $L$ becomes large.

**Proposition 1** (NTK with Ordered/Chaotic Initialization). Let $(\sigma_b, \sigma_w)$ be either in the ordered or in the chaotic phase. Then, there exist $\lambda > 0$ such that for all $\epsilon \in (0,1)$, there exists $\gamma > 0$ such that

$$\sup_{(x,x') \in B_\epsilon} |K^L(x,x') - \lambda| \leq e^{-\gamma L}.$$

The proof of proposition 1 relies on the asymptotic analysis of the second moment of the gradient. We refer the reader to section 6 in the appendix for more details.

Proposition 1 shows that $K^L$ becomes close to a constant matrix as the depth grows. The exponential convergence rate implies that even with a small number of layers, the kernel $K^L$ is close to being degenerate. This suggests that NTK training fails, and the performance of the NTK regime solution will be no better than that of a random classifier. Empirically, we find that with depth $L = 30$, the NTK training fails when the network is initialized on the Ordered phase. See Section 5 for more details.

Before stating the results for EOC initialization, we introduce the following assumption on the input space of CNN.

**Assumption 1.** [CNN input space] We assume that for all $x, x' \in \mathcal{X}$, $q^l_{\alpha,\alpha'}(x,x')$ is independent of $\alpha, \alpha'$.
Assumption 1 is a constraint on the input space of CNN. It simplifies the analysis of the NTK of CNN by linking it to that of an FFNN. We refer the reader to Section 3 in the appendix for more details. We will specify it clearly whenever we use this assumption.

With an initialization on the EOC, the convergence rate is polynomial instead of exponential. We show this in the next theorem. Hereafter, we define the Average NTK (ANTK) by \( AK^L = K^L / L \). The notation \( g(x) = \Theta(m(x)) \) means there exist two constants \( A, B > 0 \) such that \( A m(x) \leq g(x) \leq B m(x) \).

**Theorem 1 (NTK on the Edge of Chaos).** Let \( \phi \) be a non-linear activation function, \((\sigma_b, \sigma_w) \in EOC \) and \( AK^L = K^L / L \). We have that

\[
\sup_{x \in E} |AK^L(x, x) - AK^\infty(x, x)| = \Theta(L^{-1}).
\]

Moreover, there exists a constant \( \lambda \in (0, 1) \) such that for all \( \epsilon \in (0, 1) \)

\[
\sup_{(x, x') \in B_{\epsilon}} |AK^L(x, x') - AK^\infty(x, x')| = \Theta(\log(L) L^{-1}),
\]

where

- if \( \phi = ReLU \), then \( AK^\infty(x, x') = \sigma_b^2 \frac{||x|| \cdot ||x'||}{d}(1 - (1 - \lambda) \mathbb{I}_{x \neq x'}) \) with \( \lambda = 1/4 \).
- if \( \phi = Tanh \), then \( AK^\infty(x, x') = q(1 - (1 - \lambda) \mathbb{I}_{x \neq x'}) \) where \( q > 0 \) is a constant and \( \lambda = 1/3 \).

All results hold for CNN under Assumption 1.

The proof of Theorem 1 is tricky and requires a special form of inequalities to control the convergence rate (i.e. to obtain \( \Theta \) instead of \( \mathcal{O} \)). We refer the reader to section 1 in the appendix for more details about the proof techniques.

Theorem 1 shows that with an initialization on the EOC, \( K^L \) increases linearly in \( L \). Moreover, the EOC initialization slows down significantly the convergence rate (w.r.t \( L \)) of \( AK^L \) to the trivial kernel \( AK^\infty \). This is of big importance since \( AK^\infty \) is trivial and brings hardly any information on \( x \). Indeed the convergence rate of \( AK^L \) to \( AK^\infty \) is \( \mathcal{O}(\log(L) L^{-1}) \). This means that as \( L \) grows, the NTK with EOC is still much further from the trivial kernel \( AK^\infty \) compared to the NTK with the Ordered/Chaotic initialization. This allows NTK training on deeper networks compared to the Ordered phase initialization. For ReLU, a similar result appeared independently in (Huang et al., 2020) after the first version of this paper was made publicly available. However, the authors only proved an upper bound on the convergence rate of order \( \mathcal{O}(\log(L) L^{-1}) \), while our result gives the exact rate of \( \Theta(\log(L) L^{-1}) \) for both ReLU and Tanh. We also extend the results to ResNet and a Scaled form of ResNet in the next section.

3.2. Residual Neural Networks (ResNet)

Another important feature of DNNs, which is known to be highly influential, is their architecture. For residual networks, the NTK has also a simple recursion in the infinite width limit.

**Lemma 3** (NTK of a ResNet with fully connected layers in the infinite width limit). Let \( K^{res,1} \) be the exact NTK for the ResNet with \( 1 \) layer. Then

- For the first layer (without residual connections), we have for all \( x, x' \in \mathbb{R}^d \)

\[
K^{res,1}_{ii'}(x, x') = \delta_{ii'} \left( \sigma_b^2 + \frac{\sigma_w^2}{d} x \cdot x' \right).
\]

- For \( l \geq 2 \), as \( n_1, n_2, ..., n_{l-1} \to \infty \) recursively, we have for all \( i, i' \in [1 : n_l], K^{res,l}_{ii'} = \delta_{ii} K^{res}_{ii'} \), where \( K^{res}_{ii'} \) is given by the recursive formula for all \( x, x' \in \mathbb{R}^d \)

\[
K^{res}_{ii'}(x, x') = K^{res,l-1}_{ii'}(x, x')(q'(x, x') + 1) + q'(x, x').
\]

For residual networks with convolutional layers, the formula is similar to the CNN case as well.

**Lemma 4** (NTK of a ResNet with convolutional layers in the infinite width limit). Let \( K^{res,1} \) be the exact NTK for the ResNet with \( 1 \) layer. Then

- For the first layer (without residual connections), we have for all \( x, x' \in \mathbb{R}^d \)

\[
K^{res,1}_{(i,\alpha),(i',\alpha')}(x, x') = \delta_{ii'} \left( \frac{\sigma_b^2}{n_0(2k + 1)} [x, x']_{\alpha,\alpha'} + \sigma_b^2 \right).
\]

- For \( l \geq 2 \), as \( n_1, n_2, ..., n_{l-1} \to \infty \) recursively, we have for all \( i, i' \in [1 : n_l], \alpha, \alpha' \in [0 : M - 1], K^{res,l}_{(i,\alpha),(i',\alpha')}(x, x') = \delta_{ii'} K^{res,l}_{(i',\alpha')}(x, x'), \) where \( K^{res,l}_{(i,\alpha),(i',\alpha')}(x, x') \) is given by the recursive formula for all \( x, x' \in \mathbb{R}^d \), using the same notations as in lemma 2.

\[
K^{res,l}_{(i,\alpha),(i',\alpha')}(x, x') = K^{res,l-1}_{(i,\alpha),(i',\alpha')}(x, x') + \frac{1}{2k + 1} \sum_{\beta} \Psi^l_{\alpha,\beta,\alpha',\beta}.
\]

where \( \Psi^l_{\alpha,\beta,\alpha',\beta} = q_{\alpha,\alpha'}^l K^{res,l-1}_{(i,\alpha),(i',\alpha')}(x, x') + q_{\alpha,\alpha'}^l \).

The additional terms \( K^{res,l-1}_{(i,\alpha),(i',\alpha')}(x, x') \) (resp. \( K^{res,l-1}_{(i,\alpha),(i',\alpha')}(x, x') \)) in the recursive formulas of Lemma 3 (resp. Lemma 4) are due to the ResNet architecture. It turns out that this term helps in slowing down the convergence rate of the NTK. The next proposition shows that for any \( \sigma_w > 0 \), the NTK of a ResNet explodes (exponentially) as \( L \) grows. However, a normalized version \( \tilde{K}^L = K^L / \sigma_L \) of the NTK of a ResNet will always have a polynomial convergence rate to a limiting trivial kernel.

**Theorem 2** (NTK for ResNet). Consider a ResNet satisfying

\[
y_l(x) = y^{l-1}_l(x) + F(w', y^{l-1}_l(x)), \quad l \geq 2,
\]
Assumption 1. All results hold for ResNet with Convolutional layers under 

Moreover, there exists a constant 

Theorem 2 shows that the NTK of a ReLU ResNet ex-

where \( F \) is either a convolutional or dense layer (equations (3) and (4)) with ReLU activation. Let \( K_{\text{res}}^L \) be the corre-

which only grows as \( L^{1+\sigma_w^2/2} \) instead of \( L(1+\sigma_w^2)^{L-1} \); 

Second, it drastically slows down the convergence rate to the 

for all \( \epsilon \in (0,1) \) such that for all \( \epsilon \in (0,1) \)

where \( \bar{K}_{\text{res}}^L(x,x') \) converges to a limiting kernel \( \bar{K}_{\text{res}}^\infty \) at the exact polynomial rate \( \Theta(\log(L)L^{-1}) \) for all \( \sigma_w > 0 \). This allows for NTK training of deep ResNet, similarly to the EOC initialization for the FFNN or the CNN networks. However, the NTK explodes exponentially and similarly to the EOC initialization for the FFNN or the CNN networks.

The term \( \alpha_L \) in the residual NTK might cause numerical stability issues for NTK training, and the triviality of the NTK regime lives in the RKHS of the NTK.

Proposition 2 (Scaled ResNet). Consider a ResNet satisfying

\[
y'(x) = y^{-1}(x) + \frac{1}{\sqrt{L}} F(w', y^{-1}(x)), \quad l \geq 2,
\]

where \( F \) is either a convolutional or dense layer (equations (3) and (4)) with ReLU activation. Then the results of Theorem 2 apply with \( \alpha_L = L^{1+\sigma_w^2/2} \) and the convergence rate \( \Theta(\log(L)L^{-1}) \).

Proposition 2 shows that scaling the residual blocks by \( 1/\sqrt{L} \) has two important effects on the NTK: first, it stabilizes the NTK which only grows as \( L^{1+\sigma_w^2/2} \) instead of \( L(1+\sigma_w^2)^{L-1} \); second, it drastically slows down the convergence rate to the limiting (trivial) \( \bar{K}_{\text{res}}^\infty \). Both properties are highly desirable for NTK training. The second property in particular means that with the scaling, we can ‘NTK train’ deeper ResNets compared to the non-scaled ResNets. We illustrate the effectiveness of Scaled ResNet in section 5. A more aggressive scaling was studied in (Huang et al., 2020), where authors scale the blocks with \( 1/L \) instead of our scaling \( 1/\sqrt{L} \), and show that it also stabilizes the NTK of ResNet. This is the main topic of the next chapter. We particularly show that a suitable scaling ensures that the limiting NTK is universal, i.e. we can approximate any continuous function on some compact set \( K \) with a function from the Reproducing Kernel Hilbert Space of the limiting NTK. This is a desirable property since the second term in the solution of the NTK regime lives in the RKHS of the NTK.

3.3. Spectral decomposition of the limiting NTK

To refine the analysis of Section 3, we study the limiting behaviour of the spectrum of the NTK over the unit sphere \( S^{d-1} = \{ x \in \mathbb{R}^d : ||x||_2 = 1 \} \). On the sphere \( S^{d-1} \), the kernel \( K^L \) is a dot-product kernel, i.e. there exists a function \( g_L \) such that \( K^L(x,x') = g_L(x \cdot x') \) for all \( x, x' \in S^{d-1} \). This kernel type is known to be diagonalizable on the sphere \( S^{d-1} \) and the eigenfunctions are the so-called Spherical Harmonics of \( S^{d-1} \). Many concurrent results have observed this fact (Geifman et al., 2020; Cao et al., 2020; Bietti and Mairal, 2019). Our goal in the next theorem is to confirm the results of the previous section from a spectral perspective, by showing that the eigenvalues of the NTK (scaled NTK) converge to zero as the depth goes to infinity, and only the first eigenvalue remains positive (which corresponds to the constant eigenfunction).

Theorem 3 (Spectral decomposition on \( S^{d-1} \)). Let \( \kappa^L \) be either, the NTK (\( K^L \)) for an FFNN with \( L \) layers initialized on the Ordered phase, The Average NTK (\( AK^L \)) for an FFNN with \( L \) layers initialized on the EOC, or the Normalized NTK (\( \bar{K}_{\text{res}}^L \)) for a ResNet with \( L \) layers (Fully Connected). Then, for all \( \lambda \geq 1 \), there exists \( (\mu_k^L)_{k \geq 1} \) such that for all \( x, x' \in S^{d-1} \)

\[
\kappa^L(x,x') = \sum_{k \geq 0} \mu_k^L \sum_{j=1}^{N(d,k)} Y_{k,j}(x)Y_{k,j}(x').
\]

(\( Y_{k,j} \)) are spherical harmonics of \( S^{d-1} \), and \( N(d,k) \) is the number of harmonics of order \( k \).

Moreover, we have that \( 0 < \mu_0^\infty = \lim_{L \to \infty} \mu_0^L < \infty \), and for all \( k \geq 1 \), \( \lim_{L \to \infty} \mu_k^L = 0 \).
Mean-field Behaviour of Neural Tangent Kernel for Deep Neural Networks

Figure 1. Train/Test accuracy of an FFNN with ReLU activation on Fashion MNIST dataset for different depths and learning rates, trained for 10 epochs. The plot is in log-log scale.

the spherical harmonics class. Therefore, in the limit of infinite depth, the RKHS of the kernel \( \kappa^L \) is reduced to the space of constant functions.

Although the asymptotic NTK is degenerate, we show in the next section that we can leverage the asymptotic analysis of section 3 to obtain valuable insights on the choice of the learning rate.

4. Learning Rate Passband

Tuning the learning rate (LR) is crucial for the training of DNNs; a large/small LR could cause the training to fail. Empirically, the optimal LR tends to decrease as the network depth grows. In this section, we use the NTK linear model presented in Section 2 to establish the existence of an LR passband, i.e. an interval of values for the learning rate where training occurs.

Recall the dynamics of the linear model

\[
df_t(\mathcal{X}) = -\frac{1}{N} \hat{K}^L (f_t(\mathcal{X}) - \mathcal{Z}) dt.
\]  

The GD update with learning rate \( \eta \) is given by

\[
f_{t+1}(\mathcal{X}) = (I - \frac{\eta}{N} \hat{K}^L) f_t(\mathcal{X}) - \frac{\eta}{N} \hat{K}^L \mathcal{Z}.
\]  

To ensure stability of (8), a necessary condition is that \( \|I - \frac{\eta}{N} \hat{K}^L\|_F < 1 \), which implies having

\[
\eta < \frac{2}{\mu_{\text{max}} (\frac{1}{N} \hat{K}^L)}
\]

where \( \mu_{\text{max}} \) is the largest eigenvalue.

For an FFNN (or a CNN with Assumption 1) initialized on the EOC, as \( L \) grows we have that \( \hat{K}^L = qL((1 - \lambda)I + \lambda U) + O(\log(L)) \) (Theorem 1). Therefore, for large \( L \) and \( N \), we have that \( \mu_{\text{max}} (\frac{1}{N} \hat{K}^L) \sim q\lambda L \). The upper bound on \( \eta \) scale as \( 1/L \), therefore, we expect the passband to have a linear upper bound. To validate this hypothesis, we train FFNN on Fashion MNIST dataset. Figure 1 shows the train/test accuracy for different LR and depths. The slope of the red line is \(-1\) which confirms our prediction that the upper bound of the LR passband grows as \( L^{-1} \). A similar bound has been introduced recently in (Hayase and Karakida, 2020) in the different context of networks achieving dynamical isometry with Hard-Tan activation function.

On the other hand, Figure 1 shows that lower bound of the passband of depths are almost uncorrelated.

5. Experiments

5.1. Behaviour of \( K^L \) as \( L \) goes to infinity

Proposition 1, and theorems 1 and 2 show that the NTK (or scaled NTK) converge to a trivial kernel. Figure 2 shows the normalized eigenvalues of the NTK of an FFNN on a 2D sphere. On the Ordered phase, the eigenvalues converge quickly to zero as the depth grows, while with an EOC initialization, the eigenvalues converge to zero at a slower rate. For \( L = 300 \), the NTK on the EOC is ‘richer’ than the NTK on the Ordered phase, in the sense that the small eigenvalues with EOC are relatively much bigger than those with the Ordered phase initialization. This reflects directly on the RKHS of the NTK, and allows the NTK regime solution to be ‘richer’ since it is a combination of different eigenfunctions of the NTK, and not only one as in the Ordered phase (the constant eigenfunction).

5.2. Can NTK regime explain DNN performance?

We train FFNN, Vanilla CNN (stacked convolutional layers without pooling, followed by a dense layer), Vanilla ResNet (ResNet with FFNN blocks), and Scaled ResNet with different depths using two training methods:

\[1\] We currently do not have an explanation for this effect. We leave this for future work.
SGD training. We use SGD with a batchsize of 128 and a learning rate $10^{-1}$ for $L \in \{3, 30\}$ and $10^{-2}$ for $L = 300$ (this learning rate was found by a grid search of exponential step size 10; note that the optimal learning rate with NTK parameterization is usually bigger than the optimal learning rate with standard parameterization). We use 100 training epochs for $L \in \{3, 30\}$, and 150 epochs for $L = 300$.

NTK training. We use the Python library Neural-Tangents introduced by Novak et al. (2020) with $10K$ samples from MNIST/CIFAR10. This corresponds to the inversion of a $10K \times 10K$ matrix to obtain the NTK regime solution discussed in Section 2.

For the EOC initialization, we use $(\sigma_b, \sigma_w) = (0, \sqrt{2})$ for ReLU, and $(\sigma_b, \sigma_w) = (0.2, 1.298)$ for Tanh. For the Ordered phase initialization, we use $(\sigma_b, \sigma_w) = (1, 0.1)$ for both ReLU and Tanh. Table 2 displays the test accuracies for both NTK training and SGD training. The dashed lines refer to the trivial test accuracy $\sim 10\%$, which is the test accuracy of a uniform random classifier with 10 classes i.e. in these cases the model does not learn. For $L = 300$, NTK training fails for all architectures and initializations confirming the results of Theorems 1 and 2, and Proposition 1; while SGD succeeds in training FFNN and CNN with an EOC initialization and fails with an Ordered initialization, and succeeds in training ResNet with both initializations (which confirms findings in (Yang and Schoenholz, 2017b) that ResNet ‘live’ on the EOC). This proves that the NTK regime cannot explain DNN performance trained with SGD. With $L = 300$, NTK training fails with Vanilla ResNet, while it yields good performance with scaled ResNet; this also confirms the benefits of the scaling introduced in Proposition 2. However, even with scaled ResNet, the NTK training fails for depth $L = 300$.

