# Wide Deep Neural Networks


Soufiane Hayou

St John's College

University of Oxford

A thesis submitted for the degree of

*Doctor of Philosophy*

Hillary 2021

# Statement of Originality

I hereby declare that except where specific reference is made to the work of others, the intellectual contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification. My personal contributions are as detailed in the authorship forms at the end of each chapter. This dissertation is my own work except as specified in the text and authorship forms.

Soufiane Hayou
Hilary 2021

This thesis is dedicated to
my parents

# Acknowledgements

I want to start by thanking my supervisor, Arnaud Doucet, and my co-supervisor, Judith Rousseau, for their great support and patience, and for giving me the freedom to explore new ideas without hesitation. I believe the best supervisor is someone who can show the student a fertile ground and let them cultivate it; I have been lucky to have such supervisors, who, very early on, put me on the intellectual track that led to this thesis. Without their guidance, this thesis would not have been possible.

The department of statistics has been a great place for me to share knowledge and discuss new ideas with brilliant researchers. Thanks to Jean Francois Ton, Bobby He, Eugenio Clerico, Georges Deligiannidis, Yee Whye Teh, and to all the colleagues and friends with whom I shared the working space, and the colleagues from reading groups who made these experiences so rich and unique. I would like to thank the administrative and IT staff at the Oxford Statistics Department, who were always there to help whenever I needed their support.

I am grateful to the Engineering and Physical Sciences Research Council (EPSRC) and St John's College, Oxford, for providing me with the opportunity to pursue my DPhil at Oxford.

I would like to thank my family for their constant support and presence during these three years. Finally, I would like to thank Jihane, who made my time at Oxford unique and enjoyable.

# Abstract

Deep neural networks have had tremendous success in a wide range of applications where they achieve state of the art performance. Their success can be generally attributed to three main pillars: their natural back-propagation structure which allows time and resources efficient gradient computation; recent advances in optimization theory which have led to the development of fast training algorithms; and availability of computationally efficient software (neural networks frameworks such as PyTorch and Tensorflow), hardware (Graphics Processing Units(GPUs), and more recently Tensor Processing Units(TPUs)).

Deep neural networks are now the model of choice for many practitioners. As a result, there is a growing research interest in their theoretical properties. Classical results from approximation theory ensure that neural networks, even with a single hidden layer, are universal approximators, provided the model is big enough. From an optimization point of view, the loss surface of a deep neural network is generally highly non-convex, posing a big limitation on what results from Optimization theory can be applied to such models. Therefore, little is known about the local minimas, the saddle points, and the convergence of gradient based method (e.g. Stochastic Gradient Descent), with these models. Another interesting and understudied research topic is that of randomly initialized neural networks. Indeed, random neural networks provide a compelling framework that offers, in many cases, a simplified short-cut to understand theoretical properties of neural networks at initialization and Bayesian neural networks. Against this backdrop, the research presented in this thesis focuses on the theoretical properties of randomly initialized wide deep neural networks. It provides a comprehensive analysis of these models at initialization, leveraging a duality between random wide neural networks and Gaussian processes. Particularly, the research here presented pays careful attention to the role of the initialization hyperparameters, the activation function, and the neural architecture in the behaviour of these models. This level of depth allows for the derivation of principled guidelines for the training and designing of deep neural networks.

# Contents

# 1

# Introduction and Literature Review

## Contents



This dissertation follows an integrated thesis format and consists of a collection of four articles, one published in *Proceedings of the International Conference on Machine Learning (ICML) 2019*, one accepted in the *Proceedings of Artificial Intelligence and Statistics (AISTATS) 2021*, one accepted in the *Proceeding of the International Conference of Learning Representations (ICLR) 2021*, and one under revision to be submitted to *ICML 2021*. These articles are presented in chapters 2, 3, 4, and 5, respectively. All papers are self-contained; they each include a review of the literature, a bibliography, and an appendix. Although the articles are independent in format, they follow the same research direction, and form a coherent piece of work, where one article completes the other, and so forth.

This chapter is a general introduction to the topic of Randomly Initialized Wide Neural Networks. It outlines the theory and technical tools used in the articles; it provides a summary of the duality between Gaussian processes and wide neural networks, and introduces the Edge of Chaos initialization, the Neural Tangent Kernel, and Neural Networks Pruning, all of which are used in the rest of the dissertation. Finally, this chapter provides a brief description of the four chapters/articles mentioned earlier.

## 1.1 Neural Networks and Gaussian Processes

Deep neural networks have achieved state of the art performance on a variety of tasks including language processing and computer vision; see, e.g., [Goodfellow et al., 2016, Nguyen and Hein, 2018, Du et al., 2019, Zhang et al., 2016, Neyshabur et al., 2019]. The popularity of deep neural networks has motivated researchers to investigate their theoretical properties and, more importantly, their expressive power as the depth grows. To cite a few, [Montufar et al., 2014] have shown that neural networks have exponential expressive power with respect to the depth, while [Poole et al., 2016] obtained similar results using a topological measure of expressiveness.

Other works by [Neal, 1995, Lee et al., 2018a, Matthews et al., 2018] have studied neural networks at initialization and shown that they are closely related to Gaussian processes. The theoretical analysis of randomly initialized neural networks is motivated by many reasons. For instance, since neural networks training is generally a non-convex optimization problem, the choice of the initialization weights and the activation function will shape the parameter space that the optimization algorithm could explore. Moreover, the analysis of neural networks at initialization provides valuable insights on the output function, which helps understand what happens during the first steps of the optimization algorithm. Randomly initialized neural networks can also be seen as *priors* in the context of Bayesian neural networks. Knowing that the choice of the prior is crucial in Bayesian learning, studying the properties of random neural networks is essential for the design of a good prior.

### 1.1.1 Setup and Notation

Consider a Fully-connected Feedforward Neural Network (FFNN) of depth $L$, widths $(N_l)_{1\leq l\leq L}$, weights $W_{ij}^l$ and bias $B_i^l$. Given an input $x \in \mathbb{R}^d$, the propagation of this input through the network is given by

$$\begin{aligned} y_i^1(x) &= \sum_{j=1}^{d} W_{ij}^1 x_j + B_i^1, \\ y_i^l(x) &= \sum_{j=1}^{N_{l-1}} W_{ij}^l \phi(y_j^{l-1}(x)) + B_i^l, \quad \text{for } l \geq 2, \end{aligned} \tag{1.1}$$

where $\phi : \mathbb{R} \to \mathbb{R}$ is the activation function.

The first layer maps the input $x$ linearly to the neurons $(y_i^1(x))_{1\leq i\leq N_1}$. These linear *features*, also called pre-activations, are "activated" by applying $\phi$ element-wise, then passed as an input to the second layer. Repeating such transformations $L$ times defines an FFNN of depth $L$.

Table 1.1 highlights some popular choices of the activation function $\phi$, also called *non-linearity*. Figure 1.1 shows the plots of these activation functions.

**Table 1.1:** Popular activation functions with their derivatives

| **Activation function** | $\phi(x)$ | $\phi'(x)$ |
|---|---|---|
| ReLU (Rectified Linear Unit) | $x1_{x>0}$ | $1_{x>0}$ |
| ELU (Exponential Linear Unit) | $x1_{x\geq 0} + (e^x - 1)1_{x<0}$ | $1_{x\geq 0} + e^x 1_{x<0}$ |
| Sigmoid | $\frac{1}{1+e^{-x}}$ | $\frac{e^{-x}}{(1+e^{-x})^2}$ |
| Hyperbolic Tangent | $\frac{e^x - e^{-x}}{e^x + e^{-x}}$ | $\frac{4}{(e^x+e^{-x})^2}$ |

The FFNN defined above becomes closely related to Gaussian process when the width goes to infinity. The next section sheds light on this behaviour.

### 1.1.2 Gaussian Process Approximation of Neural Networks

The first link between neural networks and Gaussian processes can be traced back to [Neal, 1995], where the author demonstrated that a single layer neural network with weights initialized with a zero mean normal distribution, converges in distribution

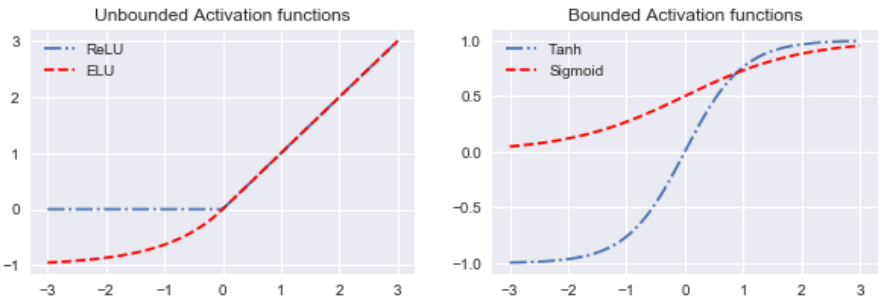



**Figure 1.1:** Plots of some activation functions

to a Gaussian process in the limit of infinite width (infinite number of neurons). Recently, [Lee et al., 2018a] generalized the result to multi-layer neural networks, and used the equivalent Gaussian process to perform bayesian inference for classification tasks; the authors consider the equivalent Gaussian process to be a *prior*, and use the mean function of the *posterior* Gaussian process[1] to compute predictions. [Matthews et al., 2018] provided a rigorous proof for the result in the multi-layer case, and [Yang, 2020] generalized the proofs to other architectures. This duality between Gaussian processes and neural networks has been leveraged in a stream of works ([Schoenholz et al., 2017, Yang and Schoenholz, 2017, Poole et al., 2016, Xiao et al., 2018]) to gain insights on the behaviour of deep neural networks at initialization in what is known as the *theory of information propagation*. The papers presented in this dissertation all make use of this theory. Unlike neural networks, Gaussian processes are usually more convenient to study since their properties are fully captured by their mean function and covariance kernel. In this sense, the Gaussian process equivalent to infinite width neural networks is a valuable 'shortcut' for researchers to (partially) understand the behaviour of overparameterized neural networks at initialization.

Recall the set of forward propagation equations for an FFNN (1.1)

$$y_i^1(x) = \sum_{j=1}^{d} W_{ij}^1 x_j + B_i^1,$$
$$y_i^l(x) = \sum_{j=1}^{N_{l-1}} W_{ij}^l \phi(y_j^{l-1}(x)) + B_i^l, \quad \text{for } l \geq 2.$$

[1] It is well known that with a Gaussian likelihood model, a Gaussian process prior leads to a posterior distribution that is also a Gaussian process

where $x \in \mathbb{R}^d$ is the input. Assume the weight and bias are randomly initialized with $W_{ij}^l \overset{iid}{\sim} \mathcal{N}(0, \frac{\sigma_w^2}{N_{l-1}})$ and $B_i^l \overset{iid}{\sim} \mathcal{N}(0, \sigma_b^2)$, where $\mathcal{N}(\mu, \sigma^2)$ denotes the normal distribution of mean $\mu$ and variance $\sigma^2$. The scaling factor $1/N_{l-1}$ in the variance of the weights is necessary to control the variance of the pre-activations. Given an input $x$, the pre-activations in the first layer $(y_j^1(x))_{1 \leq j \leq N_1}$ are *iid* random Gaussian variables with mean 0 and variance $q^1(x) = \frac{\sigma_w^2}{d}\|x\|^2 + \sigma_b^2$. More generally, it can be inferred from [Neal, 1995] that $(y_j^1(.))_{1 \leq j \leq N_1}$ are *iid* Gaussian processes with zero mean function and covariance kernel $q^1$ given by

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

[Lee et al., 2018a] extended this result to multiple layers in the limit of infinitely wide neural networks. Their proof uses an inductive argument: assume that $(y_j^{l-1}(.))_{1 \leq j \leq N_{l-1}}$ are *iid* Gaussian processes with zero mean function and covariance kernel $q^{l-1}$. Re-writing $W_{ij}^l = \frac{\sigma_w}{\sqrt{N_{l-1}}} Z_{ij}^l$, where $Z_{ij}^l \overset{iid}{\sim} \mathcal{N}(0, 1.)$, we have that

$$y_i^l(x) = \frac{\sigma_w}{\sqrt{N_{l-1}}} \sum_{j=1}^{N_{l-1}} Z_{ij}^l \phi(y_j^{l-1}(x)) + B_i^l, \quad \text{for } l \geq 2. \tag{1.2}$$

Using the Central Limit Theorem in the limit $N_{l-1} \to \infty$, it is straightforward that the processes $(y_j^l(.))_{1 \leq j \leq N_l}$ become *iid* Gaussian processes with zero mean function and covariance kernel given by

$$q^l(x, x') = \sigma_w^2 \mathbb{E}_{f \sim \mathcal{GP}(0, q^{l-1})}[\phi(f(x))\phi(f(x'))] + \sigma_b^2, \tag{1.3}$$

for all $x, x' \in \mathbb{R}^d$. The notation "$f \sim \mathcal{GP}(0, q^{l-1})$" means that $f$ is drawn from a Gaussian process with zero mean function and covariance kernel $q^{l-1}$.