### Table 3. Test accuracy on CIFAR100 for ResNet.

| Epoch 10 | Epoch 160 |
|----------|-----------|
| ResNet32 | standard 54.18 ± 1.21 | 72.49 ± 0.18 |
|          | scaled    53.89 ± 2.32 | 74.07 ± 0.22 |
| ResNet50 | standard 51.09 ± 1.73 | 73.63 ± 1.51 |
|          | scaled    55.39 ± 1.52 | 75.02 ± 0.44 |
| ResNet104| standard 47.02 ± 2.23 | 74.77 ± 0.29 |
|          | scaled    56.38 ± 2.54 | 76.14 ± 0.98 |

### 6. Conclusion

In this paper, we have shown that the infinite depth limit of the NTK regime is trivial and cannot explain the performance of DNNs. However, we proved that the performance of NTK training is initialization dependent (Table 2). These
findings add to a recent line of research which shows that the infinite width approximation of the NTK does not fully capture the training dynamics of DNNs. Indeed, recent works have shown that the NTK for finite width neural networks changes with time (Chizat and Bach, 2018; Ghorbani et al., 2019; Huang and Yau, 2020), and might even be random as shown by (Hanin and Nica, 2019) where authors prove that in the limit $n, L \to \infty$ (where $n$ is a width of the network) with fixed ratio $\gamma = \frac{L}{n}$, the limiting kernel is random. An interesting property in this regime is the “feature learning” which the NTK regime lacks. Further research is needed in order to understand the difference between the two regimes.

Acknowledgement

The project leading to this work has received funding from the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No 834175).
References

Arora, S., S. Du, W. Hu, Z. Li, and R. Wand (2019). Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. ICML.

Arora, S., S. Du, W. Hu, Z. Li, R. Salakhutdinov, and R. Wang (2019). On exact computation with an infinitely wide neural net. arXiv preprint arXiv:1904.11955.

Bietti, A. and J. Mairal (2019). On the inductive bias of neural tangent kernels. NeurIPS 2019.

Cao, Y., Z. Fang, Y. Wu, D. Zhou, and Q. Gu (2020). Towards understanding the spectral bias of deep learning. arXiv preprint arXiv:1912.01198.

Cao, Y. and Q. Gu (2019). Generalization bounds of stochastic gradient descent for wide and deep neural networks. NeurIPS.

Chizat, L. and F. Bach (2018). A note on lazy training in supervised differentiable programming. arXiv preprint arXiv:1812.07956.

Du, S., J. Lee, H. Li, L. Wang, and X. Zhai (2019). Gradient descent finds global minima of deep neural networks. ICML.

Du, S., J. Lee, Y. Tian, B. Poczos, and A. Singh (2018). Gradient descent learns one-hidden-layer CNN: Don’t be afraid of spurious local minima. ICML.

Du, S., X. Zhai, B. Poczos, and A. Singh (2019). Gradient descent provably optimizes over-parameterized neural networks. ICLR.

Geifman, A., A. Yadav, Y. Kasten, M. Galun, D. Jacobs, and R. Basri (2020). On the similarity between the laplace and neural tangent kernels. NeurIPS.

Ghorbani, B., S. Mei, T. Misiakiewicz, and A. Montanari (2019). Linearized two-layers neural networks in high dimension. arXiv preprint arXiv:1904.12191.

Hanin, B. and M. Nica (2019). Finite depth and width corrections to the neural tangent kernel. arXiv preprint arXiv:1909.05989.

Hayase, T. and R. Karakida (2020). The spectrum of fisher information of deep networks achieving dynamical isometry. arXiv PrePrint 2006.07814.

Hayou, S., A. Doucet, and J. Rousseau (2019). On the impact of the activation function on deep neural networks training. ICML.

Huang, J. and H. Yau (2020). Dynamics of deep neural networks and neural tangent hierarchy. ICML.

Huang, K., Y. Wang, M. Tao, and T. Zhao (2020). Why do deep residual networks generalize better than deep feed-forward networks? – a neural tangent kernel perspective. ArXiv preprint, arXiv:2002.06262.

Jacot, A., F. Gabriel, and C. Hongler (2018). Neural tangent kernel: Convergence and generalization in neural networks. 32nd Conference on Neural Information Processing Systems.

Karakida, R., S. Akaho, and S. Amari (2018). Universal statistics of Fisher information in deep neural networks: Mean field approach. arXiv preprint arXiv:1806.01316.

Lee, J., Y. Bahri, R. Novak, S. Schoenholz, J. Pennington, and J. Sohl-Dickstein (2018). Deep neural networks as Gaussian processes. 6th International Conference on Learning Representations.

Lee, J., L. Xiao, S. Schoenholz, Y. Bahri, J. Sohl-Dickstein, and J. Pennington (2019). Wide neural networks of any depth evolve as linear models under gradient descent. NeurIPS.

Lillicrap, T., D. Cownden, D. Tweed, and C. Akerman (2016). Random synaptic feedback weights support error backpropagation for deep learning. Nature Communications 7(13276).

MacRobert, T. (1967). Spherical harmonics: An elementary treatise on harmonic functions, with applications. Pergamon Press.

Matthews, A., J. Hron, M. Rowland, R. Turner, and Z. Ghahramani (2018). Gaussian process behaviour in wide deep neural networks. 6th International Conference on Learning Representations.

Neal, R. (1995). Bayesian learning for neural networks. Springer Science & Business Media 118.

Nguyen, Q. and M. Hein (2018). Optimization landscape and expressivity of deep CNNs. ICML.

Novak, R., L. Xiao, J. Hron, J. Lee, A. A. Alemi, J. Sohl-Dickstein, and S. S. Schoenholz (2020). Neural tangents: Fast and easy infinite neural networks in python. In International Conference on Learning Representations.

Poole, B., S. Lahiri, M. Raghu, J. Sohl-Dickstein, and S. Ganguli (2016). Exponential expressivity in deep neural networks through transient chaos. 30th Conference on Neural Information Processing Systems.

Schoenholz, S., J. Gilmer, S. Ganguli, and J. Sohl-Dickstein (2017). Deep information propagation. 5th International Conference on Learning Representations.
Xiao, L., Y. Bahri, J. Sohl-Dickstein, S. S. Schoenholz, and P. Pennington (2018). Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks. ICML 2018.

Yang, G. (2019). Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. arXiv preprint arXiv:1902.04760.

Yang, G. (2020). Tensor programs iii: Neural matrix laws. arXiv preprint arXiv:2009.10685.

Yang, G. and S. Schoenholz (2017a). Mean field residual networks: On the edge of chaos. Advances in Neural Information Processing Systems 30, 2869–2869.

Yang, G. and S. Schoenholz (2017b). Mean field residual networks: On the edge of chaos. In Advances in neural information processing systems, pp. 7103–7114.

Zhang, C., S. Bengio, M. Hardt, B. Recht, and O. Vinyals (2017). Understanding deep learning requires rethinking generalization. arXiv preprint arXiv:1611.03530.

Zou, D., Y. Cao, D. Zhou, and Q. Gu (2018). Stochastic gradient descent optimizes over-parameterized deep ReLU networks. arXiv preprint arXiv:1811.08888.
Appendix

0. Setup and notations

0.1. Neural Tangent Kernel

Consider a neural network model consisting of \( L \) layers \((y^l)_{1 \leq l \leq L}\), with \( y^l : \mathbb{R}^{n_{l-1}} \rightarrow \mathbb{R}^{n_l} \), \( n_0 = d \) and let \( \theta = (\theta^l)_{1 \leq l \leq L} \) be the flattened vector of weights and bias indexed by the layer’s index and \( p \) be the dimension of \( \theta \). Recall that \( \theta^l \) has dimension \( n_l + 1 \). The output \( f \) of the neural network is given by some transformation \( S : \mathbb{R}^{n_L} \rightarrow \mathbb{o} \) of the last layer \( y^L(x) \); \( o \) being the dimension of the output (e.g. number of classes for a classification problem). For any input \( x \in \mathbb{R}^d \), we thus have \( f(x, \theta) = s(y^L(x)) \in \mathbb{R}^o \). As we train the model, \( \theta \) changes with time \( t \) and we denote by \( \hat{\theta}_t \) the value of \( \theta \) at time \( t \) and \( f_t(x) = f(x, \hat{\theta}_t) = (f_j(x, \hat{\theta}_t), j \leq o) \). Let \( D = (x_i, z_i)_{1 \leq i \leq N} \) be the data set and let \( X = (x_i)_{1 \leq i \leq N}, Z = (z_i)_{1 \leq i \leq N} \) be the matrices of input and output respectively, with dimension \( d \times N \) and \( o \times N \). For any function \( g : \mathbb{R}^{d \times o} \rightarrow \mathbb{R}^k \), \( k \geq 1 \), we denote by \( g(X, Z) \) the matrix \((g(x_i, z_i))_{1 \leq i \leq N} \) of dimension \( k \times N \).

(Jacot et al., 2018) studied the behaviour of the output of the neural network as a function of the training time \( t \) when the network is trained using a gradient descent algorithm. (Lee et al., 2019) built on this result to linearize the training dynamics. We recall hereafter some of these results.

For a given \( \theta \), the empirical loss is given by \( \mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^{N} \ell(f(x_i, \theta), z_i) \). The full batch GD algorithm is given by

\[
\hat{\theta}_{t+1} = \hat{\theta}_t - \eta \nabla_\theta \mathcal{L}(\hat{\theta}_t),
\]

where \( \eta > 0 \) is the learning rate.

Let \( T > 0 \) be the training time and \( N_c = T/\eta \) be the number of steps of the discrete GD (1). The continuous time system equivalent to (1) with step \( \Delta t = \eta \) is given by

\[
d\theta_t = -\nabla_\theta \mathcal{L}(\theta_t)dt.
\]

This differs from the result by (Lee et al., 2019) since we use a discretization step of \( \Delta t = \eta \). It is well known that this discretization scheme leads to an error of order \( O(\eta) \) (see Appendix). Equation (2) can be re-written as

\[
d\theta_t = -\frac{1}{N} \nabla_\theta f(X, \theta_t)^T \nabla_z \ell(f(X, \theta_t), Z)dt.
\]

where \( \nabla_\theta f(X, \theta_t) \) is a matrix of dimension \( oN \times p \) and \( \nabla_z \ell(f(X, \theta_t), Z) \) is the flattened vector of dimension \( oN \) constructed from the concatenation of the vectors \( \nabla_z \ell(z_i, z_i)|_{z_i = f(x_i, \theta_t)}, i \leq N \). As a result, the output function \( f_t(x) = f(x, \theta_t) \in \mathbb{R}^o \) satisfies the following ODE

\[
df_t(x) = -\frac{1}{N} \nabla_\theta f(x, \theta_t) \nabla_\theta f(x, \theta_t)^T \nabla_z \ell(f_t(X), Z)dt.
\]

The Neural Tangent Kernel (NTK) \( K^f_{\theta_t} \) is defined as the \( o \times o \) dimensional kernel satisfying: for all \( x, x' \in \mathbb{R}^d \),

\[
K^f_{\theta_t}(x, x') = \nabla_\theta f(x, \theta_t) \nabla_\theta f(x', \theta_t)^T \in \mathbb{R}^{o \times o}
\]

\[
= \sum_{l=1}^{L} \nabla_{\theta_l} f(x, \theta_t) \nabla_{\theta_l} f(x', \theta_t)^T.
\]

We also define \( K^f_{\theta_t}(X, X') \) as the \( oN \times oN \) matrix defined blockwise by

\[
K^f_{\theta_t}(X, X') = \begin{pmatrix}
K^f_{\theta_t}(x_1, x_1) & \cdots & K^f_{\theta_t}(x_1, x_N) \\
K^f_{\theta_t}(x_2, x_1) & \cdots & K^f_{\theta_t}(x_2, x_N) \\
\vdots & \ddots & \vdots \\
K^f_{\theta_t}(x_N, x_1) & \cdots & K^f_{\theta_t}(x_N, x_N)
\end{pmatrix}.
\]
We note hereafter \( \hat{w} \) weights. The infinite width limit of the training dynamics is given by 
\[
\gamma(t) = -\frac{1}{N}\sum_{y} K^L_y(x, x) \nabla_{\hat{w}} \ell(f(x), y) dt,
\]
meaning that for all \( j \leq N \)
\[
d_f(x_j) = -\frac{1}{N}\sum_{y} K^L_y(x_j, x) \nabla_{\hat{w}} \ell(f(x), y) dt.
\]

**Infinite width dynamics.** In the case of an FFNN, (Jacot et al., 2018) proved that, with GD, the kernel \( K^L_y \) converges to a kernel \( K^L \) which depends only on \( L \) neuron (number of layers) for all \( t < T \) when \( n_1, n_2, \ldots, n_L \to \infty \), where \( T \) is an upper bound on the training time, under the technical assumption \( \int_0^T \| \nabla_{\hat{w}} \ell(f(x), y) \|_2 dt < \infty \) a.s. with respect to the initialization weights. The infinite width limit of the training dynamics is given by 
\[
d_f(x) = -\frac{1}{N}\sum_{y} K^L_y(x, x) \nabla_{\hat{w}} \ell(f(x), y) dt.
\]

We note hereafter \( \hat{K}^L = K^L(x, x) \). As an example, with the quadratic loss \( \ell(z', z) = \frac{1}{2}||z' - z||^2 \), (6) is equivalent to 
\[
d_f(x) = -\frac{1}{N}\hat{K}^L \ell(f(x), y) dt,
\]
which is a simple linear model that has a closed-form solution given by 
\[
f(x) = e^{-\frac{1}{N}\hat{K}^L t} f_0(x) + (I - e^{-\frac{1}{N}\hat{K}^L t}) Z.
\]
For general input \( x \in \mathbb{R}^d \), we have 
\[
f(x) = f_0(x) + \gamma(x, x) (I - e^{-\frac{1}{N}\hat{K}^L t}) (Z - f_0(x)).
\]
where \( \gamma(x) = K^L(x, x)K^L(x, x)^{-1} \).  

### 0.2. Architectures

Let \( \phi \) be the activation function. We consider the following architectures (FFNN and CNN)

- **FeedForward Fully-Connected Neural Network (FFNN)** Consider an FFNN of depth \( L \), widths \( (n_t)_{1 \leq t \leq L} \), weights \( w^t \) and bias \( b^t \). For some input \( x \in \mathbb{R}^d \), the forward propagation using the NTK parameterization is given by 
\[
y^0_i(x) = \sum_{j=1}^d w^1_{ij} x_j + \sigma b^1_i,
\]
\[
y^l_i(x) = \sum_{j=1}^d w^l_{ij} \phi(y^{l-1}_j(x)) + \sigma b^l_i, \quad l \geq 2.
\]

- **Convolutional Neural Network (CNN/ConvNet)** Consider a 1D convolutional neural network of depth \( L \), denoting by \( [m : n] \) the set of integers \( \{m, m + 1, \ldots, n\} \) for \( n \leq m \), the forward propagation is given by 
\[
y_{i,\alpha}^1(x) = \sum_{j=1}^m w^1_{ij,\beta} x_{j,\alpha+\beta} + \sigma b^1_i,
\]
\[
y_{i,\alpha}^l(x) = \sum_{j=1}^m w^l_{ij,\beta} \phi(y^{l-1}_{j,\alpha+\beta}(x)) + \sigma b^l_i,
\]
where \( i \in [1 : n_t] \) is the channel number, \( \alpha \in [0 : M - 1] \) is the neuron location in the channel, \( n_t \) is the number of channels in the \( l^{th} \) layer, and \( M \) is the number of neurons in each channel, \( ker_1 = [-k : k] \) is a filter with size \( 2k + 1 \) and \( v_l = n_{l-1}(2k + 1) \). Here, \( w^l \in \mathbb{R}^{n_t \times n_{l-1} \times (2k + 1)} \). We assume periodic boundary conditions, which results in having \( y_{i,\alpha}^l = y_{i,\alpha+M}^l = y_{i,\alpha-M}^l \) and similarly for \( l = 0, x_{i,\alpha+M_0} = x_{i,\alpha} = x_{i,\alpha-M_0} \). For the sake of simplification, we consider only the case of 1D CNN, the generalization to a \( m \)D CNN for \( m \in \mathbb{N} \) is straightforward.
1. Proof techniques

The techniques used in the proofs range from simple algebraic manipulation to tricky inequalities.

Lemmas 1, 2, 3, 4. The proofs of these lemmas are simple and follow the same inductive argument as in the proof of the original NTK result in (Jacot et al., 2018). Note that these results can also be obtained by simple application of the Master Theorem in (Yang, 2020) using the framework of Tensor Programs.

Proposition 1, Theorems 1, 2. The proof of these results follow two steps: Firstly, estimating the asymptotic behaviour of the NTK in the limit of large depth; secondly, controlling these behaviour using upper/lower bounds. We analyse the asymptotic behaviour of the NTK of FFNN using existing results on signal propagation in deep FFNN. However, for CNNs, the dynamics are a bit trickier since they involve convolution operators; We use some results from the theory of Circulant Matrices for this purpose.

It is relatively easy to control the dynamics of the NTK in the Ordered/Chaotic phase, however, the dynamics become a bit complicated on the Edge of Chaos and technical lemmas which we call Appendix Lemmas are introduced for this purpose.

Theorem 3. The spectral decomposition of zonal kernels on the sphere is a classical result in spectral theory which was recently applied to Neural Tangent Kernel Geifman et al. (2020); Cao et al. (2020); Bietti and Mairal (2019). In order to prove the convergence of the eigenvalues, we use Dominated Convergence Theorem, leveraging the asymptotic results in Proposition 1 and Theorems 1, 2.

2. The infinite width limit

2.1. Forward propagation

FeedForward Neural Network. For some input $x \in \mathbb{R}^d$, the propagation of this input through the network is given by

$$y_1^l(x) = \frac{\sigma_w}{\sqrt{d}} \sum_{j=1}^{d} w_{ij}^l x_j + \sigma_b b_i^l$$

$$y_k^l(x) = \frac{\sigma_w}{\sqrt{n_{l-1}}} \sum_{j=1}^{n_{l-1}} w_{ij}^l \phi(y_{j}^{l-1}(x)) + \sigma_b b_i^l, \quad l \geq 2$$

Where $\phi : \mathbb{R} \to \mathbb{R}$ is the activation function. When we take the limit $n_{l-1} \to \infty$ recursively over $l$, this implies, using Central Limit Theorem, that $y_k^l(x)$ is a Gaussian variable for any input $x$. This gives an error of order $O(1/\sqrt{n_{l-1}})$ (standard Monte Carlo error). More generally, an approximation of the random process $y_k^l(.)$ by a Gaussian process was first proposed by (Neal, 1995) in the single layer case and has been extended to the multiple layer case by (Lee et al., 2018) and (Matthews et al., 2018). The limiting Gaussian process kernels follow a recursive formula given by, for any inputs $x, x' \in \mathbb{R}^d$

$$\kappa^l(x,x') = \mathbb{E}[y_k^l(x)y_k^l(x')]$$

$$= \sigma_b^2 + \sigma_w^2 \mathbb{E}[\phi(y_k^{l-1}(x))\phi(y_k^{l-1}(x'))]$$

$$= \sigma_b^2 + \sigma_w^2 \Psi_\phi(\kappa^{l-1}(x,x'), \kappa^{l-1}(x,x'), \kappa^{l-1}(x',x')),$$

where $\Psi_\phi$ is a function that only depends on $\phi$. This provides a simple recursive formula for the computation of the kernel $\kappa^l$; see, e.g., (Lee et al., 2018) for more details.