The recursive argument above (which was used in [Lee et al., 2018a]) holds in the limit $N_1 \to \infty$, then $N_2 \to \infty, ...,$ then $N_L \to \infty$, i.e. the limit is taken recursively. This is an unrealistic scenario since popular neural networks architectures have layer widths that are usually similar; this implies that in order to use Gaussian processes as an approximation of multi-layer neural networks, layers widths should have the special hierarchy $N_{l-1} \gg N_l$. Fortunately, the proof can be modified to include the case where the widths grow at a similar rate. Indeed, [Matthews et al., 2018]

proved that Gaussian process behaviour persists in the limit of $N_1, ..., N_L \to \infty$ where the width $N_l$ can follow any increasing pattern. This result holds for all countable sets of the input space.

**Theorem 1 (Th 4. in [Matthews et al., 2018])** *Consider a randomly initialized deep neural network of the form* (1.1) *and assume that the activation function has linear growth. Let* $(N_l(n))_{1\leq l\leq L}$ *be the "widths functions", i.e. the widths parametrized by some number* $n \in \mathbb{N}$*. Then, for all increasing widths functions* $N_l(n)$*, and for any countable input set* $(x_i)_{i\geq 1}$*, the distribution of the output of the network converges in distribution to a Gaussian process as* $n \to \infty$*. The Gaussian process has mean function zero and covariance kernel given by the recursive formula* (1.3)*.*

In [Lee et al., 2018a], the authors named any Gaussian process with zero mean function and covariance kernel $q^l$ a Neural Network Gaussian Process (NNGP) of depth $l$ ([Lee et al., 2018a]); they proposed a Bayesian approach for learning with neural networks based on the NNGP where they use the mean function of the posterior Gaussian process to perform classification tasks.

The covariance kernel $q^l$ follows a recursive formula given by equation (1.3). Given two inputs $x, x'$, the value of the kernel at point $(x, x')$, given by $q^l(x, x')$, carries some 'information' about the inputs. We say that the sequence $(q^l(x, x'))_{1\leq l\leq L}$ represents the information propagation inside the network. Studying the asymptotic behaviour of $q^l(x, x')$ is crucial to gain insights on the output function of the infinite width neural network at initialization. Such analysis was carried out [Schoenholz et al., 2017]; the next section provides a summary of this topic.

## 1.2 Information Propagation and The Edge of Chaos

In [Schoenholz et al., 2017], the authors showed that, for a bounded activation function and general $(\sigma_b^2, \sigma_w^2)$, both the variance $q^l(x, x)$, resp. the correlation $c^l(x, x') = q^l(x, x')/\sqrt{q^l(x, x)}\sqrt{q^l(x', x')}$, converge exponentially fast to some limiting

value $q$, resp. $c$. In general, $q$ and $c$ depend only on $(\sigma_b^2, \sigma_w^2)$. To refine this convergence analysis, authors established the existence of $\epsilon_q$ and $\epsilon_c$ such that $|q^l(x, x') - q| \sim e^{-l/\epsilon_q}$ and $|c^l(x, x') - c| \sim e^{-l/\epsilon_c}$. The quantities $\epsilon_q$ and $\epsilon_c$ are called *depth scales* since they represent the range of depth to which the variance and correlation can propagate without being exponentially close to their corresponding limits. More precisely, if we write $\chi_1 = \sigma_w^2 \mathbb{E}[\phi'(\sqrt{q}Z)^2]$ and $\alpha = \chi_1 + \sigma_w^2 \mathbb{E}[\phi''(\sqrt{q}Z)\phi(\sqrt{q}Z)]$ then the depth scales are given by $\epsilon_q = -\log(\alpha)^{-1}$ and $\epsilon_c = -\log(\chi_1)^{-1}$. The condition $\chi_1 = 1$ corresponds to an infinite depth scale of the correlation. It is called the Edge of Chaos as it separates two phases (Ordered phase and Chaotic phase). The equation $\chi_1 = 1$ is usually seen as an equation with variables $(\sigma_b, \sigma_w)$. However, it can be generalized to a system of three variables $(\sigma_b, \sigma_w, \phi)$, where the third variable lives in function space (activation function). The boundedness condition on the activation function can be relaxed to include functions such as ReLU, ELU etc. We refer the reader to chapter 2 for more details.

**Definition 2 (Edge of Chaos, Informal)** *For $(\sigma_b, \sigma_w) \in (\mathbb{R}^+)^2$, let $q(\sigma_b, \sigma_w)$ be the limiting variance. The Edge of Chaos (EOC) is the set of values of $(\sigma_b, \sigma_w)$ satisfying $\chi_1 = \sigma_w^2 \mathbb{E}[\phi'(\sqrt{q(\sigma_b, \sigma_w)}Z)^2] = 1$.*

The euclidean quadrant $\{(\sigma_b, \sigma_w) \in \mathbb{R}^+\}$ is divided into three sets according to the value of $\chi_1$:

- **Ordered phase ($\chi_1 < 1$).** The correlation $c^l(x, x')$ converges exponentially fast to 1 for any inputs $x \neq x'$. This convergence is uniform over $\{(x, x') \in (\mathbb{R}^d)^2 : x \neq x'\}$. In this phase, if the variance terms $q^l(x, x)$ converges exponentially fast to some limiting value $q$ that is independent of $x$, then we can expect the output function of the neural network to be constant. The assumption of $q$ being independent of $x$ can be relaxed for ReLU activation function (see chapter 2 for more details).

- **Chaotic phase ($\chi_1 > 1$).** In this regime, the correlation $c^l(x, x')$ converges exponentially fast to some limiting value $c < 1$ independent of $x$. The

convergence is uniform over the set $\{(x, x') \in (\mathbb{R}^d)^2 : |1 - c^1(x, x')| > \epsilon\}$ for any $\epsilon \in (0, 1)$. Interestingly, this implies that very close inputs (in terms of correlation) lead to very different outputs. Thus, in the chaotic phase, in the limit of infinite width and depth, the output function of the neural network is non-continuous almost everywhere.

- **The Edge of Chaos ($\chi_1 = 1$).** In this regime, the correlation $c^l(x, x')$ converges to 1 at a sub-exponential rate. The convergence is uniform over the set $\{(x, x') \in (\mathbb{R}^d)^2 : |1 - c^1(x, x')| > \epsilon\}$ for any $\epsilon \in (0, 1)$. Contrary to the previous two phases where the information vanishes at an exponential rate, the sub-exponential convergence rate of the correlation on the Edge of Chaos preserves the information carried by the correlation as it "travels" inside the network. Eventually, this information will be lost since $c^l(x, x')$ converges to 1. However, it is expected that this will happen at a much deeper layer compared to the Ordered phase. Thus, one can say that the Edge of Chaos allows deeper information propagation.

Figure 1.2 illustrates the output function of an FFNN with Hyperbolic Tangent activation function for three different values of initialization hyperparameters $(\sigma_b, \sigma_w)$. The graphs show the values of the output function for $x \in [0, 1]^2$ (the notation $[0, 1]$ means that 0 and 1 are included in the interval). The output function in the Ordered phase is almost constant, while it is very 'noisy' in the Chaotic phase. On the other hand, the output function on the Edge of Chaos is smooth and varies significantly across the input space $[0, 1]^2$. In chapter 2, we give a comprehensive analysis of the Edge of Chaos and we characterize the convergence of the correlation in this regime. We particularly show that using smooth activation functions yields a convergence rate of $\mathcal{O}(l^{-1})$ for the correlation compared to $\mathcal{O}(l^{-2})$ with ReLU activation function.

The theory of the Edge of Chaos deals with information propagation at initialization. Although it gives interesting insights about the network output at initialization and how the Edge of Chaos initialization maximizes information flow

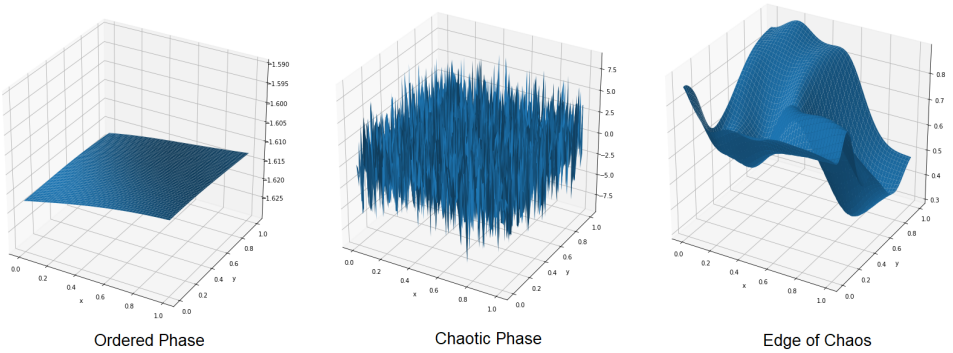



**Figure 1.2:** Output of a 300x20 Fully-Connected FeedForward Neural Network with Hyperbolic Tangent activation function for three initialization phases.

inside the network, it does not cover the training part. It turns out that training wide neural networks is related to the so-called Neural Tangent Kernel; further details on this are discussed in the section below.

## 1.3 Neural Tangent Kernel for Deep Neural Networks

Recent work by [Jacot et al., 2018] has shown that training a neural network of any kind with gradient descent in the parameter space is related to kernel gradient descent in function space with respect to the Neural Tangent Kernel (NTK). In the limit of infinite width, the Neural Tangent Kernel remains constant during training and is fully characterized by its value at initialization. Hence, in the *NTK regime* (infinite width), the training procedure is fully captured at initialization. It is therefore natural to bridge the gap between the initialization and the Neural Tangent Kernel.

### 1.3.1 Neural Tangent Kernel

In the previous section, the FFNN was presented as a simple neural network model. More generally, a neural network of depth $L \geq 1$ is given by a set of forward propagation equations

$$y^l(x) = \mathcal{F}_l(\theta_l, y^{l-1}(x)), \quad 1 \leq l \leq L, \tag{1.4}$$

where $x \in \mathbb{R}^d$ is the input, $\theta_l$ are the parameters of the $l^{th}$ layer (for FFNN, $\theta_l = (W^l, B^l)$), and $\mathcal{F}_l$ is a mapping that defines the nature of the layer. Popular choices include dense and convolutional mappings.

Now, consider a general neural network model of type (1.4) consisting of $L$ layers $(y^l)_{1\leq l\leq L}$. 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 $\theta$. In the case of an FFNN with layer widths $(N_l)_{1\leq l\leq L}$, $\theta$ has dimension $p = \sum_{l=1}^{L} N_l \times (N_{l-1} + 1)$ with the notation $N_0 = d$. In general, the output function $f$ of the neural network is given by some mapping $s : \mathbb{R}^{N_L} \to \mathbb{R}^o$ of the last layer $y^L(x)$; $o$ being the dimension of the output. For any input $x \in \mathbb{R}^d$, we thus have $f(x, \theta) = s(y^L(x)) \in \mathbb{R}^o$. For a classification problem with $k$ classes, practitioners usually choose $s$ to be the Softmax[2] function and, in this case, we have $o = k$.

As the model is trained, the parameters are updated with each gradient step, and we shall call $\theta_t$ the value of $\theta$ at training time $t$ and $f_t(x) = f(x, \theta_t) = (f_j(x, \theta_t))_{1\leq j\leq o}$. Let $D = (x_i, z_i)_{1\leq i\leq M}$ be the dataset and let $\mathcal{X} = (x_i)_{1\leq i\leq M}$, $\mathcal{Z} = (z_j)_{1\leq j\leq M}$ be the matrices of inputs and outputs respectively, with dimension $d \times M$ and $o \times M$. For any function $g : \mathbb{R}^{d\times o} \to \mathbb{R}^k$, $k \geq 1$, we denote by $g(\mathcal{X}, \mathcal{Z})$ the matrix $(g(x_i, z_i))_{1\leq i\leq M}$ of dimension $k \times M$.

[Jacot et al., 2018] studied the behaviour of the network output as a function of the training time $t$ when the network is trained using gradient descent. With parameters $\theta \in \mathbb{R}^p$, the empirical loss function is given by

$$\mathcal{L}(\theta) = \frac{1}{M} \sum_{i=1}^{M} \ell(f(x_i, \theta), z_i),$$

where $\ell : (z', z) \to \ell(z', z)$ is some loss criterion, e.g. quadratic loss, cross entropy etc.