Convolutional Neural Networks. The infinite width approximation with 1D CNN yields a recursion for the kernel. However, the infinite width here means infinite number of channels, with a Monte Carlo error of $O(1/\sqrt{n_{l-1}})$. The kernel in this case depends on the choice of the neurons in the channel and is given by

$$\kappa_{\alpha,\alpha'}^{l}(x,x') = \mathbb{E}[y_{1,\alpha}^l(x)y_{1,\alpha'}^l(x')] = \sigma_b^2 + \frac{\sigma_w^2}{2k+1} \sum_{\beta \in \text{ker}} \mathbb{E}[\phi(y_{1,\alpha+\beta}^{l-1}(x))\phi(y_{1,\alpha'+\beta}^{l-1}(x'))]$$


so that 
\[ \kappa^l_{\alpha,\alpha'}(x, x') = \sigma_b^2 + \frac{\sigma_w^2}{2k + 1} \sum_{\beta \in \ker} F_{\phi}(\kappa^{l-1}_{\alpha+\beta,\alpha'+\beta}(x, x), \kappa^{l-1}_{\alpha+\beta,\alpha'+\beta}(x, x'), \kappa^{l-1}_{\alpha+\beta,\alpha'+\beta}(x', x')). \]

The convolutional kernel \( \kappa^l_{\alpha,\alpha'} \) has the ‘self-averaging’ property; i.e. it is an average over the kernels corresponding to different combination of neurons in the previous layer. However, it is easy to simplify the analysis in this case by studying the average kernel per channel defined by 
\[ \hat{\kappa}^l = \frac{1}{N^2} \sum_{\alpha,\alpha'} \kappa^l_{\alpha,\alpha'}. \]

Indeed, by summing terms in the previous equation and using the fact that we use circular padding, we obtain
\[
\hat{\kappa}^l(x, x') = \sigma_b^2 + \frac{\sigma_w^2}{N^2} \sum_{\alpha,\alpha'} F_{\phi}(\kappa^{l-1}_{\alpha,\alpha'}(x, x), \kappa^{l-1}_{\alpha,\alpha'}(x, x'), \kappa^{l-1}_{\alpha,\alpha'}(x', x')).
\]

This expression is similar in nature to that of FFNN. We will use this observation in the proofs.

Residual Neural Networks. The infinite width limit approximation for ResNet yields similar results with an additional residual terms. It is straightforward to see that, in the case of a ResNet with FFNN-type layers, we have that 
\[
\kappa^l(x, x') = \kappa^{l-1}(x, x') + \sigma_b^2 + \sigma_w^2 F_{\phi}(\kappa^{l-1}(x, x), \kappa^{l-1}(x, x'), \kappa^{l-1}(x', x')),
\]

whereas for ResNet with CNN-type layers, we have that 
\[
\kappa^l_{\alpha,\alpha'}(x, x') = \kappa^{l-1}_{\alpha,\alpha'}(x, x') + \sigma_b^2
+ \frac{\sigma_w^2}{2k + 1} \sum_{\beta \in \ker} F_{\phi}(\kappa^{l-1}_{\alpha+\beta,\alpha'+\beta}(x, x), \kappa^{l-1}_{\alpha+\beta,\alpha'+\beta}(x, x'), \kappa^{l-1}_{\alpha+\beta,\alpha'+\beta}(x', x')).
\]

2.2. Gradient Independence

In the mean-field literature of DNNs, an omnipresent approximation in prior literature is that of the gradient independence which is similar in nature to the practice of feedback alignment (Lillicrap et al., 2016). This approximation states that, for wide neural networks, the weights used for forward propagation are independent from those used for back-propagation. When used for the computation of Neural Tangent Kernel, this approximation was proven to give the exact computation for standard architectures such as FFNN, CNN and ResNets (Yang, 2020) (Theorem D.1).

This result has been extensively used in the literature as an approximation before being proved to yields exact computation for the NTK, and theoretical results derived under this approximation were verified empirically; see references below.

Gradient Covariance back-propagation. Analytical formulas for gradient covariance back-propagation were derived using this result, in (Hayou et al., 2019; Schoenholz et al., 2017; Yang and Schoenholz, 2017b; Lee et al., 2018; Poole et al., 2016; Xiao et al., 2018; Yang, 2019). Empirical results showed an excellent match for FFNN in (Schoenholz et al., 2017), for Resnets in (Yang, 2019) and for CNN in (Xiao et al., 2018).

Neural Tangent Kernel. The Gradient Independence approximation was implicitly used in (Jacot et al., 2018) to derive the infinite width Neural Tangent Kernel (See (Jacot et al., 2018), Appendix A.1). Authors have found that this infinite width NTK computed with the Gradient Independence approximation yields excellent match with empirical (exact) NTK.

We use this result in our proofs and we refer to it simply by the Gradient Independence.

3. Discussion on Assumption 1

Assumption 1. We assume that for all \( x, x' \in \mathcal{X} \), \( q_{\alpha,\alpha'}^1(x, x') \) is independent of \( \alpha, \alpha' \).
Assumption 1 implies that, there exists some function \( e : (x, x') \mapsto e(x, x') \) such that for all \( \alpha, \alpha', x, x' \)

\[
\sum_j \sum_{\beta \in \text{ker} \alpha} x_j, \alpha + \beta x_j', \alpha' + \beta = e(x, x')
\]

This system has \( N^2 M^2 \) equations and \( N \times 2n_0 \times M \) variables. Therefore, in the case \( n_0 >> 1 \), the set of solutions \( S \) is large. By using Assumption 1, we restrict our analysis to this case. Hereafter, for all CNN analysis, for some function \( G \) and set \( E \), taking the supremum \( \sup_{(x, x') \in E} G(x, x') \) should be interpreted as \( \sup_{(x, x') \in E \cap X^2} G(x, x') \).

Another justification to assumption 1 can be attributed a self-averaging property of the dynamics of the correlation inside a CNN. We refer the reader to the proof of Appendix lemma 3 for more details.

4. Warmup: Results from the Mean-Field theory of DNNs

4.1. Notation

For FFNN layers, let \( q^l(x) := q^l(x, x) \) be the variance of \( y^l_1(x) \) (the choice of the index 1 is not important since, in the infinite width limit, the random variables \( \{y^l_i(x)\}_{i \in [1:N]} \) are iid). Let \( q^l(x, x'), \) resp. \( c^l(x, x') \) be the covariance, resp. the correlation between \( y^l_1(x) \) and \( y^l_1(x') \). For Gradient back-propagation, let \( \hat{q}^l(x, x') \) be the Gradient covariance defined by

\[
\hat{q}^l(x, x') = \mathbb{E} \left[ \frac{\partial \mathcal{L}}{\partial y^l_1}(x) \frac{\partial \mathcal{L}}{\partial y^l_1}(x') \right]
\]

where \( \mathcal{L} \) is some loss function. Similarly, let \( \hat{q}^l(x) \) be the Gradient variance at point \( x \). We also define \( \tilde{q}^l(x, x') = \sigma_0^2 \mathbb{E}[\phi'(y^{l-1}_1(x))\phi'(y^{l-1}_1(x'))] \).

For CNN layers, we use similar notation across channels. Let \( q^l_{\alpha}(x) \) be the variance of \( y^l_{1, \alpha}(x) \) (the choice of the index 1 is not important here either since, in the limit of infinite number of channels, the random variables \( \{y^l_{1, \alpha}(x)\}_{\alpha \in [1:K]} \) are iid). Let \( q^l_{\alpha, \alpha'}(x, x') \) the covariance between \( y^l_{1, \alpha}(x) \) and \( y^l_{1, \alpha'}(x') \), and \( \tilde{q}^l_{\alpha, \alpha'}(x, x') \) the corresponding correlation. We also define the pseudo-covariance \( \tilde{q}^l_{\alpha, \alpha'}(x, x') = \sigma_0^2 + \sigma_w^2 \mathbb{E}[\phi(y^{l-1}_{1, \alpha}(x))\phi(y^{l-1}_{1, \alpha'}(x')]) \) and \( \tilde{q}^l_{\alpha, \alpha'}(x, x') = \sigma_0^2 \mathbb{E}[\phi(y^{l-1}_{1, \alpha}(x))\phi(y^{l-1}_{1, \alpha'}(x'))] \).

The Gradient covariance is defined by \( \hat{q}^l_{\alpha, \alpha'}(x, x') = \mathbb{E} \left[ \frac{\partial \mathcal{L}}{\partial y^l_{1, \alpha}}(x) \frac{\partial \mathcal{L}}{\partial y^l_{1, \alpha'}}(x') \right] \).

4.1.1. Covariance Propagation

**Covariance Propagation for FFNN.** In Section 2.1, we derived the covariance kernel propagation in an FFNN. For two inputs \( x, x' \in \mathbb{R}^d \), we have

\[
q^l(x, x') = \sigma_0^2 + \sigma_w^2 \mathbb{E}[\phi(y^{l-1}_1(x))\phi(y^{l-1}_1(x'))]
\]

(12)

this can be written as

\[
q^l(x, x') = \sigma_0^2 + \sigma_w^2 \mathbb{E} \left[ \phi \left( \sqrt{q^l(x)Z_1} \right) \phi \left( \sqrt{q^l(x') (c^{-1}Z_1 + \sqrt{1 - (c^{-1})^2}Z_2)} \right) \right], \quad Z_1, Z_2 \overset{iid}{\sim} \mathcal{N}(0, 1),
\]

with \( c^{-1} := c^{-1}(x, x') \).

With ReLU, and since ReLU is positively homogeneous (i.e. \( \phi(\lambda x) = \lambda \phi(x) \) for \( \lambda \geq 0 \)), we have that

\[
q^l(x, x') = \sigma_0^2 + \frac{\sigma_w^2}{2} \sqrt{q^l(x)} \sqrt{q^l(x')} f(c^{-1})
\]

where \( f \) is the ReLU correlation function given by \( \text{(Hayou et al., 2019)} \)

\[
f(c) = \frac{1}{\pi} \left( c \arcsin c + \sqrt{1 - c^2} \right) + \frac{1}{2} c.
\]

**Covariance Propagation for CNN.** The only difference with FFNN is that the independence is across channels and not neurons. Simple calculus yields

\[
q^l_{\alpha, \alpha'}(x, x') = \mathbb{E}[y^l_{1, \alpha}(x)y^l_{1, \alpha'}(x)'] = \sigma_0^2 + \frac{\sigma_w^2}{2k + 1} \sum_{\beta \in \text{ker} \alpha} \mathbb{E}[\phi(y^{l-1}_{1, \alpha+\beta}(x))\phi(y^{l-1}_{1, \alpha'}+\beta(x'))]
\]

Observe that

\[
q^l_{\alpha, \alpha'}(x, x') = \frac{1}{2k + 1} \sum_{\beta \in \text{ker} \alpha} \tilde{q}^l_{\alpha+\beta, \alpha'+\beta}(x, x')
\]

(13)
With ReLU, we have
\[ q^l_{\alpha,\alpha'}(x, x') = \sigma_b^2 + \frac{\sigma_w^2}{2k + 1} \sum_{\beta \in \text{ker}} \sqrt{q^l_{\alpha+\beta}(x)} \sqrt{q^l_{\alpha'+\beta}(x')} f(c^{l-1}_{\alpha+\beta,\alpha'+\beta}(x, x')). \]

**Covariance propagation for ResNet with ReLU.** In the case of ResNet, only an added residual term shows up in the recursive formula. For a ResNet with FFNN layers, the recursion reads
\[ q^l(x, x') = q^{l-1}(x, x') + \sigma_b^2 + \frac{\sigma_w^2}{2k + 1} \sum_{\beta \in \text{ker}} \sqrt{q^l_{\alpha+\beta}(x)} \sqrt{q^l_{\alpha'+\beta}(x')} f(c^{l-1}_{\alpha+\beta,\alpha'+\beta}(x, x')). \]

with CNN layers, we have instead
\[ q^l_{\alpha,\alpha'}(x, x') = q^{l-1}_{\alpha,\alpha'}(x, x') + \sigma_b^2 + \frac{\sigma_w^2}{2k + 1} \sum_{\beta \in \text{ker}} \sqrt{q^l_{\alpha+\beta}(x)} \sqrt{q^l_{\alpha'+\beta}(x')} f(c^{l-1}_{\alpha+\beta,\alpha'+\beta}(x, x')). \]

4.1.2. **Gradient Covariance Back-propagation**

**Gradient back-propagation for FFNN.** The gradient back-propagation is given by
\[ \frac{\partial L}{\partial y^l_i} = \phi'(y^l_i) \sum_{j=1}^{N_{l+1}} \frac{\partial L}{\partial y^l_{j+1}} W^l_{ji}. \]

where \( L \) is some loss function. Using the Gradient Independence 2.2, we have as in (Schoenholz et al., 2017)
\[ q^l(x) = q^{l+1}(x) \frac{N_{l+1}}{N_l} \chi(q^l(x)). \]

where \( \chi(q^l(x)) = \sigma_w^2 \mathbb{E}[\phi(\sqrt{q^l(x)}Z)^2] \).

**Gradient Covariance back-propagation for CNN.** We have that
\[ \frac{\partial L}{\partial W^l_{i,j,\beta}} = \sum_{\alpha} \frac{\partial L}{\partial y^l_{i,\alpha}} \phi(y^l_{j,\alpha+\beta}). \]

Moreover,
\[ \frac{\partial L}{\partial y^l_{i,\alpha}} = \sum_{j=1}^{n} \sum_{\beta \in \text{ker}} \frac{\partial L}{\partial y^l_{j,\alpha+\beta}} W^l_{i,j,\beta} \phi(y^l_{i,\alpha}). \]

Using the Gradient Independence 2.2, and taking the average over the number of channels we have that
\[ \mathbb{E} \left[ \frac{\partial L}{\partial y^l_{i,\alpha}} \right] = \sigma_w^2 \mathbb{E}[\phi(\sqrt{q^l_{\alpha}(x)}Z)]^{2} \sum_{\beta \in \text{ker}} \mathbb{E} \left[ \frac{\partial L}{\partial y^l_{i,\alpha+\beta}} \right]^{2}. \]

We can get similar recursion to that of the FFNN case by summing over \( \alpha \) and using the periodic boundary condition, this yields
\[ \sum_{\alpha} \mathbb{E} \left[ \frac{\partial L}{\partial y^l_{i,\alpha}} \right]^{2} = \chi(q^l_{\alpha}(x)) \sum_{\alpha} \mathbb{E} \left[ \frac{\partial L}{\partial y^l_{i,\alpha}} \right]^{2}. \]

4.1.3. **Edge of Chaos (EOC)**

Let \( x \in \mathbb{R}^d \) be an input. The convergence of \( q^l(x) \) as \( l \) increases has been studied by (Schoenholz et al., 2017) and (Hayou et al., 2019). In particular, under weak regularity conditions, it is proven that \( q^l(x) \) converges to a point \( q(\sigma_b, \sigma_w) > 0 \) independent of \( x \) as \( l \to \infty \). The asymptotic behaviour of the correlations \( c'(x, x') \) between \( y^l(x) \) and \( y^l(x') \) for any two
inputs \( x \) and \( x' \) is also driven by \((\sigma_b, \sigma_w)\): the dynamics of \( c^l \) is controlled by a function \( f \) i.e. \( c^{l+1} = f(c^l) \) called the correlation function. The authors define the EOC as the set of parameters \((\sigma_b, \sigma_w)\) such that \[
\sigma^2_b E[\phi'(\sqrt{q(\sigma_b, \sigma_w)}Z)]^2 = 1
\] where \( Z \sim \mathcal{N}(0, 1) \). Similarly the Ordered, resp. Chaotic, phase is defined by \[
\sigma^2_b E[\phi'(\sqrt{q(\sigma_b, \sigma_w)}Z)]^2 < 1, \text{ resp. }
\sigma^2_b E[\phi'(\sqrt{q(\sigma_b, \sigma_w)}Z)]^2 > 1.
\] On the Ordered phase, the gradient will vanish as it backpropagates through the network, and the correlation \( c^l(x, x') \) converges exponentially to 0. Hence the output function becomes constant (hence the name 'Ordered phase'). On the Chaotic phase, the gradient explodes and the correlation converges exponentially to some limiting value \( c < 1 \) which results in the output function being discontinuous everywhere (hence the 'Chaotic' phase name). On the EOC, the second moment of the gradient remains constant throughout the backpropagation and the correlation converges to 1 at a sub-exponential rate, which allows deeper information propagation. Hereafter, \( f \) will always refer to the correlation function.

We initialize the model with \( w_i^l, b_i^l \sim \mathcal{N}(0, 1) \), where \( \mathcal{N}(\mu, \sigma^2) \) denotes the normal distribution of mean \( \mu \) and variance \( \sigma^2 \). In the remainder of this appendix, we assume that the following conditions are satisfied

- The input data is a subset of a compact set \( E \) of \( \mathbb{R}^d \), and no two inputs are co-linear.
- All calculations are done in the limit of infinitely wide networks.

### 4.2. Some results from the information propagation theory

**Results for FFNN with Tanh activation.**

**Fact 1.** For any choice of \( \sigma_b, \sigma_w \in \mathbb{R}^+ \), there exist \( q, \lambda > 0 \) such that for all \( l \geq 1 \), \( \sup_{x \in \mathbb{R}^d} |q(x, x) - q| \leq e^{-\lambda l} \). (Equation (3) and conclusion right after in (Schoenholz et al., 2017)).

**Fact 2.** On the Ordered phase, there exists \( \gamma > 0 \) such that \( \sup_{x, x' \in \mathbb{R}^d} |c^l(x, x') - 1| \leq e^{-\gamma l} \). (Equation (8) in (Schoenholz et al., 2017))

**Fact 3.** Let \( (\sigma_b, \sigma_w) \in \text{EOC} \). Using the same notation as in fact 4, we have that \( \sup_{x, x' \in \mathbb{R}^d} |c^l(x, x')| = O(l^{-1}) \). (Proposition 3 in (Hayou et al., 2019)).

**Fact 4.** Let \( B_\epsilon = \{(x, x') \in \mathbb{R}^d : c^l(x, x') < 1 - \epsilon \} \). On the Chaotic phase, there exist \( c < 1 \) such that for all \( \epsilon \in (0, 1) \), there exists \( \gamma > 0 \) such that \( \sup_{x, x' \in B_\epsilon} |c^l(x, x') - c| \leq e^{-\gamma l} \). (Equations (8) and (9) in (Schoenholz et al., 2017))

**Fact 5** (Correlation function). The correlation function \( f \) is defined by \( f(x) = \frac{\sigma_b^2 + \sigma_w^2 E[\phi(\sqrt{qZ_1})\phi(\sqrt{q}(xZ_1 + \sqrt{1-x^2}Z_2))]}{q} \) where \( q \) is given in Fact 1 and \( Z_1, Z_2 \) are iid standard Gaussian variables.

**Fact 6.** \( f \) has a derivative of any order \( j \geq 1 \) given by
\[
\phi^{(j)}(x) = \sigma^2_b q^{-1} E[\phi^{(j)}(Z_1)\phi^{(j)}(xZ_1 + \sqrt{1-x^2}Z_2)], \quad \forall x \in [-1, 1]
\]
As a result, we have that \( \phi^{(j)}(1) = \sigma^2_b q^{-1} E[\phi^{(j)}(Z_1)^2] > 0 \) for all \( j \geq 1 \).

The proof of the previous fact is straightforward following the same integration by parts technique as in the proof of Lemma 1 in (Hayou et al., 2019). The result follows by induction.

**Fact 7.** Let \( (\sigma_b, \sigma_w) \in \text{EOC} \). We have that \( f'(1) = 1 \) (by definition of EOC). As a result, the Taylor expansion of \( f \) near 1 is given by
\[
f(c) = c + \alpha(1-c)^2 - \zeta(1-c)^3 + O((1-c)^4).
\]
where \( \alpha, \zeta > 0 \).

**Proof.** The proof is straightforward using fact 6, and integral-derivative interchanging.

### Results for FFNN with ReLU activation.

**Fact 8.** The ordered phase for ReLU is given by \( \text{Ord} = \{(\sigma_b, \sigma_w) \in (\mathbb{R}^+)^2 : \sigma_w < \sqrt{2}\} \). Moreover, for any \( (\sigma_b, \sigma_w) \in \text{Ord} \), there exist \( \lambda \) such that for all \( l \geq 1 \), \( \sup_{x \in \mathbb{R}^d} |q^l(x, x) - q| \leq e^{-\lambda l} \), where \( q = \frac{\sigma_b^2}{1-\sigma_w^2} \).
The proof is straightforward using equation (12).

**Fact 9.** For any \((\sigma_b, \sigma_w)\) in the Ordered phase, there exist \(\lambda\) such that for all \(l \geq 1\), \(\sup_{(x,x') \in \mathbb{R}^d} \left| c^l(x, x') - 1 \right| \leq e^{-\lambda l} \).

The proof of this claim follows from standard Banach Fixed point theorem in the same fashion as for Tanh in (Schoenholz et al., 2017).

**Fact 10.** The Chaotic phase for ReLU is given by \(Ch = \{(\sigma_b, \sigma_w) \in (\mathbb{R}^+)^2 : \sigma_w > \sqrt{2}\} \). Moreover, for any \((\sigma_b, \sigma_w) \in Ch\), for all \(l \geq 1\), \(x \in \mathbb{R}^d\), \(q^l(x, x) \gtrsim \left(\sigma_w^2 / 2\right)^l\).

The variance explodes exponentially on the Chaotic phase, which means the output of the Neural Network can grow arbitrarily in this setting. Hereafter, when no activation function is mentioned, and when we choose "\((\sigma_b, \sigma_w)\) on the Ordered/Chaotic phase", it should be interpreted as "\((\sigma_b, \sigma_w)\) on the Ordered phase" for ReLU and "\((\sigma_b, \sigma_w)\) on the Ordered/Chaotic phase" for Tanh.

**Fact 11.** For ReLU FFNN on the EOC, we have that \(q^l(x, x) = \frac{\sigma_w^2}{2} \|x\|^2\) for all \(l \geq 1\).

The proof is straightforward using equation 12 and that \((\sigma_b, \sigma_w) = (0, \sqrt{2})\) on the EOC.

**Fact 12.** The EOC of ReLU is given by the singleton \((\sigma_b, \sigma_w) = (0, \sqrt{2})\}. In this case, the correlation function of an FFNN with ReLU is given by

\[
f(x) = \frac{1}{\pi} (x \arcsin x + \sqrt{1 - x^2}) + \frac{1}{2} x\]

(Proof of Proposition 1 in (Hayou et al., 2019)).

**Fact 13.** Let \((\sigma_b, \sigma_w) \in EOC\). Using the same notation as in fact 4, we have that

\[
\sup_{(x,x') \in B_x} |1 - c^l(x, x')| = O(t^{-2})
\]

(Follows straightforwardly from Proposition 1 in (Hayou et al., 2019)).

**Fact 14.** We have that

\[
f(c) = c + s(1 - c)^{3/2} + b(1 - c)^{5/2} + O((1 - c)^{7/2})
\]

with \(s = \frac{2\sqrt{2}}{3\pi}\) and \(b = \frac{\sqrt{2}}{30\pi}\).

This result was proven in (Hayou et al., 2019) (in the proof of Proposition 1) for order 3/2, the only difference is that here we push the expansion to order 5/2.

**Results for CNN with Tanh activation function.**

**Fact 15.** For any choice of \(\sigma_b, \sigma_w \in \mathbb{R}^+\), there exist \(q, \lambda > 0\) such that for all \(l \geq 1\), \(\sup_{x, x' \in \mathbb{R}^d} |q^l(\sigma_{\alpha, \alpha'}, x, x') - q| \leq e^{-\lambda l}\). (Equation (2.5) in (Xiao et al., 2018) and variance convergence result in (Schoenholz et al., 2017)).

The behaviour of the correlation \(c^l_{\alpha, \alpha'}(x, x')\) was studied in (Xiao et al., 2018) only in the case \(x' = x\). We give a comprehensive analysis of the asymptotic behaviour of \(c^l_{\alpha, \alpha'}(x, x')\) in the next section.

**General results on the correlation function.**

**Fact 16.** Let \(f\) be either the correlation function of Tanh or ReLU. We have that

- \(f(1) = 1\) (Lemma 2 in (Hayou et al., 2019)).
- On the ordered phase \(0 < f'(1) < 1\) (By definition).
- On the Chaotic phase \(f'(1) > 1\) (By definition).
- On the EOC, \(f'(1) = 1\) (By definition).
- On the Ordered phase and the EOC, \(1\) is the unique fixed point of \(f\) ((Hayou et al., 2019)).
• On the Chaotic phase, $f$ has two fixed points, 1 which is unstable, and $c < 1$ which is a stable fixed point (Schoenholz et al., 2017).

**Fact 17.** Let $\epsilon \in (0, 1)$. On the Ordered/Chaotic phase, with either ReLU or Tanh, there exists $\alpha \in (0, 1), \gamma > 0$ such that

$$\sup_{(x, x') \in B_\epsilon} |f'(c(x, x')) - \alpha| \leq e^{-\gamma l}$$

**Proof.** This result follows from a simple first order expansion inequality. For Tanh on the Ordered phase, we have that

$$\sup_{(x, x') \in B_\epsilon} |f'(c(x, x')) - f'(1)| \leq \zeta \sup_{(x, x') \in B_\epsilon} |c(x, x') - 1|$$

where $\zeta = \sup_{t \in (\min_{c, x' \in B_\epsilon} c(x, x'), 1)} |f''(t)| \rightarrow |f''(1)|$. We conclude for Ordered phase with Tanh using fact 2. The same argument can be used for Chaotic phase with Tanh using fact 4; in this case, $\alpha = f'(c)$ where $c$ is the unique stable fixed point of the correlation function $f$.

On the Ordered phase with ReLU, let $\tilde{f}$ be the correlation function. It is easy to see that $\tilde{f}'(c) = \frac{\pi^2}{2} f'(c)$ where $f$ is given in fact 12. $f'(x) = 1 - \frac{\pi}{2}(1 - x)^{1/2} + O((1 - x)^{3/2})$. Therefore, there exists $l_0, \zeta > 0$ such that for $l > l_0$,

$$\sup_{(x, x') \in B_\epsilon} |f'(c(x, x')) - f'(1)| \leq \zeta \sup_{(x, x') \in B_\epsilon} |c(x, x') - 1|^{1/2}$$

We conclude using fact 9. \qed

**Asymptotic behaviour of the correlation in FFNN.**

**Appendix Lemma 1** (Asymptotic behaviour of $c^l$ for ReLU). Let $(\sigma_b, \sigma_w) \in EOC$ and $\epsilon \in (0, 1)$. We have

$$\sup_{(x, x') \in B_\epsilon} \left| c^l(x, x') - 1 + \frac{\kappa}{l^2} - 3\sqrt{\frac{\log(l)}{l^3}} \right| = O(l^{-3})$$

where $\kappa = \frac{9\pi^2}{2}$. Moreover, we have that

$$\sup_{(x, x') \in B_\epsilon} \left| f'(c^l(x, x')) - 1 + \frac{\kappa}{l} - \frac{9}{2\sqrt{\kappa}} \frac{\log(l)}{l^2} \right| = O(l^{-2})$$

**Proof.** Let $(x, x') \in B_\epsilon$ and $s = \frac{3\pi}{\sqrt{2}}$. From the preliminary results, we have that $\lim_{l \rightarrow \infty} \sup_{x, x' \in \mathbb{R}^d} 1 - c^l(x, x') = 0$ (fact 13). Using fact 14, we have uniformly over $B_\epsilon$,

$$\gamma_{l+1} = \gamma_l - s\gamma_l^{3/2} - b\gamma_l^{5/2} + O(\gamma_l^{7/2})$$

where $s, b > 0$, this yields

$$\gamma_{l+1}^{-1/2} = \gamma_l^{-1/2} + \frac{s}{2} + \frac{3s^2}{8}\gamma_l^{1/2} + \frac{b}{2}\gamma_l + O(\gamma_l^{3/2}).$$

Thus, as $l$ goes to infinity

$$\gamma_{l+1}^{-1/2} - \gamma_l^{-1/2} \sim \frac{s}{2},$$

and by summing and equivalence of positive divergent series

$$\gamma_l^{-1/2} \sim \frac{s}{2} l.$$

Moreover, since $\gamma_l^{-1/2} = \gamma_l^{-1/2} + \frac{s}{2} + \frac{3s^2}{8}\gamma_l^{1/2} + O(\gamma_l^{3/2})$, using the same argument multiple times and inverting the formula yields

$$c^l(x, x') = 1 - \frac{\kappa}{l^2} + 3\sqrt{\frac{\log(l)}{l^3}} + O(l^{-3})$$
We prove a similar result for an FFNN with Tanh activation.

**Appendix Lemma 2** (Asymptotic behaviour of $c^j$ for Tanh). Let $(\sigma_b, \sigma_w) \in EOC$ and $\epsilon \in (0, 1)$. We have

$$\sup_{(x,x') \in B_\epsilon} \left| c^j(x, x') - 1 + \frac{\kappa}{l} (1 - \kappa^2 \zeta) \frac{\log(l)}{l^3} \right| = O(l^{-3})$$

where $\kappa = \frac{2}{f''(1)} > 0$ and $\zeta = \frac{f''(1)}{6} > 0$. Moreover, we have that

$$\sup_{(x,x') \in B_\epsilon} \left| f'(c^j(x, x')) - 1 + \frac{2}{l} - 2(1 - \kappa^2 \zeta) \frac{\log(l)}{l^2} \right| = O(l^{-2}).$$

**Proof.** Let $(x, x') \in B_\epsilon$ and $\lambda_l := 1 - c^j(x, x')$. Using a Taylor expansion of $f$ near 1 (fact 7), there exist $\alpha, \zeta > 0$ such that

$$\lambda_{l+1} = \lambda_{l} - \alpha \lambda_{l}^2 + \zeta \lambda_{l}^3 + O(\lambda_{l}^4)$$

Here also, we use the same technique as in the previous lemma. We have that

$$\lambda_{l+1}^{-1} = \lambda_{l}^{-1}(1 - \alpha \lambda_{l} + \zeta \lambda_{l}^2 + O(\lambda_{l}^3))^{-1} = \lambda_{l}^{-1} (1 + \alpha \lambda_{l} + (\alpha^2 - \zeta) \lambda_{l}^2 + O(\lambda_{l}^3))$$

$$= \lambda_{l}^{-1} + \alpha + (\alpha^2 - \zeta) \lambda_{l} + O(\lambda_{l}^2).$$

By summing (divergent series), we have that $\lambda_{l}^{-1} \sim \alpha l$. Therefore,

$$\lambda_{l+1}^{-1} - \lambda_{l}^{-1} - \alpha = (\alpha^2 - \beta) \alpha^{-1} l^{-1} + o(l^{-1}).$$

By summing a second time, we obtain

$$\lambda_{l}^{-1} = \alpha l + (\alpha - \beta \alpha^{-1}) \log(l) + o(\log(l)).$$

Using the same technique once again, we obtain

$$\lambda_{l}^{-1} = \alpha l + (\alpha - \beta \alpha^{-1}) \log(l) + O(1).$$

This yields

$$\lambda_{l} = \alpha^{-1} l^{-1} - \alpha^{-1} (1 - \alpha^{-2} \beta) \frac{\log(l)}{l^2} + O(l^{-2}).$$

In a similar fashion to the previous proof, we can force the upper bound in $O$ to be independent of $x$ using Appendix Lemma 5. This way, the bound depends only on $\epsilon$. This concludes the first part of the proof.

For the second part, observe that $f''(x) = 1 + (x - 1) f'''(1) + O((x - 1)^2)$, hence

$$f'(c^j(x, x')) = 1 - \frac{2}{l} + 2(1 - \alpha^{-2} \zeta) \frac{\log(l)}{l^2} + O(l^{-2})$$

which concludes the proof.
4.3. Large depth behaviour of the correlation in CNNs

For CNNs, the infinite width will always mean the limit of infinite number of channels. Recall that, by definition, $q_{\alpha,\alpha'}(x, x') = \sigma_b^2 + \sigma_w^2 E[\phi(y^{-1}_{1,\alpha}(x))\phi(y^{-1}_{1,\alpha}(x'))]$ and $q_{\alpha,\alpha'}(x, x') = E[y^l_{1,\alpha}(x)y^l_{1,\alpha}(x')]$.

Unlike FFNN, neurons in the same channel are correlated since they share the same filters. Let $x, x'$ be two inputs and $\alpha, \alpha'$ two nodes in the same channel $i$. Using Central Limit Theorem in the limit of large $n_l$ (number of channels), we have

$$q_{\alpha,\alpha'}(x, x') = \mathbb{E}[y^l_{1,\alpha}(x)y^l_{1,\alpha}(x')] = \frac{\sigma_w^2}{2k + 1} \sum_{\beta \in \text{ker}} \mathbb{E}[\phi(y^{-1}_{1,\alpha+\beta}(x))\phi(y^{-1}_{1,\alpha+\beta}(x'))] + \sigma_b^2$$

Let $c_{\alpha,\alpha'}^l(x, x')$ be the corresponding correlation. Since $q_{\alpha,\alpha'}(x, x)$ converges exponentially to $q$ which depends neither on $x$ nor on $\alpha$, the mean-field correlation as in (Schoenholz et al., 2017; Hayou et al., 2019) is given by

$$c_{\alpha,\alpha'}^l(x, x') = \frac{1}{2k + 1} \sum_{\beta \in \text{ker}} f(c_{\alpha+\beta,\alpha'+\beta}^{l-1}(x, x'))$$

where $f(c) = \frac{\sigma_w^2}{q}E[\phi(\sqrt{\beta}Z_1)\phi(\sqrt{\beta}(Z_1 + \sqrt{1-\beta^2}Z_2))] + \sigma_b^2$ and $Z_1, Z_2$ are independent standard normal variables. The dynamics of $c_{\alpha,\alpha'}^l$ become similar to those of $c^l$ in an FFNN under assumption 1. We show this in the proof of Appendix Lemma 3. In (Xiao et al., 2018), authors studied only the limiting behaviour of correlations $c_{\alpha,\alpha'}^l(x, x)$ (same input $x$), however, they do not study $c_{\alpha,\alpha'}^l(x, x')$ when $x \neq x'$. We do this in the following Lemma, which will prove also useful for the main results of the paper.

**Appendix Lemma 3** (Asymptotic behaviour of the correlation in CNN with Tanh). We consider a CNN with Tanh activation function. Let $(\sigma_b, \sigma_w) \in (\mathbb{R}^+)^2$ and $\epsilon \in (0, 1)$. Let $B_\epsilon = \{(x, x') \in \mathbb{R}^d : \sup_{\alpha,\alpha'} c_{\alpha,\alpha'}^l(x, x') < 1 - \epsilon\}$. The following statements hold:

1. If $(\sigma_b, \sigma_w)$ are on the Ordered phase, then there exists $\beta > 0$ such that

$$\sup_{(x, x') \in \mathbb{R}^d} \sup_{\alpha,\alpha'} |c_{\alpha,\alpha'}^l(x, x') - 1| = O(e^{-\beta \epsilon})$$

2. If $(\sigma_b, \sigma_w)$ are on the Chaotic phase, then for all $\epsilon > 0$ there exists $\beta > 0$ and $\epsilon \in (0, 1)$ such that

$$\sup_{(x, x') \in \mathbb{R}^d} \sup_{\alpha,\alpha'} |c_{\alpha,\alpha'}^l(x, x') - 1| = O(e^{-\beta \epsilon})$$

3. Under Assumption 1, if $(\sigma_b, \sigma_w) \in \text{EOC}$, then we have

$$\sup_{(x, x') \in \mathbb{R}^d} \sup_{\alpha,\alpha'} |c_{\alpha,\alpha'}^l(x, x') - 1 + \frac{\kappa}{7} - \kappa(1 - \kappa^2 \zeta) \frac{\log(l)}{l^3}| = O(l^{-3})$$

where $\kappa = \frac{2}{\rho^2(1)} > 0$, $\zeta = \frac{f^2(1)}{6} > 0$, and $f$ is the correlation function given in Fact 5.

We prove statements 1 and 2 for general inputs, i.e. without using Assumption 1. The third statement requires Assumption 1.

**Proof.** Let $(x, x') \in \mathbb{R}^d$. Without using assumption 1, we have that

$$c_{\alpha,\alpha'}^l(x, x') = \frac{1}{2k + 1} \sum_{\beta \in \text{ker}} f(c_{\alpha+\beta,\alpha'+\beta}^{l-1}(x, x'))$$

Writing this in matrix form yields

$$C_l = \frac{1}{2k + 1} UF(C_{l-1})$$
Where $C_l = (c^l_{a,a+\beta}(x,x'))_{a\in\{0,N-1\},\beta\in\{0:N-1\}}$ is a vector in $\mathbb{R}^{N^2}$, $U$ is a convolution matrix and $f$ is applied elementwise. As an example, for $k=1$, $U$ is given by

$$U = \begin{bmatrix}
1 & 1 & 0 & \ldots & 0 & 1 \\
1 & 1 & 0 & \ddots & 0 \\
0 & 1 & 1 & \ddots & 0 \\
0 & 0 & 1 & \ddots & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots \\
1 & 0 & \ldots & 0 & 1 & 1
\end{bmatrix}$$

For general $k$, $U$ is a Circulant symmetric matrix with eigenvalues $\lambda_1 > \lambda_2 \geq \lambda_3 \ldots \geq \lambda_N$. The largest eigenvalue of $U$ is given by $\lambda_1 = 2k + 1$ and its equivalent eigenspace is generated by the vector $e_1 = \frac{1}{\sqrt{N}}(1,1,\ldots,1) \in \mathbb{R}^{N^2}$. This yields

$$(1 + 2k)^{-l}U^l = e_1e_1^T + O(e^{-\beta l})$$

where $\beta = \log(\frac{\lambda_1}{\lambda_N})$.