The "full-batch" gradient descent algorithm (as opposed to "mini-batch" gradient descent or stochastic gradient descent) follows the update rule given by

$$\hat{\theta}_{t+1} = \hat{\theta}_t - \eta \nabla_\theta \mathcal{L}(\hat{\theta}_t), \tag{1.5}$$

[2]Given $v = (v_i)_{1\leq i\leq k} \in \mathbb{R}^k$, the Softmax function maps $v$ to $\text{Softmax}(v) = \left(\frac{e^{v_i}}{\sum_{j=1}^{k} e^{v_j}}\right)_{1\leq i\leq k}$.

where $\eta > 0$ is the learning rate.

Assume that we want to train the model with a maximal number of steps $N_s$ and let $T = \eta N_s$. This update rule can be seen as a discretization of a continuous time system known as the "gradient flow", and is given by

$$d\theta_t = -\nabla_\theta \mathcal{L}(\theta_t)dt, \tag{1.6}$$

where $\Delta t = \eta$ is the discretization step. It is well known that this discretization scheme induces an error bound of order $\mathcal{O}(\eta)$ under minimal conditions on the loss function (see Appendix of chapter 3). The training time $T$ with gradient flow represents the number of steps for the discrete system. The Ordinary Differential Equation (ODE) (1.6) can be re-written as

$$d\theta_t = -\frac{1}{M}\nabla_\theta f(\mathcal{X}, \theta_t)^T \nabla_{z'} \ell(f(\mathcal{X}, \theta_t), \mathcal{Z})dt,$$

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

$$df_t(x) = -\frac{1}{N}\nabla_\theta f(x, \theta_t)\nabla_\theta f(\mathcal{X}, \theta_t)^T \nabla_{z'} \ell(f_t(\mathcal{X}), \mathcal{Z})dt. \tag{1.7}$$

The Neural Tangent Kernel (NTK) $K_\theta^L$ is defined as the $o \times o$ dimensional kernel

$$\begin{aligned} K_{\theta_t}^L(x, x') &= \nabla_\theta f(x, \theta_t)\nabla_\theta f(x', \theta_t)^T \in \mathbb{R}^{o \times o} \\ &= \sum_{l=1}^{L} \nabla_{\theta^l} f(x, \theta_t)\nabla_{\theta^l} f(x', \theta_t)^T, \end{aligned} \tag{1.8}$$

where $x, x' \in \mathbb{R}^d$. We also define the *matrix NTK* $K_{\theta_t}^L(\mathcal{X}, \mathcal{X})$ as the $oM \times oM$ matrix defined blockwise by

$$K_{\theta_t}^L(\mathcal{X}, \mathcal{X}) = \begin{pmatrix} K_{\theta_t}^L(x_1, x_1) & \cdots & K_{\theta_t}^L(x_1, x_M) \\ K_{\theta_t}^L(x_2, x_1) & \cdots & K_{\theta_t}^L(x_2, x_M) \\ \vdots & \ddots & \vdots \\ K_{\theta_t}^L(x_M, x_1) & \cdots & K_{\theta_t}^L(x_N, x_M) \end{pmatrix}.$$

By applying (1.7) to the vector $\mathcal{X}$, we obtain

$$df_t(\mathcal{X}) = -\frac{1}{M}K_{\theta_t}^L(\mathcal{X}, \mathcal{X})\nabla_{z'} \ell(f_t(\mathcal{X}), \mathcal{Z})dt. \tag{1.9}$$

The ODE (1.9) characterizes the neural network training with gradient flow. In the case of quadratic loss, this ODE becomes tractable and admits a closed-form solution. This is discussed in the subsequent section.

### 1.3.2 The NTK regime

In the case of an FFNN with NTK parameterization (see chapter 3 for more details), [Jacot et al., 2018] proved that, with gradient flow dynamics, in the limit $N_1, N_2, ..., N_L \to \infty$ recursively, the kernel $K^L_{\theta_t}$ converges to a kernel $K^L$ which depends only on $L$ (number of layers) for all $t < T$, under the technical assumption that $\int_0^T \|\nabla_z \ell(f_t(\mathcal{X}, \mathcal{Z}))\| dt$ is finite almost surely with respect to the initialization weights. [Yang, 2020] generalized the result to the case where $\min\{N_1, N_2, ..., N_L\} \to \infty$. Let $f^\infty$ be the output of the network in this limit. We then have

$$df_t^\infty(\mathcal{X}) = -\frac{1}{M} K^L(\mathcal{X}, \mathcal{X}) \nabla_{z'} \ell(f_t^\infty(\mathcal{X}), \mathcal{Z}) dt. \tag{1.10}$$

Letting, $\hat{K}^L = K^L(\mathcal{X}, \mathcal{Z})$, and considering the quadratic loss $\ell(z', z) = \frac{1}{2}\|z' - z\|^2$, equation (1.10) is equivalent to

$$df_t^\infty(\mathcal{X}) = -\frac{1}{M} \hat{K}^L (f_t^\infty(\mathcal{X}) - \mathcal{Z}) dt. \tag{1.11}$$

The ODE (1.11) is characteristic of a simple linear model that has a closed-form solution given by

$$f_t^\infty(\mathcal{X}) = e^{-\frac{1}{M}\hat{K}^L t} f_0^\infty(\mathcal{X}) + (I - e^{-\frac{1}{M}\hat{K}^L t})\mathcal{Z}. \tag{1.12}$$

For general input $x \in \mathbb{R}^d$, we have

$$f_t^\infty(x) = f_0^\infty(x) + \gamma(x, \mathcal{X})(I - e^{-\frac{1}{M}\hat{K}^L t})(\mathcal{Z} - f_0^\infty(\mathcal{X})), \tag{1.13}$$

where $\gamma(x) = K^L(x, \mathcal{X}) K^L(\mathcal{X}, \mathcal{X})^{-1}$.

The linear model solution $f_t^\infty$ is an approximation at time $t$ of the neural network trained with gradient flow, in the ideal scenario of infinite width. In general, the kernel matrix $\hat{K}^L$ is invertible, allowing for the derivation of the infinite training time solution $f_\infty^\infty$ given by

$$f_\infty^\infty(x) = f_0^\infty(x) + \gamma(x, \mathcal{X})(\mathcal{Z} - f_0^\infty(\mathcal{X})). \tag{1.14}$$

$f_\infty^\infty$ is usually called the "NTK regime," the "NTK regime solution," or the "linear regime solution," and can be seen as an approximation of the trained neural network *without training*. [Lee et al., 2019] showed that, surprisingly, this linear model achieves high accuracy that might even compete with the original trained neural network. However, while it yields good performance for shallow neural networks, chapter 3 shows the limitations of this regime for deep neural networks.

Although modern deep neural networks have achieved state of the art performance in many tasks, these models are overparameterized; they usually have millions, if not billions, of parameters, making them impossible to implement on limited capacity devices such as mobile phones and tablets. A natural question that arises is whether these models can be *compressed* while preserving their performance. This topic has been the subject of a line of research known as *network pruning* (the name finds its origins in the similarity with trees pruning). Further discussion on this topic is provided in the next section.

## 1.4 Neural Networks Pruning

Neural Network Pruning is widely used to reduce the time and space requirements both at training and test time. The concept is to identify weights that do not contribute significantly to the network performance and remove them. These weights are usually chosen based on some criterion, e.g. weights with smallest magnitude or gradient norm. Consider a general neural network model of depth $L$ given by (1.4)

$$y^l(x) = \mathcal{F}_l(\theta_l, y^{l-1}(x)), \quad 1 \leq l \leq L,$$

where $x \in \mathbb{R}^d$ is the input. Network pruning consists of finding a binary mask $\delta$ that has the same dimension of the weights. By applying the mask $\delta$ to the weights element-wise, we obtain the *pruned network*

$$\bar{y}^l(x) = \mathcal{F}_l(\delta_l \circ \theta_l, y^{l-1}(x)), \quad 1 \leq l \leq L, \tag{1.15}$$

where $\circ$ is the Hadamard product (element-wise product).
Most pruning procedures currently available are applied after training the full neural network [LeCun et al., 1990, Hassibi et al., 1993, Mozer and Smolensky, 1989, Dong et al., 2017]. They follow the standard pruning procedure

$$\text{train} \rightarrow \text{prune} \rightarrow .... \rightarrow \text{prune} \rightarrow \text{train},$$

until the desired sparsity is achieved. For large datasets such as ImageNet[3], these procedures are slow and require extensive computational resources. This has led to the development of methods that consider pruning the neural network during training. For example, [Louizos et al., 2018] propose an algorithm which adds a $L_0$ regularization on the weights to enforce sparsity. Similarly, [Carreira-Perpiñán and Idelbayev, 2018, Alvarez and Salzmann, 2017] propose the inclusion of compression inside training steps. Other pruning variants consider training a secondary network that learns a pruning mask for a given architecture ([Li et al., 2020, Liu et al., 2019]).

Recently, [Frankle and Carbin, 2019] have introduced, and experimentally validated, the *Lottery Ticket Hypothesis* which conjectures the existence of a sparse subnetwork that achieves similar performance to the original neural network. These empirical findings have motivated the development of *pruning at initialization* methods, such as SNIP ([Lee et al., 2018b]), which demonstrated similar performance to classical pruning methods of pruning-after-training. Pruning at initialization methods consider pruning the randomly initialized network and training the sparse network. They never require training the complete neural network and are thus more memory efficient, allowing to train deep networks using limited computational resources. However, such techniques may suffer from different problems. In particular, nothing prevents such methods from pruning one whole layer of the network, a phenomenon also known as *layer-collapse*, which would typically make the network untrainable (residual neural networks are an exception since the residual connection keeps the information flow even if one layer is fully

[3]Available at `http://www.image-net.org/`

pruned). More generally, it is usually difficult to train the resulting pruned neural network [Li et al., 2018] even if no layer was fully pruned. To resolve this situation, [Lee et al., 2020]) attempted to enforce the dynamical isometry using orthogonal weights, while [Wang et al., 2020] (GraSP) used Hessian-based pruning to preserve gradient flow. Another work by [Tanaka et al., 2020] considers a data-agnostic iterative approach using the concept of synaptic flow in order to avoid layer-collapse.

For pruning at initialization, the binary mask $\delta$ is computed based on some criterion. Popular choices include:

- **Magnitude based pruning (MBP)**: the weights are pruned based on the magnitude $|W|$ (weights with small magnitude are pruned).

- **Sensitivity based pruning (SBP)**: the weights are pruned based on the values of $|W\frac{\partial \mathcal{L}}{\partial W}|$, where $\mathcal{L}$ is the loss function. This is motivated by the first order Taylor expansion of the loss function

$$\mathcal{L}_W \approx \mathcal{L}_{W=0} + W\frac{\partial \mathcal{L}}{\partial W}.$$

  This criterion is used by SNIP ([Lee et al., 2018b]).

- **Hessian based pruning (HBP)**: the weights are pruned based on some function that uses the Hessian of the loss function as in GraSP [Wang et al., 2020].

Pruning a neural network at initialization is governed by the values of the criterion at initialization. Therefore, understanding how the initialization impacts the criterion is valuable in the choice of hyperparameters. For deep neural networks, it turns out that the theory of information propagation provides valuable tools to derive principled guidelines for the choice of the network hyperparameters and the architecture design. This is the main topic of chapter 5.

## 1.5 Thesis outline

As aforementioned, this dissertation follows an integrated thesis format, and is composed of four articles in the theory of randomly initialized wide deep neural networks. These articles form a coherent piece of work and complete each other. This section provides a short overview of the articles (chapters), and delve into the specificities of each one. Lastly, as per the submission guidelines, following each article is a signed statement of authorship that details my contributions.

**On the Impact of the Initialization and the Activation function on Deep Neural Networks.** Chapter 2 is dedicated to the theoretical understanding of the Edge of Chaos. We provide a comprehensive analysis of the Edge of Chaos for different activation functions. In it, we particularly show that by choosing smooth activation functions such as Hyperbolic Tangent and ELU (see figure 1.1), the convergence rate of the correlation improves from $\mathcal{O}(l^{-2})$ (the convergence rate with ReLU) to $\mathcal{O}(l^{-1})$, meaning that the information can propagate deeper inside the network, thus allowing training deeper neural networks. We illustrate all theoretical results with extensive empirical results.

The article is referenced as:

- **Hayou, S.**, Doucet, A., and Rousseau, J. (2019). On the Impact of the Activation Function on Deep Neural Networks Training. *Proceedings of the 36th International Conference of Machine Learning (ICML 2020).*

**Neural Tangent Kernel for Deep Neural Networks.** The Neural Tangent Kernel introduced by [Jacot et al., 2018] is closely related to deep neural networks training. Indeed, authors have 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, many recent works (including our work [Hayou et al., 2019]) have

demonstrated the benefit of the Edge of Chaos initialization on the performance of deep neural networks. In chapter 3, we bridge the gap between these two concepts by quantifying the impact of the initialization and the activation function on the Neural Tangent Kernel 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, and we provide experiments illustrating our theoretical results.

The paper is under revision and will be re-submitted to the International Conference of Machine Learning (ICML 2021). It is referenced as:

- **Hayou, S.**, Doucet, A., and Rousseau, J. (2020). Mean-field Behaviour of Neural Tangent Kernel for Deep Neural Networks. *arXiv preprint arXiv:1905.13654v8.*

**Stable Residual Neural Networks.** Residual neural networks, also called ResNet, are neural network architectures given by the set of forward propagation equations

$$y^l(x) = y^{l-1}(x) + \mathcal{F}_l(\theta_l, y^{l-1}(x)), \quad 1 \leq l \leq L, \tag{1.16}$$

where $x$ is the input, and $\mathcal{F}_l$ is a mapping that defines the nature of the layer; it is usually a convolutional mapping.

While ResNet solve the problem of gradient vanishing, they might suffer from gradient exploding as the depth becomes large as it has been shown in [Yang and Schoenholz, 2017]. Moreover, recent results have shown that ResNet might lose expressivity as the depth goes to infinity [Yang and Schoenholz, 2017, Hayou et al., 2019]. To resolve these issues, we introduce in chapter 4 a new class of ResNet architectures, called Stable ResNet, that have the property of stabilizing the gradient while ensuring expressivity in the infinite depth limit.

The article is referenced as:

- **Hayou, S.**, Clerico, E., He, B., Deligiannidis, G., Doucet, A., and Rousseau, J. (2020). Stable ResNet. *To appear in the Proceedings of the* $24^{th}$ *International Conference on Artificial Intelligence and Statistics (AISTATS 2021).*

**Neural Networks Pruning at Initialization.** Modern day neural networks are usually overparameterized; they display state of the art performance while having millions, if not billions, of trainable parameters. However, there is a growing need for smaller neural networks to be able to use machine learning applications on devices with limited computational resources. A popular approach consists of using pruning techniques. These techniques have traditionally focused on pruning pre-trained neural networks [LeCun et al., 1990, Hassibi et al., 1993], and follow an iterative 'train then prune' scheme. Such procedures are memory heavy and could require days, if not months, of training on a single GPU device to find a winning 'pruned' network. Recent work by [Lee et al., 2018b] has shown promising results when pruning at initialization, meaning that we prune the untrained neural network once at initialization to achieve the desired sparsity, then train the sparse network. However, for deep neural networks, such procedures remain unsatisfactory as the resulting pruned networks can be difficult to train and, for instance, they do not prevent one layer from being fully pruned (layer-collapse). In chapter 5, we provide a comprehensive theoretical analysis of magnitude- and gradient-based pruning at initialization and training of sparse architectures. We propose novel principled approaches that resolve the layer-collapse issue, and we validate experimentally on a variety of neural network architectures.

The article is referenced as

- **Hayou, S.**, Ton, J.F., Doucet, A., Teh, Y.W. (2020). Robust Pruning at Initialization. *To appear in the Proceedings of the* $9^{th}$ *International Conference on Learning Representations (ICLR 2021).*

# 2

# On the Impact of the Initialization and the Activation function on Deep Neural Networks

# On the Impact of the Activation Function on Deep Neural Networks Training

**Soufiane Hayou** [1] **Arnaud Doucet** [1] **Judith Rousseau** [1]

## Abstract

The weight initialization and the activation function of deep neural networks have a crucial impact on the performance of the training procedure. An inappropriate selection can lead to the loss of information of the input during forward propagation and the exponential vanishing/exploding of gradients during back-propagation. Understanding the theoretical properties of untrained random networks is key to identifying which deep networks may be trained successfully as recently demonstrated by (Schoenholz et al., 2017) who showed that for deep feedforward neural networks only a specific choice of hyperparameters known as the 'Edge of Chaos' can lead to good performance. While the work by (Schoenholz et al., 2017) discuss trainability issues, we focus here on training acceleration and overall performance. We give a comprehensive theoretical analysis of the Edge of Chaos and show that we can indeed tune the initialization parameters and the activation function in order to accelerate the training and improve performance.

## 1. Introduction

Deep neural networks have become extremely popular as they achieve state-of-the-art performance on a variety of important applications including language processing and computer vision; see, e.g., (Goodfellow et al., 2016). The success of these models has motivated the use of increasingly deep networks and stimulated a large body of work to understand their theoretical properties. It is impossible to provide here a comprehensive summary of the large number of contributions within this field. To cite a few results relevant to our contributions, (Montufar et al., 2014) have shown that neural networks have exponential expressive power with respect to the depth while (Poole et al., 2016) obtained similar results using a topological measure of expressiveness.

Since the training of deep neural networks is a non-convex optimization problem, the weight initialization and the activation function will essentially determine the functional subspace that the optimization algorithm will explore. We follow here the approach of (Poole et al., 2016) and (Schoenholz et al., 2017) by investigating the behaviour of random networks in the infinite-width and finite-variance i.i.d. weights context where they can be approximated by a Gaussian process as established by (Neal, 1995), (Matthews et al., 2018) and (Lee et al., 2018).

In this paper, we focus on the so-called Edge of Chaos ( introduced by (Poole et al., 2016)). Our contribution is threefold. Firstly, we provide a comprehensive analysis of the so-called Edge of Chaos (EOC) curve and show that initializing a network on this curve leads to a deeper propagation of the information through the network and accelerates the training. In particular, we show that a feedforward ReLU network initialized on the EOC acts as a simple residual ReLU network in terms of information propagation. Secondly, we introduce a class of smooth activation functions which allow for deeper signal propagation (Proposition 3) than ReLU. In particular, this analysis sheds light on why smooth versions of ReLU (such as SiLU or ELU) perform better experimentally for deep neural networks; see, e.g., (Clevert et al., 2016), (Pedamonti, 2018), (Ramachandran et al., 2017) and (Milletarí et al., 2018). Lastly, we show the existence of optimal points on the EOC curve and we provide guidelines for the choice of such point and we demonstrate numerically the consistence of this approach. We also complement previous empirical results by illustrating the benefits of an initialization on the EOC in this context. All proofs are given in the Supplementary Material.

## 2. On Gaussian process approximations of neural networks and their stability

### 2.1. Setup and notations

We use similar notations to those of (Poole et al., 2016) and (Lee et al., 2018). Consider a fully connected feedforward random neural network of depth $L$, widths $(N_l)_{1\leq l\leq L}$, weights $W_{ij}^l \overset{iid}{\sim} \mathcal{N}(0, \frac{\sigma_w^2}{N_{l-1}})$ and bias $B_i^l \overset{iid}{\sim} \mathcal{N}(0, \sigma_b^2)$,

[1]Department of Statistics, University of Oxford, Oxford, United Kingdom. Correspondence to: Soufiane Hayou <soufiane.hayou@stats.ox.ac.uk>.

*Proceedings of the $36^{th}$ International Conference on Machine Learning*, Long Beach, California, PMLR 97, 2019. 

where $\mathcal{N}(\mu, \sigma^2)$ denotes the normal distribution of mean $\mu$ and variance $\sigma^2$. For some input $a \in \mathbb{R}^d$, the propagation of this input through the network is given for an activation function $\phi : \mathbb{R} \to \mathbb{R}$ by

$$y_i^1(a) = \sum_{j=1}^{d} W_{ij}^1 a_j + B_i^1, \tag{1}$$

$$y_i^l(a) = \sum_{j=1}^{N_{l-1}} W_{ij}^l \phi(y_j^{l-1}(a)) + B_i^l, \quad \text{for } l \geq 2. \tag{2}$$

Throughout this paper we assume that for all $l$ the processes $y_i^l(.)$ are independent (across $i$) centred Gaussian processes with covariance kernels $\kappa^l$ and write accordingly $y_i^l \overset{ind}{\sim} \mathcal{GP}(0, \kappa^l)$. This is an idealized version of the true processes corresponding to choosing $N_{l-1} = +\infty$ (which implies, using Central Limit Theorem, that $y_i^l(a)$ is a Gaussian variable for any input $a$). The approximation of $y_i^l(.)$ by a Gaussian process was first proposed by (Neal, 1995) in the single layer case and has been recently extended to the multiple layer case by (Lee et al., 2018) and (Matthews et al., 2018). The multiplayer kernel can be obtained either by taking the limits $N_l \to \infty$ sequentially or simultaneously. In both cases, the limiting kernel is the same. We recall here the expressions of the limiting Gaussian process kernels. For any input $a \in \mathbb{R}^d$, $\mathbb{E}[y_i^l(a)] = 0$ so that for any inputs $a, b \in \mathbb{R}^d$

$$\begin{aligned}\kappa^l(a,b) &= \mathbb{E}[y_i^l(a) y_i^l(b)] \\ &= \sigma_b^2 + \sigma_w^2 \mathbb{E}[\phi(y_i^{l-1}(a))\phi(y_i^{l-1}(b))] \\ &= \sigma_b^2 + \sigma_w^2 F_\phi(\kappa^{l-1}(a,a), \kappa^{l-1}(a,b), \kappa^{l-1}(b,b)),\end{aligned}$$

where $F_\phi$ is a function that only depends on $\phi$. This gives a recursion to calculate the kernel $\kappa^l$; see, e.g., (Lee et al., 2018) for more details. We can also express the kernel $\kappa^l(a,b)$ (which we denote hereafter by $q_{ab}^l$) in terms of the correlation $c_{ab}^l$ in the $l^{th}$ layer

$$q_{ab}^l = \sigma_b^2 + \sigma_w^2 \mathbb{E}[\phi(\sqrt{q_a^{l-1}} Z_1)\phi(\sqrt{q_b^{l-1}} U_2(c_{ab}^{l-1}))],$$

where $q_a^{l-1} := q_{aa}^{l-1}$, resp. $c_{ab}^{l-1} := q_{ab}^{l-1} / \sqrt{q_a^{l-1} q_b^{l-1}}$, is the variance, resp. correlation, in the $(l-1)^{th}$ layer and $U_2(x) = xZ_1 + \sqrt{1-x^2} Z_2$ where $Z_1, Z_2$ are independent standard Gaussian random variables. When it propagates through the network. $q_a^l$ is updated through the layers by the recursive formula $q_a^l = F(q_a^{l-1})$, where $F$ is the 'variance function' given by

$$F(x) = \sigma_b^2 + \sigma_w^2 \mathbb{E}[\phi(\sqrt{x} Z)^2], \qquad Z \sim \mathcal{N}(0,1). \tag{3}$$

Throughout this paper, $Z, Z_1, Z_2$ will always denote independent standard Gaussian variables, and $a, b$ two inputs for the network.

Before starting our analysis, we define the transform $V$ for a function $\phi$ defined on $\mathbb{R}$ by $V[\phi](x) = \sigma_w^2 \mathbb{E}[\phi(\sqrt{x}Z)^2]$ for $x \geq 0$. We have $F = \sigma_b^2 + V[\phi]$.

Let $E$ and $G$ be two subsets of $\mathbb{R}$. We define the following sets of functions for $k \in \mathbb{N}$ by

$$\begin{aligned}\mathcal{D}^k(E,G) &= \{f : E \to G \text{ such that } f^{(k)} \text{ exists}\} \\ \mathcal{C}^k(E,G) &= \{f \in \mathcal{D}^k(E,G) \text{ such that } f^{(k)} \text{ is continuous}\} \\ \mathcal{D}_g^k(E,G) &= \{f \in \mathcal{D}^k(E,G) : \forall j \leq k, \mathbb{E}[f^{(j)}(Z)^2] < \infty\} \\ \mathcal{C}_g^k(E,G) &= \{f \in \mathcal{C}^k(E,G) : \forall j \leq k, \mathbb{E}[f^{(j)}(Z)^2] < \infty\}\end{aligned}$$

where $f^{(k)}$ is the $k^{\text{th}}$ derivative of $f$. When $E$ and $G$ are not explicitly mentioned, we assume $E = G = \mathbb{R}$.

### 2.2. Limiting behaviour of the variance and covariance operators

We analyze here the limiting behaviour of $q_a^l$ and $c_{a,b}^l$ as $l$ goes to infinity. From now onwards, we will also assume without loss of generality that $c_{ab}^1 \geq 0$ (similar results can be obtained straightforwardly when $c_{ab}^1 \leq 0$). We first need to define the *Domains of Convergence* associated with an activation function $\phi$.