This provides another justification to Assumption 1; as $l$ grows, and assuming that $C_l \rightarrow e_1$ (which we show in the remainder of this proof), $C_l$ exhibits a self-averaging property since $C_l \approx \frac{1}{2\pi l+1}UC_{l-1}$. This system concentrates around the average value of the entries of $C_l$ as $l$ grows. Since the variances converge to a constant $q$ as $l$ goes to infinity (fact 15), this approximation implies that the entries of $C_l$ become almost equal as $l$ goes to infinity, thus making assumption 1 almost satisfied in deep layers. Let us now prove the statements.

1. Let $(\sigma_b, \sigma_u)$ be in the Ordered phase, $(x, x') \in \mathbb{R}^d$ and $c^l_{a,a'} = \min_{\alpha, \alpha'} c^l_{\alpha,\alpha'}(x,x')$. Using the fact that $f$ is non-decreasing, we have that $c^l_{\alpha,\alpha'}(x,x') \geq \frac{1}{2\pi l+1} \sum_{\beta \in \ker} c^{l-1}_{\alpha+\beta,\alpha'+\beta}(x,x') \geq c^l_{m}$. Taking the minimum again over $\alpha, \alpha'$, we have $c^l_{m} \geq c^l_{m-1}$. Therefore, $c^l_{m}$ is non-decreasing and converges to the unique fixed point of $f$ which is $c = 1$. Moreover, the convergence rate is exponential using the fact that (fact 16) $0 < f'(1) < 1$. To see this, observe that

$$\sup_{\alpha,\alpha'} |1 - c^l_{\alpha,\alpha'}(x,x')| \leq \left( \sup_{\zeta \in [c^{-1},1]} f'(|\zeta|) \times \sup_{\alpha,\alpha'} |1 - c^l_{\alpha,\alpha'}(x,x')| \right)$$

Knowing that $\sup_{\zeta \in [c^{-1},1]} f'(|\zeta|) \rightarrow f'(1) < 1$, we conclude. Moreover, the convergence is uniform in $(x,x')$ since the convergence rate depends only on $f'(1)$.

2. Let $\epsilon \in (0,1)$. In the Chaotic phase, the only difference is the limit $c = c_1 < 1$ and the Supremum is taken over $B_1$ to avoid points where $c^l(x,x') = 1$. In the Chaotic phase (fact 16), $f$ has two fixed points, 1 is an unstable fixed point and $c_1 \in (0,1)$ which is the unique stable fixed point. We conclude by following the same argument.

3. Let $\epsilon \in (0,1)$ and $(\sigma_b, \sigma_u) \in$ EOC. Using the same argument of monotony as in the previous cases and that $f$ has 1 as unique fixed point, we have that $\lim_{l \rightarrow \infty} \sup_{x,x'} \sup_{\alpha,\alpha'} |1 - c^l_{\alpha,\alpha'}(x,x')| = 0$.

From fact 7, the Taylor expansion of $f$ near 1 is given by

$$f(c) = c + \alpha(1-c)^2 - \zeta(1-c)^3 + O((1-c)^4).$$

where $\alpha = \frac{f''(1)}{2}$ and $\zeta = \frac{f^{(3)}(1)}{6}$. Using fact 6, we know that $f^{(k)}(1) = \sigma_u^2 q^{k-1} E[\phi^{(k)}(\sqrt{q}Z)^2]$. Therefore, we have $\alpha > 0$, and $\zeta < 0$.

Under assumption 1, it is straightforward that for all $\alpha, \alpha'$, and $l \geq 1$

$$c^l_{\alpha,\alpha'}(x,x') = c^l(x,x')$$

i.e. $c^l_{\alpha,\alpha'}$ are equal for all $\alpha, \alpha'$. The dynamics of $c^l(x,x')$ are exactly the dynamics of the correlation in an FFNN. We conclude using Appendix Lemma 2.
It is straightforward that the previous Appendix Lemma extend to ReLU activation, with slightly different dynamics. In this case, we use Appendix Lemma 1 to conclude for the third statement.

**Appendix Lemma 4** (Asymptotic behaviour of the correlation in CNN with ReLU-like activation functions). We consider a CNN with ReLU activation. Let \((\sigma_b, \sigma_w) \in (\mathbb{R}^+)^2\). Let \((\sigma_b, \sigma_w) \in (\mathbb{R}^+)^2\) and \(\epsilon \in (0, 1)\). The following statements hold

1. If \((\sigma_b, \sigma_w)\) are on the Ordered phase, then there exists \(\beta > 0\) such that
\[
\sup_{(x,x') \in \mathbb{R}^d} |c^l_{\alpha,\alpha'}(x, x') - 1| = O(e^{-\beta l})
\]

2. If \((\sigma_b, \sigma_w)\) are on the Chaotic phase, then there exists \(\beta > 0\) and \(c \in (0, 1)\) such that
\[
\sup_{(x,x') \in B_r} |c^l_{\alpha,\alpha'}(x, x') - c| = O(e^{-\beta l})
\]

3. Under Assumption 1, if \((\sigma_b, \sigma_w) \in EOC\), then
\[
\sup_{(x,x') \in B_r} \sup_{(x,x') \in B_r} \left| c^l_{\alpha,\alpha'}(x, x') - 1 + \frac{\kappa}{l^2} - 3 \sqrt{\kappa} \frac{\log(l)}{l^3} \right| = O(l^{-3})
\]

where \(\kappa = \frac{9\pi^2}{2}\).

**Proof.** The proof is similar to the case of Tanh in Appendix Lemma 3. The only difference is that we use Appendix Lemma 1 to conclude for the third statement. 

\[\square\]

### 5. A technical tool for the derivation of uniform bounds

Results in Theorem 1 and 2 and Proposition 1 involve a supremum over the set \(B_r\). To obtain such results, we need a ‘uniform’ Taylor analysis of the correlation \(c^l(x, x')\) (see the next section) where uniformity is over \((x, x') \in B_r\). It turns out that such result is trivial when the correlation follows a dynamical system that is controlled by a non-decreasing function. We clarify this in the next lemma.

**Appendix Lemma 5** (Uniform Bounds). Let \(A \subset \mathbb{R}\) be a compact set and \(g\) a non-decreasing function on \(A\). Define the sequence \(\zeta_l\) by \(\zeta_l = g(\zeta_{l-1})\) and \(\zeta_0 \in A\). Assume that there exist \(\alpha_l, \beta_l\) that do not depend on \(\zeta_l\), with \(\beta_l = o(\alpha_l)\), such that for all \(\zeta_0 \in A\),
\[
\zeta_l = \alpha_l + O_{\zeta_0}(\beta_l)
\]
where \(O_{\zeta_0}\) means that the \(O\) bound depends on \(\zeta_0\). Then, we have that
\[
\sup_{\zeta_0 \in A} |\zeta_l - \alpha_l| = O(\beta_l)
\]

i.e. we can choose the bound \(O\) to be independent of \(\zeta_0\).

**Proof.** Let \(\zeta_{0,m} = \min A\) and \(\zeta_{0,M} = \max A\). Let \((\zeta_{m,l})\) and \((\zeta_{M,l})\) be the corresponding sequences. Since \(g\) is non-decreasing, we have that for all \(\zeta_0 \in A\), \(\zeta_{m,l} \leq \zeta_l \leq \zeta_{M,l}\). Moreover, by assumption, there exists \(M_1, M_2 > 0\) such that
\[
|\zeta_{m,l} - \alpha_l| \leq M_1 |\beta_l|
\]
and
\[
|\zeta_{M,l} - \alpha_l| \leq M_2 |\beta_l|
\]

therefore,
\[
|\zeta_l - \alpha_l| \leq \max(|\zeta_{m,l} - \alpha_l|, |\zeta_{M,l} - \alpha_l|) \leq \max(M_1, M_2) |\beta_l|
\]

which concludes the proof. 

\[\square\]

Note that Appendix Lemma 5 can be easily extended to Taylor expansions with ‘\(o\)’ instead of ‘\(O\)’. We will use this result in the proofs, by refereeing to Appendix Lemma 5.
6. Proofs of Section 3: Large Depth Behaviour of Neural Tangent Kernel

6.1. Proofs of the results of Section 3.1

In this section, we provide proofs for the results of Section 3.1 in the paper.

Recall that Lemma 1 in the paper is a generalization of Theorem 1 in (Jacot et al., 2018) and is reminded here. The proof is simple and follows similar induction techniques as in (Jacot et al., 2018).

**Lemma 1** (Generalization of Th. 1 in (Jacot et al., 2018)). Consider an FFNN of the form (3). Then, as \( n_1, n_2, ..., n_{L-1} \to \infty \), we have for all \( x, x' \in \mathbb{R}^d \), \( i, i' \leq n_L \), \( K^L_i(x, x') = \delta_{ii'}K^L(x, x') \), where \( K^L(x, x') \) is given by the recursive formula

\[
K^L(x, x') = q^L(x, x')K^{L-1}(x, x') + q^L(x, x'),
\]

where \( q^L(x, x') = \sigma_b^2 + \sigma_w^2E[\phi(y_{i-1}^{-1}(x))\phi(y_{i-1}^{-1}(x'))] \) and \( q^L(x, x') = \sigma_w^2E[\phi'(y_{i-1}^{-1}(x))\phi'(y_{i-1}^{-1}(x'))]. \)

**Proof.** The proof for general \( \sigma_w \) is similar to when \( \sigma_w = 1 \) (Jacot et al., 2018) which is a proof by induction.

For \( l \geq 2 \) and \( i \in [1 : n_l] \)

\[
\partial_{\theta_{i,j}}y_{i+1}^{l+1}(x) = \frac{\sigma_w}{\sqrt{n_l}} \sum_{j=1}^{n_l} w_{ij}^{l+1} \phi'(y_j^l(x))\partial_{\theta_{i,j}}y_j^l(x).
\]

Therefore,

\[
(\partial_{\theta_{i,j}}y_{i}^{l+1}(x))(\partial_{\theta_{i,j}}y_{i}^{l+1}(x))' = \sigma_w^2 \sum_{j,j'}{w_{ij}^{l+1}w_{ij'}^{l+1}\phi'(y_j^l(x))\phi'(y_{j'}^l(x'))}\partial_{\theta_{i,j}}y_j^l(x)\partial_{\theta_{i,j}}y_{j'}^l(x').
\]

Using the induction hypothesis, namely that as \( n_0, n_1, ..., n_{l-1} \to \infty \), for all \( j, j' \leq n_l \) and all \( x, x' \)

\[
\partial_{\theta_{i,j}}y_{j'}^l(x)(\partial_{\theta_{i,j}}y_{j'}^l(x'))' \to K^l(x, x')1_{j = j'},
\]

we then obtain for all \( n_l \), as \( n_0, n_1, ..., n_{l-1} \to \infty \)

\[
\sigma_w^2 \sum_{j,j'}{w_{ij}^{l+1}w_{ij'}^{l+1}\phi'(y_j^l(x))\phi'(y_{j'}^l(x'))}\partial_{\theta_{i,j}}y_j^l(x)(\partial_{\theta_{i,j}}y_{j'}^l(x'))' \to \sigma_w^2 \sum_{j}{(w_{ij}^{l+1})^2\phi'(y_j^l(x))\phi'(y_{j'}^l(x'))K^l(x, x')}
\]

and letting \( n_l \) go to infinity, the law of large numbers, implies that

\[
\sigma_w^2 \sum_{j}{(w_{ij}^{l+1})^2\phi'(y_j^l(x))\phi'(y_{j'}^l(x'))K^l(x, x') \to q^{l+1}(x, x')K^l(x, x')}
\]

Moreover, we have that

\[
(\partial_{u_{i+1}^l,y_{i}^{l+1}(x)}(\partial_{u_{i+1}^l,y_{i}^{l+1}(x')}'))' + (\partial_{u_{i+1}^l,y_{i}^{l+1}(x)}(\partial_{u_{i+1}^l,y_{i}^{l+1}(x')}'))' = \frac{\sigma_w^2}{n_l} \sum_{j}{\phi(y_j^l(x))\phi(y_{j'}^l(x')) + \sigma_b^2}
\]

\[
\to_{n_l \to \infty} \sigma_w^2E[\phi(y_j^l(x))\phi(y_{j'}^l(x'))] + \sigma_b^2 = q^{l+1}(x, x').
\]

which ends the proof.

We now provide the recursive formula satisfied by the NTK of a CNN, namely Lemma 2 of the paper.
Lemma 2 (Infinite width dynamics of the NTK of a CNN). Consider a CNN of the form (4), then we have that for all \( x, x' \in \mathbb{R}^d, i, i' \leq n_1 \) and \( \alpha, \alpha' \in [0 : M - 1] \)

\[
K_{(i,\alpha),(i',\alpha')}^1(x, x') = \delta_{i'i'} \left( \frac{\sigma_w^2}{n_0(2k + 1)} [x, x']_{\alpha,\alpha'} + \sigma_b^2 \right)
\]

For \( l \geq 2 \), as \( n_1, n_2, \ldots, n_{l-1} \to \infty \) recursively, we have for all \( i, i' \leq n_l, \alpha, \alpha' \in [0 : M - 1] \), \( K_{(i,\alpha),(i',\alpha')}^l(x, x') = \delta_{i'i'} K_{\alpha,\alpha'}^l(x, x') \), where \( K_{\alpha,\alpha'}^l \) is given by the recursive formula

\[
K_{\alpha,\alpha'}^l = \frac{1}{2k + 1} \sum_{\beta \in \text{ker}_l} \Psi_{\alpha,\alpha'}^{l-1}
\]

where \( \Psi_{\alpha,\alpha'}^{l-1} = \hat{q}^l_{\alpha,\alpha'} K_{\alpha,\alpha'}^{l-1} + \hat{q}_1^{l-1,\alpha} \), and \( \hat{q}^l_{\alpha,\alpha'}, \hat{q}_1^{l-1,\alpha} \) are defined in Lemma 1, with \( y_{1,\alpha}^{l-1}(x), y_{1,\alpha}^{l-1}(x') \) in place of \( y_{l-1}^{1}(x), y_{l-1}^{1}(x') \).

Proof. Let \( x, x' \) be two inputs. We have that

\[
y_{1,\alpha}^l(x) = \frac{\sigma_w}{\sqrt{v_l}} \sum_{j=1}^{n_0} \sum_{\beta \in \text{ker}_l} w_{i,j,\beta}^1 x_{j,\alpha + \beta} + \sigma_b h_i^l
\]

\[
y_{1,\alpha}^l(x) = \frac{\sigma_w}{\sqrt{v_l}} \sum_{j=1}^{n_0} \sum_{\beta \in \text{ker}_l} w_{i,j,\beta}^1 \phi(y_{j,\alpha + \beta}^{l-1}(x)) + \sigma_b h_i^l
\]

therefore

\[
K_{(i,\alpha),(i',\alpha')}^1(x, x') = \sum_r \left( \sum_j \sum_{\beta \in \text{ker}_l} \frac{\partial y_{1,\alpha}^r(x)}{\partial w_{r,j,\beta}^1} \frac{\partial y_{1,\alpha'}^r(x')}{\partial w_{r,j,\beta}^1} \right) + \frac{\partial y_{1,\alpha}^1(x)}{\partial h_i^1} \frac{\partial y_{1,\alpha'}^1(x)}{\partial h_i^1}
\]

\[
= \delta_{i'i'} \left( \frac{\sigma_w^2}{n_0(2k + 1)} \sum_{\beta \in \text{ker}_l} x_{j,\alpha + \beta} x_{j,\alpha' + \beta} + \sigma_b^2 \right)
\]

Assume the result is true for \( l - 1 \), let us prove it for \( l \). Let \( \theta_{1:l-1} \) be model weights and bias in the layers 1 to \( l - 1 \). Let \( \frac{\partial \theta_{1:l-1} y_{1,\alpha}^l(x)}{\partial \theta_{1,l-1}} \). We have that

\[
\frac{\partial \theta_{1:l-1} y_{1,\alpha}^l(x)}{\partial \theta_{1,l-1}} = \frac{\sigma_w}{\sqrt{n_{l-1}(2k + 1)}} \sum_{\beta} w_{i,j,\beta}^l \phi'(y_{j,\alpha + \beta}^{l-1}) \frac{\partial \theta_{1,l-1} y_{1,\alpha + \beta}^{l-1}(x)}{\partial \theta_{1,l-1} y_{1,\alpha + \beta}^{l-1}(x)}
\]

this yields

\[
\frac{\partial \theta_{1:l-1} y_{1,\alpha}^l(x)}{\partial \theta_{1,l-1}} \frac{\partial \theta_{1:l-1} y_{1,\alpha'}^l(x)}{\partial \theta_{1,l-1}} = \frac{\sigma_w^2}{n_{l-1}(2k + 1)} \sum_{\beta} w_{i,j,\beta}^l w_{i,j,\beta'}^l \phi'(y_{j,\alpha + \beta}^{l-1}) \phi'(y_{j',\alpha' + \beta}^{l-1}) \frac{\partial \theta_{1,l-1} y_{1,\alpha + \beta}^{l-1}(x)}{\partial \theta_{1,l-1} y_{1,\alpha + \beta}^{l-1}(x)} \frac{\partial \theta_{1,l-1} y_{1,\alpha' + \beta}^{l-1}(x)}{\partial \theta_{1,l-1} y_{1,\alpha' + \beta}^{l-1}(x)}
\]

as \( n_1, n_2, \ldots, n_{l-2} \to \infty \) and using the induction hypothesis, we have

\[
\frac{\partial \theta_{1:l-1} y_{1,\alpha}^l(x)}{\partial \theta_{1,l-1}} \frac{\partial \theta_{1:l-1} y_{1,\alpha'}^l(x)}{\partial \theta_{1,l-1}} \to \frac{\sigma_w^2}{n_{l-1}(2k + 1)} \sum_{\beta, \beta'} w_{i,j,\beta}^l w_{i,j,\beta'}^l \phi'(y_{j,\alpha + \beta}^{l-1}) \phi'(y_{j',\alpha' + \beta}^{l-1}) K_{(j,\alpha + \beta),(j',\alpha' + \beta)}^{l-1}(x, x')
\]
We conclude using the fact that $K^{l-1}_{(j,{\alpha+\beta}),(j,\alpha'+\beta)}(x,x') = K^{l-1}_{(1,\alpha+\beta),(1,\alpha'+\beta)}(x,x')$ for all $j$ since the variables are iid across the channel index $j$. Now letting $n_{l-1} \to \infty$, we have that

$$\frac{1}{(2k+1)} \sum_{\beta,\beta'} q^l_{\alpha+\beta,\alpha'+\beta} K^{l-1}_{(1,\alpha+\beta),(1,\alpha'+\beta)}(x,x')$$

We conclude using the fact that

$$\partial_{\theta_{l+1}} y^l_{i,\alpha}(x) \partial_{\theta_{l+1}} y^l_{i',\alpha'}(x)^T \to \delta_{ii'} \left( -\frac{\sigma^2_w}{2k+1} \sum_{\beta} E[y^l_{\alpha+\beta}(x)] \phi(y^l_{\alpha+\beta}(x')) + \sigma^2_b \right)$$

To alleviate notations, we use hereafter the notation $K^L$ for both the NTK of FFNN and CNN. For FFNN, it represents the recursive kernel $K^L$ given by lemma 1, whereas for CNN, it represents the recursive kernel $K^L_{\alpha,\alpha'}$ for any $\alpha, \alpha'$, which means all results that follow are true for any $\alpha, \alpha'$. The following proposition establishes that any initialization on the Ordered or Chaotic phase, leads to a trivial limiting NTK as the number of layers $L$ becomes large.