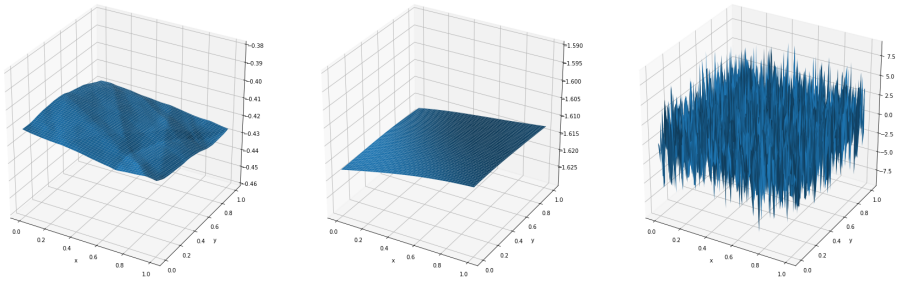
**Definition 1.** *Let $\phi \in \mathcal{D}_g^0$, $(\sigma_b, \sigma_w) \in (\mathbb{R}^+)^2$.*
*(i) Domain of convergence for the variance $D_{\phi,var}$ : $(\sigma_b, \sigma_w) \in D_{\phi,var}$ if there exists $K > 0$, $q \geq 0$ such that for any input $a$ with $q_a^1 \leq K$, $\lim_{l\to\infty} q_a^l = q$. We denote by $K_{\phi,var}(\sigma_b, \sigma_w)$ the maximal $K$ satisfying this condition.*
*(ii) Domain of convergence for the correlation $D_{\phi,corr}$: $(\sigma_b, \sigma_w) \in D_{\phi,corr}$ if there exists $K > 0$ such that for any two inputs $a, b$ with $q_a^1, q_b^1 \leq K$, $\lim_{l\to\infty} c_{ab}^l = 1$. We denote by $K_{\phi,corr}(\sigma_b, \sigma_w)$ the maximal $K$ satisfying this condition.*

*Remark*: Typically, $q$ in Definition 1 is a fixed point of the variance function defined in (3). Therefore, it is easy to see that for any $(\sigma_b, \sigma_w)$ such that $F$ is non-decreasing and admits at least one fixed point, we have $K_{\phi,var}(\sigma_b, \sigma_w) \geq q$ where $q$ is the minimal fixed point; i.e. $q := \min\{x : F(x) = x\}$. Thus, if we re-scale the input data to have $q_a^1 \leq q$, the variance $q_a^l$ converges to $q$. We can also re-scale the variance $\sigma_w$ of the first layer (only) to assume that $q_a^1 \leq q$ for all inputs $a$.

The next Lemma gives sufficient conditions under which $K_{\phi,var}$ and $K_{\phi,corr}$ are infinite.

**Lemma 1.** *Assume $\phi''$ exists at least in the distribution sense.*[1]
*Let $M_\phi := \sup_{x\geq 0} \mathbb{E}[|\phi'^2(xZ) + \phi''(xZ)\phi(xZ)|]$. Assume*

[1] ReLU admits a Dirac mass in 0 as second derivative and so is covered by our developments.



(a) ReLU with $(\sigma_b, \sigma_w) = (1, 1)$ (b) Tanh with $(\sigma_b, \sigma_w) = (1, 1)$ (c) Tanh with $(\sigma_b, \sigma_w) = (0.3, 2)$

*Figure 1.* Draws of outputs for ReLU and Tanh networks for different parameters $(\sigma_b, \sigma_w)$. Figures (a) and (b) show the effect of an initialization in the ordered phase, the outputs are nearly constant. Figure (c) shows the effect of an initialization in the chaotic phase.

*$M_\phi < \infty$, then for $\sigma_w^2 < \frac{1}{M_\phi}$ and $\sigma_b \geq 0$, we have $(\sigma_b, \sigma_w) \in D_{\phi,var}$ and $K_{\phi,var}(\sigma_b, \sigma_w) = \infty$.*
*Let $C_{\phi,\delta} := \sup_{x,y\geq 0,|x-y|\leq\delta,c\in[0,1]} \mathbb{E}[|\phi'(xZ_1)\phi'(y(cZ_1 + \sqrt{1-c^2}Z_2)|]$. Assume $C_{\phi,\delta} < \infty$ for some $\delta > 0$, then for $\sigma_w^2 < \min(\frac{1}{M_\phi}, \frac{1}{C_\phi})$ and $\sigma_b \geq 0$, we have $(\sigma_b, \sigma_w) \in D_{\phi,var} \cap D_{\phi,corr}$ and $K_{\phi,var}(\sigma_b, \sigma_w) = K_{\phi,corr}(\sigma_b, \sigma_w) = \infty$.*

The proof of Lemma 1 is straightforward. We prove that $\sup F'(x) = \sigma_w^2 M_\phi$ and then apply the Banach fixed point theorem. Similar ideas are used for $C_{\phi,\delta}$.

*Example*: For ReLU activation function, we have $M_{ReLU} = 1/2$ and $C_{ReLU,\delta} \leq 1$ for any $\delta > 0$.

In the domain of convergence $D_{\phi,var} \cap D_{\phi,corr}$, for all $a, b \in \mathbb{R}^d$, we have $y_i^\infty(a) = y_i^\infty(b)$ almost surely and the outputs of the network are constant functions. Figures 1(a) and 1(b) illustrate this behaviour for ReLU and Tanh with inputs in $[0, 1]^2$ using a network of depth $L = 20$ with $N_l = 300$ neurons per layer. The draws of outputs of these networks are indeed almost constant.

Under the conditions of Lemma 1, both the variance and the correlations converge exponentially fast (contraction mapping). To refine this convergence analysis, (Schoenholz et al., 2017) established the existence of $\epsilon_q$ and $\epsilon_c$ such that $|q_a^l - q| \sim e^{-l/\epsilon_q}$ and $|c_{ab}^l - 1| \sim e^{-l/\epsilon_c}$ when fixed points exist. The quantities $\epsilon_q$ and $\epsilon_c$ are called 'depth scales' since they represent the range of depth to which the variance and correlation can propagate without being exponentially close to their limits. More precisely, if we write $\chi_1 = \sigma_w^2 \mathbb{E}[\phi'(\sqrt{q}Z)^2]$ and $\alpha = \chi_1 + \sigma_w^2 \mathbb{E}[\phi''(\sqrt{q}Z)\phi(\sqrt{q}Z)]$ then the depth scales are given by $\epsilon_q = -\log(\alpha)^{-1}$ and $\epsilon_c = -\log(\chi_1)^{-1}$. The equation $\chi_1 = 1$ corresponds to an infinite depth scale of the correlation. It is called the EOC as it separates two phases: an ordered phase where the correlation converges to 1 if $\chi_1 < 1$ and a chaotic phase where $\chi_1 > 1$ and the correlations do not converge to 1. In this chaotic regime, it has been observed in (Schoenholz et al., 2017) that the correlations converge to some value $c < 1$ when $\phi(x) = \text{Tanh}(x)$ and that $c$ is independent of the correlation between the inputs. This means that very close inputs (in terms of correlation) lead to very different outputs. Therefore, in the chaotic phase, at the limit of infinite width and depth, the output function of the neural network is non-continuous everywhere. Figure 1(c) shows an example of such behaviour for Tanh.
(Schoenholz et al., 2017) focused on the existence of the depth scales in the Ordered/Chaotic phase and the empirical success of the Edge of Chaos initialization. This success was the motivation behind our work which provides a theoretical analysis of the Edge of Chaos for different activation functions. Let us first give a formal definition of the Edge of Chaos.

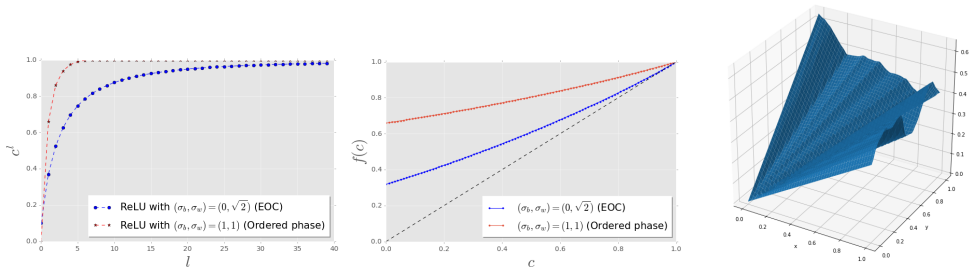
**Definition 2** (Edge of Chaos)**.** *For $(\sigma_b, \sigma_w) \in D_{\phi,var}$, let $q$ be the limiting variance[2]. The Edge of Chaos (EOC) is the set of values of $(\sigma_b, \sigma_w)$ satisfying $\chi_1 = \sigma_w^2 \mathbb{E}[\phi'(\sqrt{q}Z)^2] = 1$.*

To further study the EOC regime, the next lemma introduces a function $f$ called the 'correlation function' showing that that the correlations have the same asymptotic behaviour as the time-homogeneous dynamical system $c_{ab}^{l+1} = f(c_{ab}^l)$.

**Lemma 2.** *Let $(\sigma_b, \sigma_w) \in D_{\phi,var} \cap D_{\phi,corr}$ such that $q > 0$, $a, b \in \mathbb{R}^d$ and $\phi$ a measurable function such that $\sup_{x\in S} \mathbb{E}[\phi(xZ)^2] < \infty$ for all compact sets $S$. Define $f_l$ by $c_{ab}^{l+1} = f_l(c_{ab}^l)$ and $f$ by $f(x) = \frac{\sigma_b^2 + \sigma_w^2 \mathbb{E}[\phi(\sqrt{q}Z_1)\phi(\sqrt{q}(xZ_1 + \sqrt{1-x^2}Z_2))}{q}$. Then $\lim_{l\to\infty} \sup_{x\in[0,1]} |f_l(x) - f(x)| = 0$.*

The condition on $\phi$ in Lemma 2 is violated only by activation functions with square exponential growth (which are not used in practice), so from now onwards, we use this

[2]The limiting variance is a function of $(\sigma_b, \sigma_w)$ but we do not emphasize it notationally.




(a) Convergence of the correlation to 1 with $c^0 = 0.1$

(b) Correlation function $f$

(c) Output of 100x20 ReLU network on the EOC

*Figure 2.* Impact of the EOC initialization on the correlation and the correlation function. In (a), the correlation converges to 1 at a sub-exponential rate when the network is initialized on the EOC. In (b), the correlation function $f$ satisfies $f'(1) = 1$ on the EOC.

approximation in our analysis. Note that being on the EOC is equivalent to $(\sigma_b, \sigma_w)$ satisfying $f'(1) = 1$. In the next section, we analyze this phase transition carefully for a large class of activation functions.

## 3. Edge of Chaos

To illustrate the effect of the initialization on the EOC, we plot in Figure 2(c) the output of a ReLU neural network with 20 layers and 100 neurons per layer with parameters $(\sigma_b^2, \sigma_w^2) = (0, 2)$ (as we will see later EOC $= \{(0, \sqrt{2})\}$ for ReLU). Unlike the output in Figure 1(a), this output displays much more variability. However, we prove below that the correlations still converge to 1 even in the EOC regime, albeit at a slower rate.

### 3.1. ReLU-like activation functions

ReLU has replaced classical activations (sigmoid, Tanh,...) which suffer from gradient vanishing (see e.g. (Glorot et al., 2011) and (Nair and Hinton, 2010)). Many variants such as Leaky-ReLU were also shown to enjoy better performance in test accuracy (Xu et al., 2015). This motivates the analysis of such functions from an initialization point of view. Let us first define this class.

**Definition 3** (ReLU-like functions)**.** *A function $\phi$ is ReLU-like if it is of the form*

$$\phi(x) = \begin{cases} \lambda x & \text{if } x > 0 \\ \beta x & \text{if } x \leq 0 \end{cases}$$

*where $\lambda, \beta \in \mathbb{R}$.*

ReLU corresponds to $\lambda = 1$ and $\beta = 0$. For this class of activation functions, the EOC in terms of definition 2 is reduced to the empty set. However, we can define a weak version of the EOC for this class. From Lemma 1, when $\sigma_w < \sqrt{\frac{2}{\lambda^2+\beta^2}}$, the variances converge to $q = \frac{\sigma_b^2}{1-\sigma_w^2/2}$ and the correlations converge to 1 exponentially fast. If $\sigma_w > \sqrt{\frac{2}{\lambda^2+\beta^2}}$ the variances converge to infinity. We then have the following result.