**Proposition 1** (Limiting Neural Tangent Kernel with Ordered/Chaotic Initialization). Let $(\sigma_b, \sigma_w)$ be either in the ordered or in the chaotic phase. Then, there exist $\lambda > 0$ such that for all $\epsilon \in (0, 1)$, there exists $\gamma > 0$ such that

$$\sup_{(x,x') \in B_\epsilon} |K^L(x,x') - \lambda| \leq e^{-\gamma L}.$$ 

We will use the next lemma in the proof of proposition 1.

**Appendix Lemma 6.** Let $(a_l)$ be a sequence of non-negative real numbers such that $\forall l \geq 0, a_{l+1} \leq a_l + ke^{-\beta l}$, where $\alpha \in (0, 1)$ and $k, \beta > 0$. Then there exists $\gamma > 0$ such that $\forall l \geq 0$, $a_l \leq e^{-\gamma l}$.

**Proof.** Using the inequality on $a_l$, we can easily see that

$$a_l \leq a_0 \alpha^l + k \sum_{j=0}^{l-1} \alpha^j e^{-\beta(l-j)} \leq a_0 \alpha^l + \frac{k}{2} e^{-\beta l/2} + \frac{k}{2} \alpha^{l/2}$$

where we divided the sum into two parts separated by index $l/2$ and upper-bounded each part. The existence of $\gamma$ is straightforward.

Now we prove Proposition 1.

**Proof.** We prove the result for FFNN first. Let $x, x'$ be two inputs. From lemma 1, we have that

$$K^l(x,x') = K^{l-1}(x,x') q^l(x,x') + q^l(x,x')$$

where $q^l(x,x') = \sigma^2_b + \sigma^2_w x^T x'$ and $q^l(x,x') = \sigma^2_b + \sigma^2_w E_{f \sim N(0,q^{-1})}[\phi(f(x))\phi(f(x'))]$ and $q^l(x,x') = \sigma^2_a E_{f \sim N(0,q^{-1})}[\phi'(f(x))\phi'(f(x'))]$. From facts 1, 2, 4, 9, 17, in the ordered/chaotic phase, there exist $k, \beta, \eta, l_0 > 0$ and $\alpha \in (0, 1)$ such that for all $l \geq l_0$ we have

$$\sup_{(x,x') \in B_\epsilon} |q^l(x,x') - k| \leq e^{-\beta l}.$$
and

\[ \sup_{(x,x') \in B} |q^l(x,x') - \alpha| \leq e^{-\eta l}. \]

Therefore, there exists \( M > 0 \) such that for any \( l \geq l_0 \) and \( x, x' \in \mathbb{R}^d \)

\[ K^l(x, x') \leq M. \]

Letting \( r_l = \sup_{(x,x') \in B} |K^l(x, x') - \frac{k}{1+\alpha}| \), we have

\[ r_l \leq \alpha r_{l-1} + Me^{-\eta l} + e^{-\beta l}. \]

We conclude using Appendix Lemma 6.

Under Assumption 1, the proof is similar for CNN, using Appendix Lemmas 3 and 4.

Now, we show that the Initialization on the EOC improves the convergence rate of the NTK wrt \( L \). We first prove two preliminary lemmas that will be useful for the proof of the next proposition. Hereafter, the notation \( g(x) = \Theta(m(x)) \) means there exist two constants \( A, B > 0 \) such that \( Am(x) \leq g(x) \leq Bm(x) \).

**Appendix Lemma 7.** Let \( A, B, \Lambda \subset \mathbb{R}^+ \) be three compact sets, and \( (a_l), (b_l), (\lambda_l) \) be three sequences of non-negative real numbers such that for all \( (a_0, b_0, \lambda_0) \in A \times B \times \Lambda \)

\[ a_l = a_{l-1} + b_l, \quad \lambda_l = 1 - \frac{\alpha}{l} + O(l^{-1-\beta}), \quad b_l = q(b_0) + o(l^{-1}), \]

where \( \alpha \in \mathbb{N}^+ \) independent of \( a_0, b_0, \lambda_0, q(b_0) \geq 0 \) is a limit that depends on \( b_0 \), and \( \beta \in (0, 1) \).

Assume the ‘\( O \)’ and ‘\( o \)’ depend only on \( A, B, \Lambda \subset \mathbb{R} \). Then, we have

\[ \sup_{(a_0, b_0, \lambda_0) \in A \times B \times \Lambda} \left| \frac{a_l}{l} - \frac{q}{1+\alpha} \right| = O(l^{-\beta}). \]

**Proof.** Let \( A, B, \Lambda \subset \mathbb{R} \) be three compact sets and \( (a_0, b_0, \lambda_0) \in A \times B \times \Lambda \). It is easy to see that there exists a constant \( G > 0 \) independent of \( a_0, b_0, \lambda_0 \) such that \( |a_l| \leq G \times l + |a_0| \) for all \( l \geq 0 \). Letting \( r_l = \frac{a_l}{l} \), we have that for \( l \geq 2 \)

\[ r_l = r_{l-1}(1 - \frac{1}{l})\left(1 - \frac{\alpha}{l} + O(l^{-1-\beta})\right) + \frac{q}{l} + o(l^{-2}) \]

\[ = r_{l-1}(1 - \frac{1 + \alpha}{l}) + \frac{q}{l} + O(l^{-1-\beta}). \]

where \( O \) bound depends only on \( A, B, \Lambda \). Letting \( x_l = r_l - \frac{q}{l+\alpha} \), there exists \( M > 0 \) that depends only on \( A, B, \Lambda, \) and \( l_0 > 0 \) that depends only on \( \alpha \) such that for all \( l \geq l_0 \)

\[ x_{l-1}(1 - \frac{1 + \alpha}{l}) - \frac{M}{l^{1+\beta}} \leq x_l \leq x_{l-1}(1 - \frac{1 + \alpha}{l}) + \frac{M}{l^{1+\beta}}. \]

Let us deal with the right hand inequality first. By induction, we have that

\[ x_l \leq x_{l_0-1} \prod_{k=l_0}^{l} \left(1 - \frac{1 + \alpha}{k}\right) + M \sum_{k=l_0}^{l} \prod_{j=k+1}^{l} \left(1 - \frac{1 + \alpha}{j}\right) \frac{1}{k^{1+\beta}}. \]

By taking the logarithm of the first term in the right hand side and using the fact that \( \sum_{k=l_0}^{l} \frac{1}{k} = \log(l) + O(1) \), we have

\[ \prod_{k=l_0}^{l} \left(1 - \frac{1 + \alpha}{k}\right) = \Theta(l^{-1-\alpha}). \]
where the bound $\Theta$ does not depend on $l_0$. For the second part, observe that
\[
\prod_{j=k+1}^{l} \left(1 - \frac{1 + \alpha}{j}\right) = \frac{(l - \alpha - 1)!}{l!} \frac{k!}{(k - \alpha - 1)!}
\]
and
\[
\frac{k!}{(k - \alpha - 1)!} \frac{1}{k^{1+\beta}} \sim_{k \to \infty} k^{\alpha - \beta}.
\]
Since $\alpha \geq 1$ ($\alpha \in \mathbb{N}^*$), then the serie with term $k^{\alpha - \beta}$ is divergent and we have that
\[
\sum_{k=l_0}^{l} \frac{k!}{(k - \alpha - 1)!} \frac{1}{k^{1+\beta}} \sim \sum_{k=1}^{l} k^{\alpha - \beta}
\]
\[
\sim \int_{1}^{l} t^{\alpha - \beta} dt
\]
\[
\sim \frac{1}{\alpha - \beta + 1} l^{\alpha - \beta + 1}.
\]
Therefore, it follows that
\[
\sum_{k=l_0}^{l} \prod_{j=k+1}^{l} \left(1 - \frac{1 + \alpha}{j}\right) \frac{1}{k^{1+\beta}} = \frac{(l - \alpha - 1)!}{l!} \sum_{k=l_0}^{l} \frac{k!}{(k - \alpha - 1)!} \frac{1}{k^{1+\beta}} \sim \frac{1}{\alpha} l^{-\beta}.
\]
This proves that
\[
x_l \leq M \alpha l^{-\beta} + o(l^{-\beta}).
\]
where the ‘$o$’ bound depends only on $A, B, \Lambda$. Using the same approach for the left-hand inequality, we prove that
\[
x_l \geq -M \alpha l^{-\beta} + o(l^{-\beta}).
\]
This concludes the proof. 

The next lemma is a different version of the previous lemma which will be useful for other applications.

**Appendix Lemma 8.** Let $A, B, \Lambda \subset \mathbb{R}^+$ be three compact sets, and $(a_l), (b_l), (\lambda_l)$ be three sequences of non-negative real numbers such that for all $(a_0, b_0, \lambda_0) \in A \times B \times \Lambda$
\[
a_l = a_{l-1} \lambda_l + b_l, \quad b_l = q(b_0) + \mathcal{O}(l^{-1}),
\]
\[
\lambda_l = 1 - \frac{\alpha}{l} + \frac{\log(l)}{l^2} + \mathcal{O}(l^{-2}),
\]
where $\alpha \in \mathbb{N}^*$, $\kappa \neq 0$ both do not depend on $a_0, b_0, \lambda_0$, $q(b_0) \in \mathbb{R}^+$ is a limit that depends on $b_0$.
Assume the ‘$\mathcal{O}$’ and ‘$o$’ depend only on $A, B, \Lambda \subset \mathbb{R}$. Then, we have
\[
\sup_{(a_0, b_0, \lambda_0) \in A \times B \times \Lambda} \left| \frac{a_l}{l} - \frac{q}{1 + \alpha} \right| = \Theta(\log(l)l^{-1})
\]

**Proof.** Let $A, B, \Lambda \subset \mathbb{R}$ be three compact sets and $(a_0, b_0, \lambda_0) \in A \times B \times \Lambda$. Similar to the proof of Appendix Lemma 7, there exists a constant $G > 0$ independent of $a_0, b_0, \lambda_0$ such that $|a_l| \leq G \times l + |a_0|$ for all $l \geq 0$, therefore $(a_l/l)$ is bounded. Let $r_l = \frac{a_l}{l}$. We have
\[
r_l = r_{l-1}(1 - \frac{1}{l})(1 - \frac{\alpha}{l} + \frac{\log(l)}{l^2} + \mathcal{O}(l^{-1-\beta})) + \frac{q}{l} + \mathcal{O}(l^{-2})
\]
\[
= r_{l-1}(1 + \frac{\alpha}{l} + r_{l-1} \frac{\log(l)}{l^2} + \frac{q}{l} + \mathcal{O}(l^{-2}).
\]
Let $x_l = r_l - \frac{q}{1+\alpha}$. It is clear that $\lambda_l = 1 - \frac{\alpha}{l} + O(l^{-3/2})$. Therefore, using appendix lemma 7 with $\beta = 1/2$, we have $r_l \to \frac{q}{1+\alpha}$ uniformly over $a_0, b_0, \lambda_0$. Thus, assuming $\kappa > 0$ (for $\kappa < 0$, the analysis is the same), there exists $\kappa_1, \kappa_2, M, l_0 > 0$ that depend only on $A, B, \Lambda$ such that for all $l \geq l_0$

\[
x_{l-1}(1 - \frac{1 + \alpha}{l}) + \kappa_1 \frac{\log(l)}{l^2} - \frac{M}{l^2} \leq x_l \leq x_{l-1}(1 - \frac{1 + \alpha}{l}) + \kappa_2 \frac{\log(l)}{l^2} + \frac{M}{l^2}.
\]

It follows that

\[
x_l \leq x_{l_0} \prod_{k=l_0}^{l} (1 - \frac{1 + \alpha}{k}) + \sum_{k=l_0}^{l} \prod_{j=k+1}^{l} (1 - \frac{1 + \alpha}{j}) \frac{\kappa_1 \log(k) + M}{k^2}
\]
and

\[
x_l \geq x_{l_0} \prod_{k=l_0}^{l} (1 - \frac{1 + \alpha}{k}) + \sum_{k=l_0}^{l} \prod_{j=k+1}^{l} (1 - \frac{1 + \alpha}{j}) \frac{\kappa_1 \log(k) - M}{k^2}.
\]

Recall that we have

\[
\prod_{k=l_0}^{l} (1 - \frac{1 + \alpha}{k}) = \Theta(l^{-1-\alpha})
\]
and

\[
\prod_{j=k+1}^{l} (1 - \frac{1 + \alpha}{j}) = \frac{(l - \alpha - 1)!}{k!} \frac{k!}{(k-\alpha-1)!}
\]
so that

\[
\frac{k!}{(k-\alpha-1)!} \frac{\kappa_1 \log(k) - M}{k^2} \sim k \to \infty \log(k) k^{\alpha-1}.
\]

Therefore, we obtain

\[
\sum_{k=l_0}^{l} \frac{k!}{(k-\alpha-1)!} \frac{\kappa_1 \log(k) - M}{k^2} \sim \sum_{k=1}^{l} \log(k) k^{\alpha-1}
\]
\[
\sim \int_1^{l} \log(t)t^{\alpha-1} \, dt
\]
\[
\sim C_1 l^{\alpha} \log(l),
\]
where $C_1 > 0$ is a constant. Similarly, there exists a constant $C_2 > 0$ such that

\[
\sum_{k=1}^{l} \frac{k!}{(k-\alpha-1)!} \frac{\kappa_2 \log(k) + M}{k^2} \sim C_2 l^{\alpha} \log(l).
\]

Moreover, having that $\frac{(l-\alpha-1)!}{l!} \sim \frac{1}{l^{1-\alpha}}$ yields

\[
x_l \leq C' l^{-1} \log(l) + o(l^{-1} \log(l))
\]
where $C'$ and $'o$ depend only on $A, B, \Lambda$. Using the same analysis, we get

\[
x_l \geq C'' l^{-1} \log(l) + o(l^{-1} \log(l))
\]
where $C''$ and $'o$ depend only on $A, B, \Lambda$, which concludes the proof.
We have that
\[
\sup_{x \in E} |\tilde{K}^L(x, x) - \tilde{K}^\infty(x, x)| = \mathcal{O}(L^{-1})
\]

Moreover, there exists a constant \( \lambda \in (0, 1) \) such that for all \( \epsilon \in (0, 1) \)
\[
\sup_{(x, x') \in B_\epsilon} |\tilde{K}^L(x, x') - \tilde{K}^\infty(x, x')| = \Theta(\log(L)L^{-1}).
\]

where
- if \( \phi \) is ReLU-like, then \( \tilde{K}^\infty(x, x') = \frac{\sigma^2}{d} \|x\| \|x'\| (1 - (1 - \lambda)\mathbb{1}_{x \neq x'}) \).
- if \( \phi \) is Tanh, then \( \tilde{K}^\infty(x, x') = q(1 - (1 - \lambda)\mathbb{1}_{x \neq x'}) \) where \( q > 0 \) is a constant.

**Proof.** We start by proving the results for FFNN, then we generalize them to the case of CNN.

**Case 1: FFNN.** Let \( \epsilon \in (0, 1) \), \( (\sigma_b, \sigma_w) \in \text{EOC} \), \( x, x' \in \mathbb{R}^d \) and recall \( c^l(x, x') = \frac{\eta q^l(x, x')}{\eta q^l(x, x') + q^l(x, x)} \). Let \( \gamma_l := 1 - c^l(x, x') \) and \( f \) be the correlation function defined by the recursive equation \( c^{l+1} = f(c^l) \). By definition, we have that \( q^l(x, x) = 1 \). Therefore
\[
K^l(x, x) = K^{l-1}(x, x) + \frac{\sigma_w^2}{d} \|x\|^2 = \frac{l \sigma_w^2}{d} \|x\|^2 = l \tilde{K}^\infty(x, x)
\]

which concludes the proof for \( K^L(x, x) \). Note that the results is 'exact' for ReLU, which means the upper bound \( \mathcal{O}(L^{-1}) \) is valid but not optimal in this case. However, we will see that this bound is optimal for Tanh.

From Appendix Lemma 1, we have that
\[
\sup_{(x, x') \in B_\epsilon} \left| c^l(x, x') - 1 + \frac{\kappa}{l^2} - 3 \sqrt{\kappa \frac{\log(l)}{l^3}} \right| = \mathcal{O}(l^{-3})
\]
where \( \kappa = \frac{9 \pi^2}{2} \). Moreover, we have that
\[
\sup_{(x, x') \in B_\epsilon} \left| f'(c^l(x, x')) - 1 + \frac{3}{l} - \frac{9}{2 \sqrt{\kappa}} \frac{\log(l)}{l^2} \right| = \mathcal{O}(l^{-2}).
\]

Using Appendix Lemma 8 with \( a_l = K^{l+1}(x, x'), b_l = q^{l+1}(x, x'), \lambda_l = f'(c^l(x, x')) \), we conclude that
\[
\sup_{(x, x') \in B_\epsilon} \left| \frac{K^{l+1}(x, x')}{l} - \frac{1}{4} \frac{\sigma_w^2}{d} \|x\| \|x'\| \right| = \Theta(\log(l)l^{-1})
\]
Using the compactness of \( B_\epsilon \), we conclude that
\[
\sup_{(x, x') \in B_\epsilon} \left| \frac{K^l(x, x')}{l} - \frac{1}{4} \frac{\sigma_w^2}{d} \|x\| \|x'\| \right| = \Theta(\log(l)l^{-1})
\]

- **\( \phi = \text{Tanh} \):** The case of Tanh is similar to that of ReLU with small differences in technical lemmas used to conclude. From Appendix Lemma 2, we have that
\[
\sup_{(x, x') \in B_\epsilon} \left| c^l(x, x') - 1 + \frac{\kappa}{l} - \kappa (1 - \kappa^2) \frac{\log(l)}{l^3} \right| = \mathcal{O}(l^{-3})
\]
where $\kappa = \frac{2}{\theta^2} > 0$ and $\zeta = \frac{f(1)}{b} > 0$. Moreover, we have that

$$\sup_{(x,x') \in B_r} \left| f'(c(x,x')) - 1 + \frac{2}{l} - 2(1 - \kappa^2 \zeta) \frac{\log(l)}{l^2} \right| = O(l^{-2}).$$

We conclude in the same way as in the case of ReLU using Appendix Lemma 8. The only difference is that, in this case, the limit of the sequence $b_l = q^{l+1}(x, x')$ is the limiting variance $q$ (from facts 3, 1) does not depend on $(x, x').$

**Case 2: CNN.** Under Assumption 1, the NTK of a CNN is the same as that of an FFNN. Therefore, the results on the NTK of FFNN are all valid to the NTK of CNN $K_{\alpha,\alpha'}^l$ for any $\alpha, \alpha'.$

### 6.2. Proofs of the results of Section 3.2 on ResNets

In this section, we provide proofs for lemmas 3 and 4 together with Theorem 3 and proposition 2 on ResNets.

Lemma 3 in the paper gives the recursive formula for the mean-field NTK of a ResNet with Fully Connected blocks.