**Lemma 3** (Weak EOC)**.** *Let $\phi$ be a ReLU-like function with $\lambda, \beta$ defined as above. Then $f_l'$ does not depend on $l$, and $f_l'(1) = 1$ and $q^l$ bounded holds if and only if $(\sigma_b, \sigma_w) = (0, \sqrt{\frac{2}{\lambda^2+\beta^2}})$.*

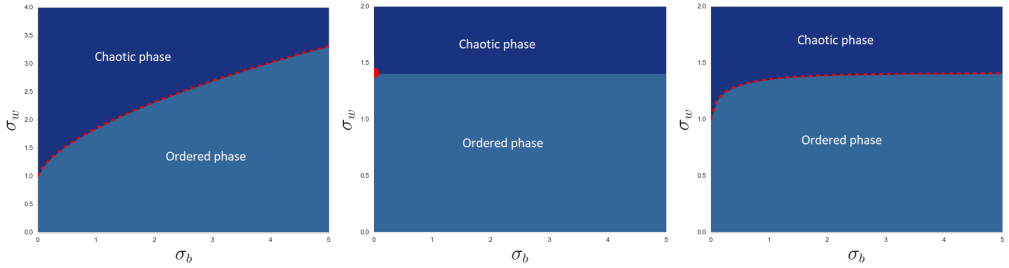
*We call the singleton $\{(0, \sqrt{\frac{2}{\lambda^2+\beta^2}})\}$ the weak EOC.*

The non existence of EOC for ReLU-like activation in the sense of definition 2 is due to the fact that the variance is unchanged ($q_a^l = q_a^1$) on the weak EOC, so that the limiting variance $q$ depends on $a$. However, this does not impact the analysis of the correlations, therefore, hereafter the weak EOC is also called the EOC.

This class of activation functions has the interesting property of preserving the variance across layers when the network is initialized on the EOC. We show in Proposition 1 below that, in the EOC regime, the correlations converge to 1 at a slower rate (slower than exponential). We only present the result for ReLU but the generalization to the whole class is straightforward.

**Example: ReLU**: The EOC is reduced to the singleton $(\sigma_b^2, \sigma_w^2) = (0, 2)$, hence we should initialize ReLU networks using the parameters $(\sigma_b^2, \sigma_w^2) = (0, 2)$. This result coincides with the recommendation in (He et al., 2015) whose objective was to make the variance constant as the input propagates but who did not analyze the propagation of the correlations. (Klambauer et al., 2017) performed a similar analysis by using the 'Scaled Exponential Linear Unit' activation (SELU) that makes it possible to center the mean and normalize the variance of the post-activation $\phi(y)$. The propagation of the correlations was not discussed therein either.

Figure 2(b) displays the correlation function $f$ for two differ-




(a) Tanh (b) ReLU (c) ELU

*Figure 3.* EOC curves for different activation functions (red dashed line). For smooth activation functions (Figures (a) and (c)), the EOC is a curve in the plane $(\sigma_b, \sigma_w)$, while it is reduced to a single point for ReLU.

ent sets of parameters $(\sigma_b, \sigma_w)$. The blue graph corresponds to the EOC $(\sigma_b^2, \sigma_w^2) = (0, 2)$, and the red one corresponds to an ordered phase $(\sigma_b, \sigma_w) = (1, 1)$.

In the next result, we show that a fully connected feedforward ReLU network initialized on the EOC (weak sense) acts as if it has residual connections in terms of correlation propagation. We further show that the correlations converge to 1 at a polynomial rate of $1/l^2$ on the EOC instead of an exponential rate in the ordered phase. Having a slow convergence rate for the correlation ensures that for the same depth $L$, the output function is relatively more expressive compared to the output function of a network with exponential convergence rate for the correlation (Ordered/Chaotic initialization). Knowing that a well-behaved output function at initialization is a good starting point for any gradient based algorithm, this suggests that the EOC initialization would yield better performance results compared to other initializations (Ordered/Chaotic), especially for deep neural networks.

**Proposition 1** (EOC acts as Residual connections)**.** *Consider a ReLU network with parameters $(\sigma_b^2, \sigma_w^2) = (0, 2) \in$ EOC and correlations $c_{ab}^l$. Consider also a ReLU network with simple residual connections given by*

$$\overline{y}_i^l(a) = \overline{y}_i^{l-1}(a) + \sum_{j=1}^{N_{l-1}} \overline{W}_{ij}^l \phi(\overline{y}_j^{l-1}(a)) + \overline{B}_i^l$$

*where $\overline{W}_{ij}^l \sim \mathcal{N}(0, \frac{\overline{\sigma}_w^2}{N_{l-1}})$ and $\overline{B}_i^l \sim \mathcal{N}(0, \overline{\sigma}_b^2)$. Let $\overline{c}_{ab}^l$ be the corresponding correlation. Then, for any $\overline{\sigma}_w > 0$ and $\overline{\sigma}_b = 0$, there exists a constant $\gamma > 0$ such that*

$$1 - c_{ab}^l \sim \gamma(1 - \overline{c}_{ab}^l) \sim \frac{9\pi^2}{2l^2} \quad as \quad l \to \infty$$

### 3.2. Smooth activation functions

It is not easy to characterize the set of activation functions for which the EOC is non trivial. For example, a shifted version of the the Softplus activation function has a trivial EOC with only $\sigma_b = 0$. In this case, we also have $q = 0$ (the limiting variance) which implies that the output function is null. However, we can show that under some conditions on the transforms $V[\phi]$ and $[\phi']$, the existence of a non-trivial EOC is guaranteed.

**Algorithm 1** EOC curve

**Input:** $\phi$ satisfying conditions of Proposition 2, $\sigma_b$
Initialize $q = 0$
**while** $q$ has not converged **do**
  $q = \sigma_b^2 + \frac{V[\phi](q)}{V[\phi'](q)}$
**end while**
**return** $(\sigma_b, \frac{1}{\sqrt{V[\phi'](q)}})$

**Proposition 2.** *Let $\phi \in \mathcal{D}_g^1$ be non ReLU-like such that $\phi(0) = 0$ and $\phi'(0) \neq 0$. Assume that $V[\phi]$ is non-decreasing and $V[\phi']$ is non-increasing. Let $\sigma_{max} := \sqrt{\sup_{x \geq 0} |x - \frac{V[\phi](x)}{V[\phi'](x)}|}$ and for $\sigma_b < \sigma_{max}$ let $q_{\sigma_b}$ be the smallest fixed point of the function $\sigma_b^2 + \frac{V[\phi]}{V[\phi']}$. Then we have EOC $= \{(\sigma_b, \frac{1}{\sqrt{\mathbb{E}[\phi'(\sqrt{q_{\sigma_b}} Z)^2]}}) : \sigma_b < \sigma_{max}\}$.*

*Example :* Tanh and ELU (defined by $\phi_{ELU}(x) = x$ for $x \geq 0$ and $\phi_{ELU}(x) = e^x - 1$ for $x < 0$) satisfy all conditions of Proposition 2. We prove in the Appendix that SiLU (a.k.a Swish) has an EOC.

Using Proposition 2, we propose Algorithm 1 to determine the EOC curves.

Figure 3 shows the EOC curves for different activation functions. For ReLU, the EOC is reduced to a point while smooth activation functions have an EOC curve (ELU is a smooth approximation of ReLU).

A natural question which arises from the analysis above is whether we can have $\sigma_{max} = \infty$. The answer is yes for the following large class of 'Tanh-like' activation functions.

**Definition 4** (Tanh-like activation functions)**.** *Let $\phi \in \mathcal{D}^2(\mathbb{R}, \mathbb{R})$. $\phi$ is Tanh-like if*

1. *$\phi$ bounded, $\phi(0) = 0$, and for all $x \in \mathbb{R}$, $\phi'(x) \geq 0$, $x\phi''(x) \leq 0$ and $x\phi(x) \geq 0$.*
2. *There exist $\alpha > 0$ such that $|\phi'(x)| \gtrsim e^{-\alpha|x|}$ for large $x$ (in norm).*

**Lemma 4.** *Let $\phi$ be a Tanh-like activation function, then $\phi$ satisfies all conditions of Proposition 2 and EOC $= \{(\sigma_b, \frac{1}{\sqrt{\mathbb{E}[\phi'(\sqrt{q}Z)^2]}}) : \sigma_b \in \mathbb{R}^+\}$.*

Recall that the convergence rate of the correlation to 1 for ReLU-like activations on the EOC is $\mathcal{O}(1/\ell^2)$. We can improve this rate by taking a sufficiently regular activation function. Let us first define a regularity class $\mathcal{A}$.

**Definition 5.** *Let $\phi \in \mathcal{D}_g^2$. We say that $\phi$ is in $\mathcal{A}$ if there exists $n \geq 1$, a partition $(S_i)_{1\leq i\leq n}$ of $\mathbb{R}$ and $g_1, g_2, ..., g_n \in \mathcal{C}_g^2$ such that $\phi^{(2)} = \sum_{i=1}^n 1_{S_i} g_i$.*

This class includes activations such as Tanh, SiLU, ELU (with $\alpha = 1$). Note that $\mathcal{D}_g^k \subset \mathcal{A}$ for all $k \geq 3$.
For activation functions in $\mathcal{A}$, the next proposition shows that the correlation converges to 1 at the rate $\mathcal{O}(1/\ell)$ which is better than $\mathcal{O}(1/\ell^2)$ of ReLU-like activation functions.

**Proposition 3** (Convergence rate for smooth activations)**.** *Let $\phi \in \mathcal{A}$ such that $\phi$ is non-linear (i.e. $\phi^{(2)}$ is non-identically zero). Then, on the EOC, we have $1 - c^l \sim \frac{\beta_q}{l}$ where $\beta_q = \frac{2\mathbb{E}[\phi'(\sqrt{q}Z)^2]}{q\mathbb{E}[\phi''(\sqrt{q}Z)^2]}$*

Choosing a smooth activation function from class $\mathcal{A}$ is therefore better for deep neural networks since it provides deeper information propagation. This could explain for example why smooth versions of ReLU such as ELU perform better (Clevert et al., 2016) (see experimental results). Figure 3.2 shows the evolution of the correlation through the network layers for different activation functions. For function in $\mathcal{A}$ (Tanh and ELU), the graph shows a rate of $\mathcal{O}(1/\ell)$ as expected compared to $\mathcal{O}(1/\ell^2)$ for ReLU.

*Remark:* The convergence rate of $\mathcal{O}(1/\ell)$ is optimal. This is because this rate is fixed by the Taylor expansion of the correlation function around 1 as shown in the proof of Proposition 3.

So far, we have discussed the impact of the EOC and the smoothness of the activation function on the behaviour of $c^l$. We now refine this analysis by studying $\beta_q$ as a function of $(\sigma_b, \sigma_w)$. We also show that $\beta_q$ plays a more important role in the information propagation process. Indeed, we show that $\beta_q$ controls the propagation of the correlation and the back-propagation of the Gradients. For the back-propagation part, we use the approximation that the weights used during forward propagation are independent of the weights used during backpropagation. This simplifies the calculations for the gradient backpropagation; see (Schoenholz et al., 2017) for details and (Yang, 2019) for a theoretical justification.

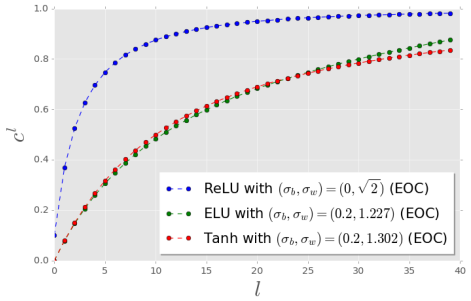



*Figure 4.* Impact of the smoothness of the activation function on the convergence of the correlations on the EOC. The convergence rate for ReLU is $\mathcal{O}(1/\ell^2)$ and $\mathcal{O}(1/\ell)$ for Tanh and ELU.