**Lemma 3 (NTK of a ResNet with Fully Connected blocks in the infinite width limit).** Let $x, x'$ be two inputs and $K^{\text{res}, l}$ be the exact NTK for the Residual Network with 1 layer. Then, we have

- **For the first layer (without residual connections),** we have for all $x, x' \in \mathbb{R}^d$

  $$K^{\text{res}, 1}_{ii'}(x, x') = \delta_{ii'} \left( \sigma_b^2 + \frac{\sigma_w^2}{d} x \cdot x' \right),$$

  where $x \cdot x'$ is the inner product in $\mathbb{R}^d$.

- **For $l \geq 2,$ as $n_1, n_2, \ldots, n_{L-1} \to \infty,$ we have for all $i, i' \in [1 : n_1], K^{\text{res}, l}_{ii'}(x, x') = \delta_{ii'} K^{\text{res}, l-1}_{ii}(x, x')$, where $K^{\text{res}, l-1}_{ii}(x, x')$ is given by the recursive formula have for all $x, x' \in \mathbb{R}^d$ and $l \geq 2,$ as $n_1, n_2, \ldots, n_l \to \infty$ recursively, we have

  $$K^{\text{res}, l}_{ii}(x, x') = K^{\text{res}, l-1}_{ii}(x, x')(\tilde{q}^l(x, x') + 1) + \tilde{q}^l(x, x').$$

**Proof.** The first result is the same as in the FFNN case since we assume there is no residual connections between the first layer and the input. We prove the second result by induction.

- Let $x, x' \in \mathbb{R}^d.$ We have

  $$K^{\text{res}, 1}_{ii}(x, x') = \sum_j \frac{\partial y^1_{i}(x)}{\partial w_{1ij}} \frac{\partial y^1_{i}(x)}{\partial w_{1ij}} + \frac{\partial y^1_{i}(x)}{\partial b_1} \frac{\partial y^1_{i}(x)}{\partial b_1} = \frac{\sigma_w^2}{d} x \cdot x' + \sigma_b^2.$$

- The proof is similar to the FeedForward network NTK. For $l \geq 2$ and $i \in [1 : n_1]$

  $$\partial_{b_{ij},y^l_{i,j}+1}(x) = \partial_{b_{ij},y^l_{i,j}}(x) + \frac{\sigma_w}{\sqrt{n_l}} \sum_{j=1}^{n_l} w_{ij}^{l+1} \phi'(y^l_{j}(x)) \partial_{b_{ij},y^l_{i,j}}(x).$$

Therefore, we obtain

$$(\partial_{b_{ij},y^l_{i,j}+1}(x))(\partial_{b_{ij},y^l_{i,j}+1}(x'))^t = (\partial_{b_{ij},y^l_{i,j}}(x))(\partial_{b_{ij},y^l_{i,j}}(x'))^t + \frac{\sigma_w}{\sqrt{n_l}} \sum_{j=1}^{n_l} w_{ij}^{l+1} \phi'(y^l_{j}(x)) \phi'(y^l_{j}(x')) \partial_{b_{ij},y^l_{i,j}}(x)(\partial_{b_{ij},y^l_{i,j}}(x'))^t + I$$

where

$$I = \frac{\sigma_w}{\sqrt{n_l}} \sum_{j=1}^{n_l} w_{ij}^{l+1} \phi'(y^l_{j}(x)) \partial_{b_{ij},y^l_{i,j}}(x)(\partial_{b_{ij},y^l_{i,j}}(x'))^t.$$
Using the induction hypothesis, as $n_0, n_1, \ldots, n_{l-1} \to \infty$, we have that
\[
(\partial_{\theta_{l-1}} y_{l,i}^{\ell+1}(x)) (\partial_{\theta_{l-1}} y_{l,i}^{\ell+1}(x'))^t + \frac{\sigma_w^2}{n_{l-1}} \sum_{j,j'} w_{ij}^{\ell+1} w_{ij'}^{\ell+1} \phi'(y_j(x)) \phi'(y_{j'}(x')) \partial_{\theta_{l-1}} y_{l,j}^{\ell}(x) (\partial_{\theta_{l-1}} y_{l,j'}^{\ell}(x'))^t + I
\]
\[
\to K_{\text{res}}^l(x, x') + \frac{\sigma_w^2}{n_{l-1}} \sum_{j} (w_{ij}^{\ell+1})^2 \phi'(y_j(x)) \phi'(y_j(x')) K_{\text{res}}^l(x, x') + I',
\]
where $I' = \frac{\sigma_w^2}{n_{l-1}} w_{ii}^{\ell+1} (\phi'(y_i(x)) + \phi'(y_i(x')))$ $K_{\text{res}}^l(x, x').$

As $n_l \to \infty$, we have that $I' \to 0$. Using the law of large numbers, as $n_l \to \infty$
\[
\frac{\sigma_w^2}{n_l} \sum_{j} (w_{ij}^{\ell+1})^2 \phi'(y_j(x)) \phi'(y_j(x')) K_{\text{res}}^l(x, x') \to q^l+1(x, x') K_{\text{res}}^l(x, x').
\]
Moreover, we have that
\[
(\partial_{w^l+1} y_{i}^{\ell+1}(x)) (\partial_{w^l+1} y_{i}^{\ell+1}(x'))^t + (\partial_{b^l+1} y_{i}^{\ell+1}(x)) (\partial_{b^l+1} y_{i}^{\ell+1}(x'))^t = \frac{\sigma_w^2}{n_l} \sum_{j} \phi(y_j(x)) \phi(y_j(x')) + \sigma_b^2
\]
\[
\to_{n_l \to \infty} \sigma_w^2 E[\phi(y_i(x)) \phi(y_i(x'))] + \sigma_b^2 = q^l+1(x, x').
\]
We conclude using the fact that
\[\partial_{\theta_{1:l-1}} y_{l,\alpha}(x) \partial_{\theta_{1:l-1}} y_{l',\alpha'}(x)^T = \partial_{\theta_{1:l-1}} y_{l-1,\alpha}(x) \partial_{\theta_{1:l-1}} y_{l'-1,\alpha'}(x)^T + \frac{\sigma_w^2}{n_{l-1}(2k+1)} \sum_{j,\beta} w_{l,j,\beta} w_{l',j',\beta'} \phi'(y_{j,\alpha}^{-1}) \phi'(y_{j',\alpha'}^{-1}) \partial_{\theta_{1:l-1}} y_{l-1,\alpha+\beta}(x) \partial_{\theta_{1:l-1}} y_{l'-1,\alpha'+\beta}(x)^T + I,\]
where
\[I = \frac{\sigma_w}{\sqrt{n_{l-1}(2k+1)}} \sum_{j,\beta} w_{l,j,\beta} \phi'(y_{j,\alpha}^{-1}) \partial_{\theta_{1:l-1}} y_{l-1,\alpha}(x) \partial_{\theta_{1:l-1}} y_{l-1,\alpha}(x)^T + \partial_{\theta_{1:l-1}} y_{l-1,\alpha+\beta}(x) \partial_{\theta_{1:l-1}} y_{l-1,\alpha}(x)^T.\]

As \(n_1, n_2, \ldots, n_{l-2} \to \infty\) and using the induction hypothesis, we have
\[\partial_{\theta_{1:l-1}} y_{l,\alpha}(x) \partial_{\theta_{1:l-1}} y_{l',\alpha'}(x)^T \to \delta_{l'} K_{\alpha,\alpha'}^{-1}(x, x') + \frac{\sigma_w^2}{n_{l-1}(2k+1)} \sum_{j,\beta} w_{l,j,\beta} w_{l',j',\beta'} \phi'(y_{j,\alpha}^{-1}) \phi'(y_{j',\alpha'}^{-1}) K_{\alpha+\beta,\alpha'+\beta}^{-1}(x, x'),\]
where \(f'(c_{\alpha+\beta,\alpha'+\beta}(x, x')) = \sigma_w^2 \mathbb{E}[\phi'(y_{j,\alpha}^{-1}) \phi'(y_{j',\alpha'}^{-1})].\)

We conclude using the fact that
\[\partial_{\theta_{l,l}} y_{l,\alpha}(x) \partial_{\theta_{l,l}} y_{l',\alpha'}(x)^T \to \delta_{l'} \left(\frac{\sigma_w^2}{2k+1} \sum_{\beta} \mathbb{E}[\phi'(y_{l+\beta}(x)) \phi'(y_{l'+\beta}(x'))] + \frac{\sigma_w^2}{\beta^3}\right).\]

Before moving to the main theorem on ResNets, we first prove a Lemma on the asymptotic behaviour of \(c^l\) for ResNet.

**Appendix Lemma 9** (Asymptotic expansion of \(c^l\) for ResNet). Let \(\epsilon \in (0, 1)\) and \(\sigma_w > 0\). We have for FFNN
\[\sup_{(x, x') \in B_1} \left| c^l(x, x') - 1 + \frac{\kappa \sigma_w}{l^2} - 3\sqrt{\frac{\log(l)}{l^3}} \right| = O(l^{-3})\]
where \(\kappa = \frac{9 \sigma_w^2}{2} (1 + \frac{1}{\sigma_w^2})^2\). Moreover, we have that
\[\sup_{(x, x') \in B_1} \left| f'(c^l(x, x')) - 1 + \frac{3(1 + \frac{\epsilon}{\pi})}{l} - \frac{3 \sqrt{2 \log(l)}}{2 \pi \epsilon} \right| = O(l^{-2}).\]

where \(f\) is the ReLU correlation function given in fact 12. Moreover, this result holds also for CNNs where the supremum should be replaced by \(\sup_{(x, x') \in B_1} \sup_{\alpha, \alpha'}\).

**Proof.** We first prove the result for ResNet with fully connected layers, then we generalize it to convolutional layers. Let \(\epsilon \in (0, 1)\).

- Let \(x \neq x' \in \mathbb{R}^d\), and \(c^l := c^l(x, x')\). It is straightforward that the variance terms follow the recursive form
\[q^l(x, x) = q^{l-1}(x, x) + \sigma_w^2 / 2q^{l-1}(x, x) = (1 + \sigma_w^2 / 2)^{l-1} q^1(x, x)\]
Leveraging this observation, we have that

\[ c^{l+1} = \frac{1}{1 + \alpha} c^l + \frac{\alpha}{1 + \alpha} f(c^l), \]

where \( f \) is the ReLU correlation function given in fact 12 and \( \alpha = \frac{\sigma_w^2}{2} \). Recall that

\[ f(c) = \frac{1}{\pi c} \arcsin(c) + \frac{1}{\pi \sqrt{1 - c^2}} + \frac{1}{2} c. \]

As in the proof of Appendix Lemma 1, let \( \gamma_l = 1 - c^l \), therefore, using Taylor expansion of \( f \) near 1 given in fact 14 yields

\[ \gamma_{l+1} = \gamma_l - \frac{\alpha s}{1 + \alpha} \gamma_l^{3/2} - \frac{\alpha b}{1 + \alpha} \gamma_l^{5/2} + O(\gamma_l^{7/5}). \]

This form is exactly the same as in the proof of Appendix Lemma 1 with \( s' = \frac{\alpha s}{1+\alpha} \) and \( b' = \frac{\alpha b}{1+\alpha} \). Thus, following the same analysis we conclude.

For the second result, observe that the derivation is the same as in Appendix Lemma 1.

• Under Assumption 1, results of FFNN hold for CNN.

The next theorem shows that no matter what the choice of \( \sigma_w > 0 \), the normalized NTK of a ResNet will always have a subexponential convergence rate to a limiting \( \tilde{K}_{\text{res}}^{\infty} \).

**Theorem 2 (NTK for ResNet).** Consider a ResNet satisfying

\[ y^l(x) = y^{l-1}(x) + \mathcal{F}(w^l, y^{l-1}(x)), \quad l \geq 2, \quad (17) \]

where \( \mathcal{F} \) is either a convolutional or dense layer (equations (3) and (4)) with ReLU activation. Let \( K_{\text{res}}^L \) be the corresponding NTK and \( \tilde{K}_{\text{res}}^L = K_{\text{res}}^L / \alpha_L \) (Normalized NTK) with \( \alpha_L = L(1 + \frac{\sigma_w^2}{4}) L^{-1} \). If the layers are convolutional assume Assumption 1 holds. Then, we have

\[ \sup_{x \in E} |\tilde{K}_{\text{res}}^L(x, x) - \tilde{K}_{\text{res}}^{\infty}(x, x)| = \Theta(L^{-1}) \]

Moreover, there exists a constant \( \lambda \in (0, 1) \) such that for all \( \epsilon \in (0, 1) \)

\[ \sup_{x, x' \in B_s} |\tilde{K}_{\text{res}}^L(x, x') - \tilde{K}_{\text{res}}^{\infty}(x, x')| = \Theta(L^{-1} \log(L)), \]

where \( \tilde{K}_{\text{res}}^{\infty}(x, x') = \frac{\sigma_w^2 \|x\| \|x'\|}{d} (1 - (1 - \lambda) 1_{x \neq x'}) \).

**Proof.** Case 1: ResNet with Fully Connected layers.

Let \( \epsilon \in (0, 1) \) and \( x \neq x' \in \mathbb{R}^d \). We first prove the result for the diagonal term \( K_{\text{res}}^{L_l}(x, x) \) then \( K_{\text{res}}^{L_l}(x, x') \).

• Diagonal terms: using properties on the correlation function \( f \) (fact 12), we have that \( q^l(x, x) = \frac{\sigma_w^2}{2} f(1) = \frac{\sigma_w^2}{2} \). Moreover, it is easy to see that the variance terms for a ResNet follow the recursive formula \( q^l(x, x) = q^{l-1}(x, x) + \sigma_w^2 / 2 \times q^{l-1}(x, x) \), hence

\[ q^l(x, x) = (1 + \sigma_w^2 / 2)^{l-1} \frac{\sigma_w^2}{d} \|x\|^2 \quad (18) \]

Recall that the recursive formula of NTK of a ResNet with FFNN layers is given by (Appendix Lemma 3)

\[ K_{\text{res}}^{L_l}(x, x') = K_{\text{res}}^{L_{l-1}}(x, x')(q^l(x, x') + 1) + q^l(x, x') \]
Hence, for the diagonal terms we obtain

$$K_{\text{res}}^{l}(x, x) = K_{\text{res}}^{l-1}(x, x)(\frac{\sigma_w^2}{2} + 1) + q^l(x, x)$$

Letting $\hat{K}_{\text{res}}^{l} = K_{\text{res}}^{l}/(1 + \frac{\sigma_w^2}{2})^{l-1}$ yields

$$\hat{K}_{\text{res}}^{l}(x, x) = \hat{K}_{\text{res}}^{l-1}(x, x) + \frac{\sigma_w^2}{d} \|x\|^2$$

Therefore, $\hat{K}_{\text{res}}^{l}(x, x) = \hat{K}_{\text{res}}^{l}(x, x) + (1 - 1/l) \frac{\sigma_w^2}{2} ||x||^2$, the conclusion is straightforward since $E$ is compact and $(K')_{\text{res}}^{l}(x, x)$ is continuous (hence, bounded on $E$).

- The argument is similar to that of Theorem 1 with few differences. From appendix lemma 9 we have that

$$\sup_{(x,x') \in B_\epsilon} \left| \frac{c^l(x, x') - 1}{2} - \frac{\kappa}{2l^2} - 3\sqrt{\kappa} \frac{\log(l)}{l^3} \right| = \mathcal{O}(l^{-3})$$

where $\kappa = \frac{\sigma_w^2}{2} (1 + \frac{2}{\sigma_w^2})^2$. Moreover, we have that

$$\sup_{(x,x') \in B_\epsilon} \left| f'(c^l(x, x')) - 1 + \frac{3(1 + \frac{2}{\sigma_w^2})}{l} - \frac{3\sqrt{2} \log(l)}{2l^2} \right| = \mathcal{O}(l^{-2})$$

Let $\alpha = \frac{\sigma_w^2}{2}$. We also have $q^{l+1}(x, x') = \alpha f'(c^l(x, x'))$ where $f$ is the ReLU correlation function given in fact 12. It follows that for all $(x, x') \in B_\epsilon$

$$1 + q^{l+1}(x, x') = (1 + \alpha)(1 - 3l^{-1} + \frac{\log(l)}{l^2} + \mathcal{O}(l^{-3}))$$

for some constant $\zeta \neq 0$ that does not depend on $x, x'$. The bound $\mathcal{O}$ does not depend on $x, x'$ either. Now let $a_l = \frac{K_{\text{res}}^{l+1}(x, x')}{(1 + \alpha)^{l+1}}$. Using the recursive formula of the NTK, we obtain

$$a_l = \lambda_l a_{l-1} + b_l$$

where $\lambda_l = 1 - 3l^{-1} + \frac{\log(l)}{l^2} + \mathcal{O}(l^{-3})$, $b_l = \frac{\sigma_w^2}{2} \sqrt{\|x\| \|x'\|} f'(c^l(x, x')) = q(x, x') + \mathcal{O}(l^{-2})$ with $q(x, x') = \frac{\sigma_w^2}{2} \sqrt{\|x\| \|x'\|}$ and where we used the fact that $c^l(x, x') = 1 + \mathcal{O}(l^{-2})$ (Appendix Lemma 1) and the formula for ResNet variance terms given by equation (18). Observe that all bounds $\mathcal{O}$ are independent from the inputs $(x, x')$. Therefore, using Appendix Lemma 8, we have

$$\sup_{x, x' \in B_\epsilon} \left| K_{\text{res}}^{L+1}(x, x')/L(1 + \alpha)^L - K_{\text{res}}^\infty(x, x') \right| = \Theta(L^{-1} \log(L)),$$

which can also be written as

$$\sup_{x, x' \in B_\epsilon} \left| K_{\text{res}}^{L}(x, x')/(L - 1)(1 + \alpha)^{L-1} - K_{\text{res}}^\infty(x, x') \right| = \Theta(L^{-1} \log(L)),$$

We conclude by observing that $K_{\text{res}}^{L}(x, x')/(L - 1)(1 + \alpha)^{L-1} = K_{\text{res}}^{L}(x, x')/L(1 + \alpha)^{L-1} + \mathcal{O}(L^{-1})$ where $\mathcal{O}$ can be chosen to depend only on $\epsilon$.

**Case 2: ResNet with Convolutional layers.** Under Assumption 1, the dynamics of the correlation and NTK are exactly the same for FFNN, hence all results on FFNN apply to CNN.

Now let us prove the Scaled Resnet result. Before that, we prove the following Lemma
Appendix Lemma 10. Consider a Residual Neural Network with the following forward propagation equations
\[ y^l(x) = y^{l-1}(x) + \frac{1}{\sqrt{l}}F(w^l, y^{l-1}(x)), \quad l \geq 2. \] (19)

where \( F \) is either a convolutional or dense layer (equations 3 and 4) with ReLU activation. Then there exists \( \zeta, \nabla > 0 \) such that for all \( \epsilon \in (0, 1) \)
\[ \sup_{(x,x') \in B(\epsilon)(l)} \left| 1 - c^l(x,x') - \frac{\zeta}{\log(l)^2} + \frac{\nabla}{\log(l)^3} \right| = o\left( \frac{1}{\log(l)^3} \right) \]
where the bound \( 'o' \) depends only on \( \epsilon \).