**Proposition 4.** *Let $\phi \in \mathcal{A}$ be a non-linear activation function such that $\phi(0) = 0$, $\phi'(0) \neq 0$. Assume that $V[\phi]$ is non-decreasing and $V[\phi'])$ is non-increasing, and let $\sigma_{max} > 0$ be defined as in Proposition 2. Let $E$ be a differentiable loss function and define the gradient with respect to the $l^{th}$ layer by $\frac{\partial E}{\partial y^l} = (\frac{\partial E}{\partial y_i^l})_{1\leq i\leq N_l}$ and let $\tilde{Q}_{ab}^l = \mathbb{E}[\frac{\partial E}{\partial y^l(a)}^T \frac{\partial E}{\partial y^l(b)}]$ (Covariance matrix of the gradients during backpropagation). Recall that $\beta_q = \frac{2\mathbb{E}[\phi'(\sqrt{q}Z)^2]}{q\mathbb{E}[\phi''(\sqrt{q}Z)^2]}$.*

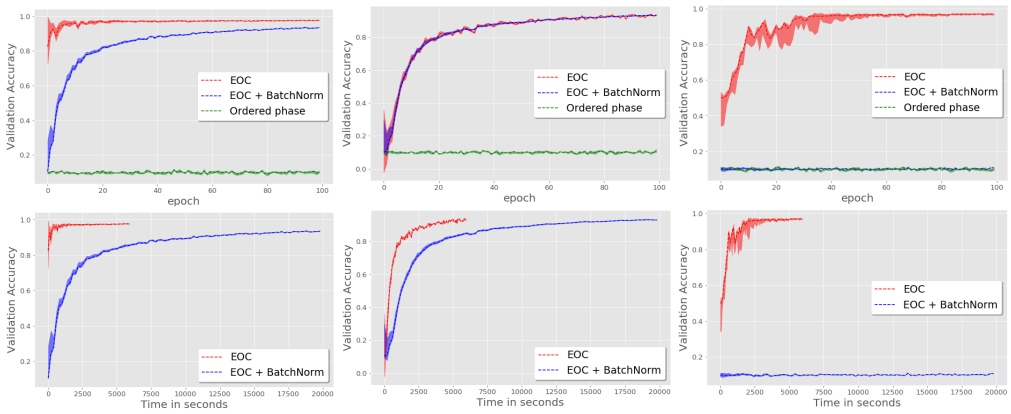
*Then, for any $\sigma_b < \sigma_{max}$, by taking $(\sigma_b, \sigma_w) \in EOC$ we have*

- $\sup_{x\in[0,1]} |f(x) - x| \leq \frac{1}{\beta_q}$
- *For $l \geq 1$, $|\frac{\mathrm{Tr}(\tilde{Q}_{ab}^l)}{\mathrm{Tr}(\tilde{Q}_{ab}^{l+1})} - 1| \leq \frac{2}{\beta_q}$*

*Moreover, we have*

$$\lim_{\substack{\sigma_b\to 0 \\ (\sigma_b,\sigma_w)\in EOC}} \beta_q = \infty.$$

The result of Proposition 4 suggests that by taking small $\sigma_b$, we can achieve two important things. First, it makes the function $f$ close to the identity function, this slows further the convergence of the correlations to 1, i.e., the information propagates deeper inside the network. Note that the only activation functions satisfying $f(x) = x$ for all $x \in [0, 1]$ are linear functions which are not useful. Second, it makes the Trace of the covariance matrix of the gradients approximately constant through layers, which means, we avoid vanishing of the information during backpropagation (More precisely, we preserve the overall spectrum of the covariance matrix since the Trace is the sum of the eigenvalues).
We also have $\lim_{\sigma_b\to 0} q = 0$ so that if $\sigma_b$ too small then $y^l(a) \approx 0$. Hence, a trade-off has to be taken into account when initializing on the EOC. Using Proposition 4, we can deduce the maximal depth to which the correlations




(a) ELU (b) ReLU (c) Tanh

*Figure 5.* 100 epochs of the training curve (test accuracy) for different activation functions for depth 200 and width 300 using SGD. The red curves correspond to the EOC, the green ones corresponds to an ordered phase, and the blue curves corresponds to an Initialization on the EOC plus a Batch Normalization after each layer. The upper figures show the test accuracies with respect to the epochs while the lower figures show the accuracies with respect to time.

can propagate without being within a distance $\epsilon$ to 1. Indeed, we have for all $l$, $|c^{l+1} - c^l| \leq \frac{1}{\beta_q}$, therefore for $L \geq 1$, $|c^L - c^0| \leq \frac{L}{\beta_q}$. Assuming $c^0 < c < 1$ for all inputs where $c$ is a constant, the maximal depth we can reach without loosing $(1-\epsilon) \times 100\%$ of the information is $L_{max} = \lfloor \beta_q(1 - c - \epsilon) \rfloor$, this satisfies $\lim_{\sigma_b \to 0} L_{max} = \infty$.

**Choice of $\sigma_b$ on the Edge of Chaos :** Given a network of depth $L$, it follows that selecting a value of $\sigma_b$ on the EOC such that $\beta_q \approx L$ appears appropriate.

We verify numerically the benefits of this rule in the next section.
Note that ReLU-like activation functions do not satisfy conditions of Proposition 4. The next lemma gives easy-to-verify sufficient conditions for Proposition 4.

**Lemma 5.** *Let $\phi \in \mathcal{A}$ such that $x\phi(x)\phi'(x) \geq 0$ and $\phi(x)\phi''(x) \leq 0$ for all $x \in \mathbb{R}$. Then, $\phi$ satisfies all conditions of Proposition 4.*

*Example*: Tanh and ELU satisfy all conditions of Lemma 5. This may partly explain why ELU performs experimentally better than ReLU (see next section). Another example is an activation function of the form $\lambda x + \beta \text{Tanh}(x)$ where $\lambda, \beta \in \mathbb{R}$. We check the performance of these activations in the next section.

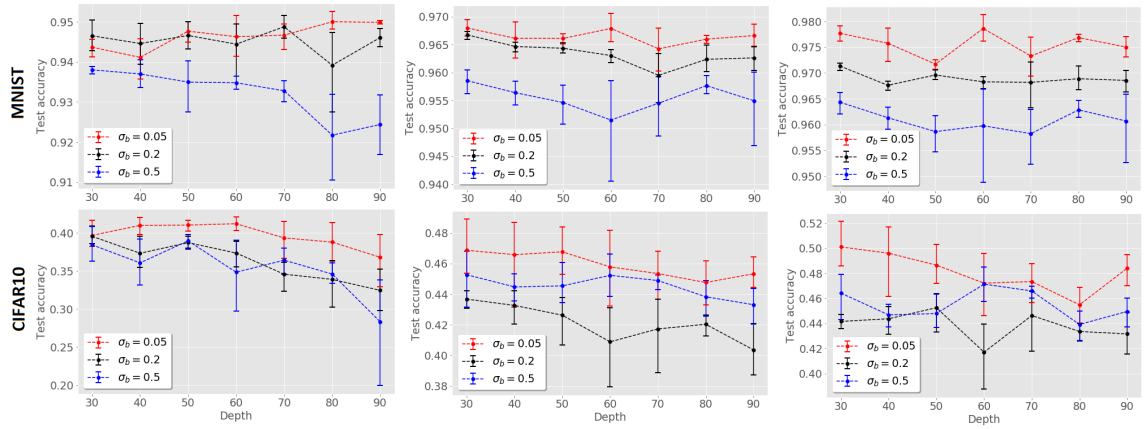
## 4. Experiments

In this section, we demonstrate empirically the theoretical results established above. We show that:

- For deep networks, only an initialization on the EOC could make the training possible, and the initialization on the EOC performs better than Batch Normalization.
- Smooth activation functions in the sense of Proposition 3 perform better than ReLU-like activation, especially for very deep networks.
- Choosing the right point on the EOC further accelerates the training.

We demonstrate empirically our results on the MNIST and CIFAR10 datasets for depths $L$ between 10 and 200 and width 300. We use SGD and RMSProp for training. We performed a grid search between $10^{-6}$ and $10^{-2}$ with exponential step of size 10 to find the optimal learning rate. For SGD, a learning rate of $\sim 10^{-3}$ is nearly optimal for $L \leq 150$, for $L > 150$, the best learning rate is $\sim 10^{-4}$. For RMSProp,$10^{-5}$ is nearly optimal for networks with depth $L \leq 200$ (for deeper networks, $10^{-6}$ gives better results). We use a batchsize of 64.

**Initialization on the Edge of Chaos**. We initialize randomly the network by sampling $W_{ij}^l \overset{iid}{\sim} \mathcal{N}(0, \sigma_w^2/N_{l-1})$ and $B_i^l \overset{iid}{\sim} \mathcal{N}(0, \sigma_b^2)$. Figure 5 shows that the initialization on the EOC dramatically accelerates the training for ELU, ReLU and Tanh. The initialization in the ordered phase (here we used $(\sigma_b, \sigma_w) = (1, 1)$ for all activations) results




(a) Epoch 10 (b) Epoch 50 (c) Epoch 100

*Figure 6.* Test accuracies for Tanh network with depths between 30 and 90 and width 300 using different points on the EOC.

*Table 1.* Test accuracies for width 300 and depth 200 with different activation function on MNIST and CIFAR10 after 100 epochs

| **MNIST** | EOC | EOC + BN | ORD PHASE |
|---|---|---|---|
| RELU | **93.57± 0.18** | 93.11± 0.21 | 10.09± 0.61 |
| ELU | **97.62± 0.21** | 93.41± 0.3 | 10.14± 0.51 |
| TANH | **97.20± 0.3** | 10.74± 0.1 | 10.02± 0.13 |

| **CIFAR10** | EOC | EOC + BN | ORD PHASE |
|---|---|---|---|
| RELU | **36.55± 1.15** | 35.91± 1.52 | 9.91± 0.93 |
| ELU | **45.76± 0.91** | 44.12± 0.93 | 10.11± 0.65 |
| TANH | **44.11± 1.02** | 10.15± 0.85 | 9.82± 0.88 |

*Table 2.* Test accuracies for width 300 and depth 200 with different activation function on MNIST and CIFAR10 after 10, 50 and 100 epochs

| **MNIST** | EPOCH 10 | EPOCH 50 | EPOCH 100 |
|---|---|---|---|
| RELU | 66.76± 1.95 | 88.62± 0.61 | 93.57± 0.18 |
| ELU | **96.09± 1.55** | **97.21± 0.31** | **97.62± 0.21** |
| TANH | 89.75± 1.01 | 96.51± 0.51 | 97.20± 0.3 |

| **CIFAR10** | EPOCH 10 | EPOCH 50 | EPOCH 100 |
|---|---|---|---|
| RELU | 26.46± 1.68 | 33.74± 1.21 | 36.55± 1.15 |
| ELU | **35.95± 1.83** | **45.55± 0.91** | **47.76± 0.91** |
| TANH | 34.12± 1.23 | 43.47± 1.12 | 44.11± 1.02 |

in the optimization algorithm being stuck eventually at a very poor test accuracy of $\sim 0.1$ (equivalent to selecting the output uniformly at random). Figure 5 also shows that EOC combined to BatchNorm results in a worse learning curve and dramatically increases the training time. Note that it is crucial here to initialize BatchNorm parameters to $\alpha = 1$ and $\beta = 0$ in order to keep our analysis on the forward propagation on the EOC valid for networks with BatchNorm. Table 1 presents test accuracy after 100 epochs for different activation functions and different training methods (EOC, EOC+BatchNorm, Ordered phase) on MNIST and CIFAR10. For all activation functions but Softplus, EOC initialization leads to the best performance. Adding BatchNorm to the EOC initialization makes the training worse, this can be explained the fact that parameters $\alpha$ and $\beta$ are also modified during the first backpropagation. This invalidates the EOC results for gradient backpropagation (see proof of Proposition 4).

**Impact of the smoothness of the activation function on the training.** Table 2 shows the test accuracy at different epochs for ReLU, ELU, Tanh. Smooth activation functions perform better than ReLU. More experimental results with RMSProp and other activation functions of the form $x + \alpha\text{Tanh}(x)$ are provided in the supplementary material.

**Selection of a point on the EOC.** We have shown that a sensible choice is to select $\sigma_b$ such that $L \sim \beta_q$ on the EOC. Figure 6 shows test accuracy of a Tanh network for different depths using $\sigma_b \in \{0.05, 0.2, 0.5\}$. With $\sigma_b = 0.05$, we have $\beta_q \sim 50$. We see for depth 50, the red curve ($\sigma_b = 0.05$) is the best. For other depths $L$ between 30 and 90, $\sigma_b = 0.05$ is the value that makes $\beta_q$ the closest to $L$ among $\{0.05, 0.2, 0.5\}$, which explains why the red curve is approximately better for all depths between 30 and 90. To further confirm this finding, we search numerically for the best $\sigma_b \in \{2k \times 10^{-2} : k \in [1, 50]\}$ for depths $30, 100, 200$. Table 3 shows the results.

*Table 3.* Best test accuracy achieved after 100 epochs with Tanh on MNIST

| DEPTH | $L = 30$ | $L = 50$ | $L = 200$ |
|---|---|---|---|
| $2k \times 10^{-2}$ | 0.080 | 0.040 | 0.020 |
| WITH RULE $\beta_q \approx L$ | 0.071 | 0.030 | 0.022 |

## 5. Discussion

The Gaussian process approximation of Deep Neural Networks was used by (Schoenholz et al., 2017) to show that, in the regime, very deep Tanh networks are trainable only on the EOC. We give here a comprehensive analysis of the EOC for a large class of activation functions. We also prove that smoothness plays a major role in terms of signal propagation. Numerical results in Table 2 confirm this finding. Moreover, we introduce a rule to choose the optimal point on the EOC, this point is a function of the depth. As the depth goes to infinity (e.g. $L = 400$), we need smaller $\sigma_b$ to achieve the best signal propagation. However, the limiting variance $q$ also becomes close to zero as $\sigma_b$ goes to zero. To avoid this problem, one possible solution is to change the activation function to ensure that the coefficient $\beta_q$ becomes large independently of the choice of $\sigma_b$ on the EOC. Indeed, we show in the supplementary material that by choosing activation function of the form $x + \alpha\text{Tanh}(x)$, we can make $\beta_q$ arbitrarily large without changing $\sigma_b$.