For CNN, under Assumption 1, the result holds and the supremum is taken also over \( \alpha, \alpha' \), i.e.
\[ \sup_{(x,x') \in B(\epsilon)(\alpha, \alpha')} \left| 1 - c^l_{\alpha, \alpha'}(x,x') - \frac{\zeta}{\log(l)^2} + \frac{\nabla}{\log(l)^3} \right| = o\left( \frac{1}{\log(l)^3} \right) \]

Proof. We first start with the dense layer case. Let \( \epsilon \in (0, 1) \) and \( (x,x') \in B(\epsilon) \) be two inputs and denote by \( c^l := c^l(x,x') \). Following the same machinery as in the proof of Appendix Lemma 9, we have that
\[ c^l = \frac{1}{1 + \alpha_l}c^{l-1} + \frac{\alpha_l}{1 + \alpha_l}f(c^{l-1}) \]
where \( \alpha_l = \frac{s^2}{2l} \). Using fact 12, it is straightforward that \( f' \geq 0 \), hence \( f \) is non-decreasing. Therefore, \( c^l \geq c^{l-1} \) and \( c^l \) converges to a fixed point \( c \). Let us prove that \( c = 1 \). By contradiction, suppose \( c < 1 \) so that \( f(c) - c > 0 \) (\( f \) has a unique fixed point which is 1). This yields
\[ c^l - c = c^{l-1} - c + \frac{f(c) - c}{l} + O\left( \frac{c^l}{l} \right) + O(l^{-2}) \]
by summing, this leads to \( c^l - c \sim (f(c) - c) \log(l) \) which is absurd since \( f(c) \neq c \) ( \( f \) has only 1 as a fixed point).

We conclude that \( c = 1 \). Using the non-decreasing nature of \( f \), it is easy to conclude that the convergence is uniform over \( B(\epsilon) \).

Now let us find the asymptotic expansion of \( 1 - c^l \). Recall the Taylor expansion of \( f \) near 1 given in fact 14
\[ f(c) = 1 - c + s(1-c)^{3/2} + b(1-c)^{5/2} + O((1-c)^{7/2}) \] (20)
where \( s = \frac{2\sqrt{7}}{3\pi} \) and \( b = \frac{\sqrt{7}}{3\pi} \). Letting \( \gamma_l = 1 - c^l \), we obtain
\[ \gamma_l = \gamma_{l-1} - s\delta_l \gamma_{l-1}^{3/2} - b\delta_l \gamma_{l-1}^{5/2} + O(\delta_l \gamma_{l-1}^{7/2}) \]
which yields
\[ \gamma_l^{1/2} = \gamma_{l-1}^{1/2} + \frac{s}{2} \delta_l + \frac{3}{8} s^2 \delta_l^2 \gamma_{l-1}^{1/2} + \frac{b}{2} \delta_l \gamma_{l-1} + O(\delta_l \gamma_{l-1}^{3/2}) \] (21)
therefore, we have that
\[ \gamma_l^{1/2} \sim \frac{s\sigma_w^2}{4} \log(l) \]
and \( 1 - c^l \sim \frac{\zeta}{\log(l)^2} \) where \( \zeta = 16/s^2\sigma_w^4 \).

we can further expand the asymptotic approximation to have
\[ 1 - c^l = \frac{\zeta}{\log(l)^2} - \frac{\nabla}{\log(l)^3} + o\left( \frac{1}{\log(l)^3} \right) \]
where \( \nabla > 0 \) the ‘\( o \)’ holds uniformly for \( (x,x') \in B(\epsilon) \) as in the proof of Appendix Lemma 1.

This result holds for a ResNet with CNN layers under Assumption 1 since the dynamics are the same in this case.
Theorem 2 becomes \( \alpha \).

Proof. We use the same techniques as in the non scaled case. Let us prove the result for fully connected layers, the proof for convolutional layers follows the same analysis. Let \( \epsilon \) where \( F \).

Proposition 2

• Recall that \( \lambda \) where \( \zeta \) where \( \sigma \).

Now we proceed in the same way as in the proof of Appendix Lemma 8. Let \( x_t = \frac{a_t}{l} - q \), then there exists \( M_1, M_2 > 0 \) such that

\[
x_{l-1}(1 - \frac{1}{l}) - M_1 l^{-1} \log(l)^{-1} \leq x_t \leq x_{l-1}(1 - \frac{1}{l}) - M_2 l^{-1} \log(l)^{-1}
\]
therefore, there exists \( l_0 \) independent of \((x, x')\) such that for all \( l \geq l_0 \)

\[
x_l \leq x_{l_0} \prod_{k=l_0}^{l} \left(1 - \frac{1}{k}\right) - M_2 \sum_{k=l_0}^{l} \prod_{j=k+1}^{l} \left(1 - \frac{1}{j}\right) k^{-1} \log(k)^{-1}
\]

and

\[
x_l \geq x_{l_0} \prod_{k=l_0}^{l} \left(1 - \frac{1}{k}\right) - M_1 \sum_{k=l_0}^{l} \prod_{j=k+1}^{l} \left(1 - \frac{1}{j}\right) k^{-1} \log(k)^{-1}
\]

after simplification, we have that

\[
\sum_{k=l_0}^{l} \prod_{j=k+1}^{l} \left(1 - \frac{1}{j}\right) k^{-1} \log(k)^{-1} = \Theta \left( \frac{1}{l} \int_{1}^{l} \frac{1}{\log(t)} dt \right) = \Theta(\log(l)^{-1})
\]

where we have used the asymptotic approximation of the Logarithmic Integral function \( \text{Li}(x) = \int_{1}^{x} \frac{1}{\log(t)} \sim_{x \to \infty} \frac{x}{\log(x)} \)

we conclude that \( \alpha_L = L \times \prod_{k=1}^{l} \left(1 + \sigma^2 / 2k\right) \sim L^{1 + \frac{d-2}{2}} \) and the convergence rate of the NTK is now \( \Theta(\log(L)^{-1}) \) which is better than \( \Theta(L^{-1}) \). The convergence is uniform over the set \( \mathcal{B}_r \).

In the limit of large \( L \), the matrix NTK of the scaled resnet has the following form

\[
A_{K_{\text{res}}}^\dagger = qU + \log(L)^{-1} \Theta(M_L)
\]

where \( U \) is the matrix of ones, and \( M_L \) has all elements but the diagonal equal to 1 and the diagonal terms are \( \Theta(L^{-1} \log(L)) \to 0 \). Therefore, \( M_L \) is inversible for large \( L \) which makes \( K_{\text{res}}^\dagger \) also inversible. Moreover, observe that the convergence rate for scaled resnet is \( \log(L)^{-1} \) which means that for the same depth \( L \), the NTK remains far more expressive for scaled resnet compared to standard resnet, this is particularly important for the generalization.

\[
\square
\]

### 6.3. Spectral decomposition of the limiting NTK

#### 6.3.1. Review on Spherical Harmonics

We start by giving a brief review of the theory of Spherical Harmonics (MacRobert, 1967). Let \( \mathbb{S}^{d-1} \) be the unit sphere in \( \mathbb{R}^d \) defined by \( \mathbb{S}^{d-1} = \{ x \in \mathbb{R}^d : \|x\|_2 = 1 \} \). For some \( k \geq 1 \), there exists a set \( \{Y_{k,j}\}_{1 \leq j \leq N(d, k)} \) of Spherical Harmonics of degree \( k \) with \( N(d, k) = \frac{2k + d - 2}{k - 2} \binom{k + d - 3}{d - 2} \).

The set of functions \( \{Y_{k,j}\}_{k \geq 1, j \in [1:N(d, k)]} \) form an orthonormal basis with respect to the uniform measure on the unit sphere \( \mathbb{S}^{d-1} \).

For some function \( g \), the Hecke-Funk formula is given by

\[
\int_{\mathbb{S}^{d-1}} g(⟨x, w⟩) Y_{k,j}(w) \nu_{d-1}(w) = \frac{\Omega_{d-1}}{\Omega_d} Y_{k,j}(x) \int_{-1}^{1} g(t) P_k^d(t) (1 - t^2)^{(d-3)/2} dt
\]

where \( \nu_{d-1} \) is the uniform measure on the unit sphere \( \mathbb{S}^{d-1} \), \( \Omega_d \) is the volume of the unit sphere \( \mathbb{S}^{d-1} \), and \( P_k^d \) is the multi-dimensional Legendre polynomials given explicitly by Rodrigues’ formula

\[
P_k^d(t) = \left( -\frac{1}{2} \right)^k \frac{\Gamma(d+1)}{\Gamma(k + \frac{d-2}{2})} (1 - t^2)^{\frac{d-2}{2}} \left( \frac{d}{dt} \right)^k (1 - t^2)^{k + \frac{d-3}{2}}
\]

\( (P_k^d)_{k \geq 0} \) form an orthogonal basis of \( L^2([-1, 1], (1 - t^2)^{\frac{d-3}{2}} dt) \), i.e.

\[
\langle P_k^d, P_{k'}^d \rangle_{L^2([-1, 1](1 - t^2)^{\frac{d-3}{2}} dt)} = \delta_{k, k'}
\]
where $\delta_{ij}$ is the Kronecker symbol. Moreover, we have

$$||P^d_k||^2_{L^2([-1,1],(1-t^2)^{d-3}dt)} = \frac{(k+d-3)!}{(d-3)(k-d+3)!}$$

Using the Heck-Funk formula, we can easily conclude that any dot product kernel on the unit sphere $\mathbb{S}^{d-1}$, i.e. and kernel of the form $\kappa(x,x') = g(\langle x, x' \rangle)$ can be decomposed on the Spherical Harmonics basis. Indeed, for any $x, x' \in \mathbb{S}^{d-1}$, the decomposition on the spherical harmonics basis yields

$$\kappa(x,x') = \sum_{k \geq 0} \sum_{j=1}^{N(d,k)} \left[ \Omega_{d-1} \Omega_d \int_{-1}^{1} g(t) P^d_k(t)(1-t^2)^{(d-3)/2}dt \right] Y_{k,j}(x) Y_{k,j}(x')$$

we conclude that

$$\kappa(x,x') = \sum_{k \geq 0} \mu_k \sum_{j=1}^{N(d,k)} Y_{k,j}(x) Y_{k,j}(x')$$

where $\mu_k = \frac{\Omega_{d-1} \Omega_d}{144} \int_{-1}^{1} g(t) P^d_k(t)(1-t^2)^{(d-3)/2}dt$.

We use these result in the proof of the next theorem.

**Theorem 3 (Spectral decomposition).** Let $\kappa^L$ be either, the NTK ($K^L$) for an FFNN with $L$ layers initialized on the Ordered phase, The Average NTK ($\bar{K}^L$) for an FFNN with $L$ layers initialized on the EOC, or the Normalized NTK ($\bar{K}_{res}^L$) for a ResNet with $L$ layers (Fully Connected). Then, for all $L \geq 1$, there exists $(\mu^L_k)_{k \geq 1}$ such that for all $x, x' \in \mathbb{S}^{d-1}$

$$\kappa^L(x,x') = \sum_{k \geq 0} \mu^L_k \sum_{j=1}^{N(d,k)} Y_{k,j}(x) Y_{k,j}(x')$$

$(Y_{k,j})_{k \geq 0, j \in [1:N(d,k)]}$ are spherical harmonics of $\mathbb{S}^{d-1}$, and $N(d,k)$ is the number of harmonics of order $k$.

Moreover, we have that $0 < \mu^0_L = \lim_{L \to \infty} \mu^L_0 < \infty$, and for all $k \geq 1, \lim_{L \to \infty} \mu^L_k = 0$.

**Proof.** From the recursive formulas of the NTK for FFNN, CNN and ResNet architectures, it is straightforward that on the unit sphere $\mathbb{S}^{d-1}$, the kernel $\kappa^L$ is zonal in the sense that it depends only on the scalar product, more precisely, for all $L \geq 1$, there exists a function $g_L$ such that for all $x, x' \in \mathbb{S}^{d-1}$

$$\kappa^L(x,x') = g_L(\langle x, x' \rangle)$$

using the previous results on Spherical Harmonics, we have that for all $x, x' \in \mathbb{S}^{d-1}$

$$\kappa^L(x,x') = \sum_{k \geq 0} \mu^L_k \sum_{j=1}^{N(d,k)} Y_{k,j}(x) Y_{k,j}(x')$$

where $\mu^L_k = \frac{\Omega_{d-1} \Omega_d}{144} \int_{-1}^{1} g_L(t) P^d_k(t)(1-t^2)^{(d-3)/2}dt$.

For $k = 0$, we have that for all $L \geq 1, \mu^L_0 = \frac{\Omega_{d-1} \Omega_d}{144} \int_{-1}^{1} g_L(t)(1-t^2)^{(d-3)/2}dt$. By a simple dominated convergence argument, we have that $\lim_{L \to \infty} \mu^L_0 = q \lambda \frac{\Omega_{d-1} \Omega_d}{144} \int_{-1}^{1}(1-t^2)^{(d-3)/2}dt > 0$, where $q, \lambda$ are given in Theorems 1, 2 and Proposition 1 (where we take $q = 1$ for the Ordered/Chaotic phase initialization in Proposition 1). Using the same argument, we have that for $k \geq 1, \lim_{L \to \infty} \mu^L_k = q \lambda \frac{\Omega_{d-1} \Omega_d}{144} \int_{-1}^{1} t^d_k(t)(1-t^2)^{(d-3)/2}dt = q \lambda \frac{\Omega_{d-1} \Omega_d}{144} \langle P^d_0, P^d_k \rangle_{L^2([-1,1](1-t^2)^{(d-3)/2}dt)} = 0$. 

\[]
References

Arora, S., S. Du, W. Hu, Z. Li, and R. Wand (2019). Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. *ICML*.

Arora, S., S. Du, W. Hu, Z. Li, R. Salakhutdinov, and R. Wang (2019). On exact computation with an infinitely wide neural net. *arXiv preprint arXiv:1904.11955*.

Bietti, A. and J. Mairal (2019). On the inductive bias of neural tangent kernels. *NeurIPS 2019*.

Cao, Y., Z. Fang, Y. Wu, D. Zhou, and Q. Gu (2020). Towards understanding the spectral bias of deep learning. *arXiv prePrint 1912.01198*.

Cao, Y. and Q. Gu (2019). Generalization bounds of stochastic gradient descent for wide and deep neural networks. *NeurIPS*.

Chizat, L. and F. Bach (2018). A note on lazy training in supervised differentiable programming. *arXiv preprint arXiv:1812.07956*.

Du, S., J. Lee, H. Li, L. Wang, and X. Zhai (2019). Gradient descent finds global minima of deep neural networks. *ICML*.

Du, S., J. Lee, Y. Tian, B. Poczos, and A. Singh (2018). Gradient descent learns one-hidden-layer CNN: Don’t be afraid of spurious local minima. *ICML*.

Du, S., X. Zhai, B. Poczos, and A. Singh (2019). Gradient descent provably optimizes over-parameterized neural networks. *ICLR*.

Geifman, A., A. Yadav, Y. Kasten, M. Galun, D. Jacobs, and R. Basri (2020). On the similarity between the laplace and neural tangent kernels. *NeurIPS*.

Ghorbani, B., S. Mei, T. Misakiewicz, and A. Montanari (2019). Linearized two-layers neural networks in high dimension. *arXiv preprint arXiv:1904.12191*.

Hanin, B. and M. Nica (2019). Finite depth and width corrections to the neural tangent kernel. *arXiv preprint arXiv:1909.05989*.

Hayase, T. and R. Karakida (2020). The spectrum of fisher information of deep networks achieving dynamical isometry. *arXiv PrePrint 2006.07814*.

Hayou, S., A. Doucet, and J. Rousseau (2019). On the impact of the activation function on deep neural networks training. *ICML*.

Huang, J. and H. Yau (2020). Dynamics of deep neural networks and neural tangent hierarchy. *ICML*.

Huang, K., Y. Wang, M. Tao, and T. Zhao (2020). Why do deep residual networks generalize better than deep feedforward networks? - a neural tangent kernel perspective. *ArXiv preprint, arXiv:2002.06262*.

Jacot, A., F. Gabriel, and C. Hongler (2018). Neural tangent kernel: Convergence and generalization in neural networks. *32nd Conference on Neural Information Processing Systems*.

Karakida, R., S. Akaho, and S. Amari (2018). Universal statistics of Fisher information in deep neural networks: Mean field approach. *arXiv preprint arXiv:1806.01316*.

Lee, J., Y. Bahri, R. Novak, S. Schoenholz, J. Pennington, and J. Sohl-Dickstein (2018). Deep neural networks as Gaussian processes. *6th International Conference on Learning Representations*.

Lee, J., L. Xiao, S. Schoenholz, Y. Bahri, J. Sohl-Dickstein, and J. Pennington (2019). Wide neural networks of any depth evolve as linear models under gradient descent. *NeurIPS*.

Lillicrap, T., D. Cownden, D. Tweed, and C. Akerman (2016). Random synaptic feedback weights support error backpropagation for deep learning. *Nature Communications 7(13276)*.
MacRobert, T. (1967). *Spherical harmonics: An elementary treatise on harmonic functions, with applications*. Pergamon Press.

Matthews, A., J. Hron, M. Rowland, R. Turner, and Z. Ghahramani (2018). Gaussian process behaviour in wide deep neural networks. *6th International Conference on Learning Representations*.

Neal, R. (1995). Bayesian learning for neural networks. *Springer Science & Business Media 118*.

Nguyen, Q. and M. Hein (2018). Optimization landscape and expressivity of deep CNNs. *ICML*.

Novak, R., L. Xiao, J. Hron, J. Lee, A. A. Alemi, J. Sohl-Dickstein, and S. S. Schoenholz (2020). Neural tangents: Fast and easy infinite neural networks in python. In *International Conference on Learning Representations*.

Poole, B., S. Lahiri, M. Raghu, J. Sohl-Dickstein, and S. Ganguli (2016). Exponential expressivity in deep neural networks through transient chaos. *30th Conference on Neural Information Processing Systems*.

Schoenholz, S., J. Gilmer, S. Ganguli, and J. Sohl-Dickstein (2017). Deep information propagation. *5th International Conference on Learning Representations*.

Xiao, L., Y. Bahri, J. Sohl-Dickstein, S. S. Schoenholz, and P. Pennington (2018). Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks. *ICML 2018*.

Yang, G. (2019). Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. *arXiv preprint arXiv:1902.04760*.

Yang, G. (2020). Tensor programs iii: Neural matrix laws. *arXiv preprint arXiv:2009.10685*.

Yang, G. and S. Schoenholz (2017a). Mean field residual networks: On the edge of chaos. *Advances in Neural Information Processing Systems 30*, 2869–2869.

Yang, G. and S. Schoenholz (2017b). Mean field residual networks: On the edge of chaos. In *Advances in neural information processing systems*, pp. 7103–7114.

Zhang, C., S. Bengio, M. Hardt, B. Recht, and O. Vinyals (2017). Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*.

Zou, D., Y. Cao, D. Zhou, and Q. Gu (2018). Stochastic gradient descent optimizes over-parameterized deep ReLU networks. *arXiv preprint arXiv:1811.08888*. 