Our results have implications for Bayesian neural networks which have received renewed attention lately; see, e.g., (Hernandez-Lobato and Adams, 2015) and (Lee et al., 2018). They indeed indicate that, if one assigns i.i.d. Gaussian prior distributions to the weights and biases, we need to select not only the prior parameters $(\sigma_b, \sigma_w)$ on the EOC but also an activation function satisfying Proposition 3 to obtain a non-degenerate prior on the induced function space.

## References


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

I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

G.F. Montufar, R. Pascanu, K. Cho, and Y. Bengio. On the number of linear regions of deep neural networks. *Advances in Neural Information Processing Systems*, 27: 2924–2932, 2014.

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

R.M. Neal. *Bayesian Learning for Neural Networks*, volume 118. Springer Science & Business Media, 1995.

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

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

D.A. Clevert, T. Unterthiner, and S. Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). *ICLR*, 2016.

D. Pedamonti. Comparison of non-linear activation functions for deep neural networks on mnist classification task. *arXiv 1804.02763*, 2018.

P. Ramachandran, B. Zoph, and Q.V. Le. Searching for activation functions. *arXiv e-print 1710.05941*, 2017.

M. Milletarí, T. Chotibut, and P. Trevisanutto. Expectation propagation: a probabilistic view of deep feed forward networks. *arXiv:1805.08786*, 2018.

X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier neural networks. *AISTATS*, 2011.

V. Nair and G.E. Hinton. Rectified linear units improve restricted boltzmann machines. *ICML*, 2010.

B. Xu, N. Wang, T. Chen, and M. Li. Empirical evaluation of rectified activations in convolution network. *arXiv:1505.00853*, 2015.

K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *ICCV*, 2015.

G. Klambauer, T. Unterthiner, and A. Mayr. Self-normalizing neural networks. *Advances in Neural Information Processing Systems*, 30, 2017.

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

J. M. Hernandez-Lobato and R.P. Adams. Probabilistic backpropagation for scalable learning of Bayesian neural networks. *ICML*, 2015.

# A. Proofs

We provide in this supplementary material the proofs of theoretical results presented in the main document, and we give additive theoretical and experimental results. For the sake of clarity we recall the results before giving their proofs.

### A.1. Convergence to the fixed point: Proposition 1

**Lemma 1.** *Let $M_\phi := \sup_{x\geq 0}\mathbb{E}[|\phi'^2(xZ)+\phi''(xZ)\phi(xZ)|]$. Suppose $M_\phi < \infty$, then for $\sigma_w^2 < \frac{1}{M_\phi}$ and any $\sigma_b$, we have $(\sigma_b,\sigma_w) \in D_{\phi,var}$ and $K_{\phi,var}(\sigma_b,\sigma_w) = \infty$*

*Moreover, let $C_{\phi,\delta} := \sup_{x,y\geq 0,|x-y|\leq\delta,c\in[0,1]}\mathbb{E}[|\phi'(xZ_1)\phi'(y(cZ_1+\sqrt{1-c^2}Z_2)|]$. Suppose $C_{\phi,\delta} < \infty$ for some positive $\delta$, then for $\sigma_w^2 < \min(\frac{1}{M_\phi},\frac{1}{C_\phi})$ and any $\sigma_b$, we have $(\sigma_b,\sigma_w) \in D_{\phi,var}\cap D_{\phi,corr}$ and $K_{\phi,var}(\sigma_b,\sigma_w) = K_{\phi,corr}(\sigma_b,\sigma_w) = \infty$.*

*Proof.* To abbreviate the notation, we use $q^l := q_a^l$ for some fixed input $a$.

**Convergence of the variances:** We first consider the asymptotic behaviour of $q^l = q_a^l$. Recall that $q^l = F(q^{l-1})$ where

$$F(x) = \sigma_b^2 + \sigma_w^2\mathbb{E}[\phi(\sqrt{x}Z)^2].$$

The first derivative of this function is given by

$$F'(x) = \sigma_w^2\mathbb{E}[\frac{Z}{\sqrt{x}}\phi'(\sqrt{x}Z)\phi(\sqrt{x}Z)] = \sigma_w^2\mathbb{E}[\phi'(\sqrt{x}Z)^2+\phi''(\sqrt{x}Z)\phi(\sqrt{x}Z)], \tag{4}$$

where we use Gaussian integration by parts, $\mathbb{E}[ZG(Z)] = \mathbb{E}[G'(Z)]$, an identity satisfied by any function $G$ such that $\mathbb{E}[|G'(Z)|] < \infty$.

Using the condition on $\phi$, we see that the function $F$ is a contraction mapping for $\sigma_w^2 < \frac{1}{M_\phi}$ and the Banach fixed-point theorem guarantees the existence of a unique fixed point $q$ of $F$, with $\lim_{l\to+\infty} q^l = q$. Note that this fixed point depends only on $F$, therefore this is true for any input $a$ and $K_{\phi,var}(\sigma_b,\sigma_w) = \infty$.

**Convergence of the covariances:** Since $M_\phi < \infty$, then for all $a, b \in \mathbb{R}^d$ there exists $l_0$ such that $|\sqrt{q_a^l} - \sqrt{q_b^l}| < \delta$ for all $l > l_0$. Let $l > l_0$, using Gaussian integration by parts, we have

$$\frac{dc_{ab}^{l+1}}{dc_{ab}^l} = \sigma_w^2\mathbb{E}[|\phi'(\sqrt{q_a^l}Z_1)\phi'(\sqrt{q_b^l}(c_{ab}^lZ_1+\sqrt{1-(c_{ab}^l)^2}Z_2)|].$$

We cannot use the Banach fixed point theorem directly because the integrated function here depends on $l$ through $q^l$. For ease of notation, we write $c^l := c_{ab}^l$. We have

$$|c^{l+1}-c^l| = |\int_{c^{l-1}}^{c^l}\frac{dc^{l+1}}{dc^l}(x)dx| \leq \sigma_w^2C_\phi|c^l-c^{l-1}|.$$

Therefore, for $\sigma_w^2 < \min(\frac{1}{M_\phi},\frac{1}{C_\phi})$, $c^l$ is a Cauchy sequence and it converges to a limit $c \in [0,1]$. At the limit

$$c = f(c) = \frac{\sigma_b^2+\sigma_w^2\mathbb{E}[\phi(\sqrt{q}z_1)\phi(\sqrt{q}(cz_1+\sqrt{1-c^2}z_2)))]}{q}.$$

The derivative of this function is given by

$$f'(x) = \sigma_w^2\mathbb{E}[\phi'(\sqrt{q}Z_1)\phi'(\sqrt{q}(xZ_1+\sqrt{1-x}Z_2)].$$

By assumption on $\phi$ and the choice of $\sigma_w$, we have $\sup_x|f'(x)| < 1$ so $f$ is a contraction and has a unique fixed point. Since $f(1) = 1$ then $c = 1$. The above result is true for any $a, b$, therefore $K_{\phi,var}(\sigma_b,\sigma_w) = K_{\phi,corr}(\sigma_b,\sigma_w) = \infty$. □

**Lemma 2.** *Let $(\sigma_b, \sigma_w) \in D_{\phi,var} \cap D_{\phi,corr}$ such that $q > 0$, $a, b \in \mathbb{R}^d$ and $\phi$ an activation function such that $\sup_{x\in K} \mathbb{E}[\phi(xZ)^2] < \infty$ for all compact sets $K$. Define $f_l$ by $c_{a,b}^{l+1} = f_l(c_{a,b}^l)$ and $f$ by $f(x) = \frac{\sigma_b^2+\sigma_w^2\mathbb{E}[\phi(\sqrt{q}Z_1)\phi(\sqrt{q}(xZ_1+\sqrt{1-x^2}Z_2))}{q}$. Then $\lim_{l\to\infty} \sup_{x\in[0,1]} |f_l(x) - f(x)| = 0$.*

*Proof.* For $x \in [0, 1]$, we have

$$f_l(x) - f(x) = (\frac{1}{\sqrt{q_a^l q_b^l}} - \frac{1}{q})(\sigma_b^2 + \sigma_w^2\mathbb{E}[\phi(\sqrt{q_a^l}Z_1)\phi(\sqrt{q_b^l}u_2(x))])$$
$$+ \frac{\sigma_w^2}{q}(\mathbb{E}[\phi(\sqrt{q_a^l}Z_1)\phi(\sqrt{q_b^l}u_2(x))] - \mathbb{E}[\phi(\sqrt{q}Z_1)\phi(\sqrt{q}u_2(x))]),$$

where $u_2(x) := xZ_1 + \sqrt{1-x^2}Z_2$. The first term goes to zero uniformly in $x$ using the condition on $\phi$ and Cauchy-Schwartz inequality. As for the second term, it can be written again as

$$\mathbb{E}[(\phi(\sqrt{q_a^l}Z_1) - \phi(\sqrt{q}Z_1))\phi(\sqrt{q_b^l}u_2(x))] + \mathbb{E}[\phi(\sqrt{q}Z_1)(\phi(\sqrt{q_b^l}u_2(x)) - \phi(\sqrt{q}u_2(x)))].$$

Using Cauchy-Schwartz and the condition on $\phi$, both terms can be controlled uniformly in $x$ by an integrable upper bound. We conclude using dominated convergence. □

**Lemma 3** (Weak EOC)**.** *Let $\phi$ be a ReLU-like function with $\lambda, \beta$ defined as above. Then $f_l'$ does not depend on $l$, and having $f_l'(1) = 1$ and $q^l$ bounded is only achieved for the singleton $(\sigma_b, \sigma_w) = (0, \sqrt{\frac{2}{\lambda^2+\beta^2}})$. The Weak EOC is defined as this singleton.*

*Proof.* We write $q^l = q_a^l$ throughout the proof. Note first that the variance satisfies the recursion:

$$q^{l+1} = \sigma_b^2 + \sigma_w^2\mathbb{E}[\phi(Z)^2]q^l = \sigma_b^2 + \sigma_w^2\frac{\lambda^2+\beta^2}{2}q^l. \tag{5}$$

For all $\sigma_w < \sqrt{\frac{2}{\lambda^2+\beta^2}}$, $q = \sigma_b^2\left(1 - \sigma_w^2(\lambda^2+\beta^2)/2\right)^{-1}$ is a fixed point. This is true for any input, therefore $K_{\phi,var}(\sigma_b, \sigma_w) = \infty$ and (i) is proved.

Now, the EOC equation is given by $\chi_1 = \sigma_w^2\mathbb{E}[\phi'(Z)^2] = \sigma_w^2\frac{\lambda^2+\beta^2}{2}$. Therefore, $\sigma_w^2 = \frac{2}{\lambda^2+\beta^2}$. Replacing $\sigma_w^2$ by its critical value in (5) yields

$$q^{l+1} = \sigma_b^2 + q^l.$$

Thus $q = \sigma_b^2 + q$ if and only if $\sigma_b = 0$, otherwise $q^l$ diverges to infinity. So the frontier is reduced to a single point $(\sigma_b^2, \sigma_w^2) = (0, \mathbb{E}[\phi'(Z)^2]^{-1})$, and the variance does not depend on $l$.

□

**Proposition 1** (EOC acts as Residual connections)**.** *Consider a ReLU network with parameters $(\sigma_b^2, \sigma_w^2) = (0, 2) \in EOC$ and let $c_{ab}^l$ be the corresponding correlation. Consider also a ReLU network with simple residual connections given by*

$$\overline{y}_i^l(a) = \overline{y}_i^{l-1}(a) + \sum_{j=1}^{N_{l-1}} \overline{W}_{ij}^l\phi(\overline{y}_j^{l-1}(a)) + \overline{B}_i^l,$$

*where $\overline{W}_{ij}^l \sim \mathcal{N}(0, \frac{\overline{\sigma}_w^2}{N_{l-1}})$ and $\overline{B}_i^l \sim \mathcal{N}(0, \overline{\sigma}_b^2)$. Let $\overline{c}_{ab}^l$ be the corresponding correlation. Then, by taking $\overline{\sigma}_w > 0$ and $\overline{\sigma}_b = 0$, there exists a constant $\gamma > 0$ such that*

$$1 - c_{ab}^l \sim \gamma(1 - \overline{c}_{ab}^l) \sim \frac{9\pi^2}{2l^2}$$

*as $l \to \infty$.*

*Proof.* Let us first give a closed-form formula of the correlation function $f$ of a ReLU network. In this case, we have $f(x) = 2\mathbb{E}[(Z_1)_+(xZ_1 + \sqrt{1-x^2}Z_2)_+]$ where $(x)_+ := x1_{x>0}$. Let $x \in [0,1]$, $f$ is differentiable and satisfies

$$f'(x) = 2\mathbb{E}[1_{Z_1>0}1_{xZ_1+\sqrt{1-x^2}Z_2>0}],$$

which is also differentiable. Simple algebra leads to

$$f^{"}(x) = \frac{1}{\pi\sqrt{1-x^2}}.$$

Since $\arcsin'(x) = \frac{1}{\sqrt{1-x^2}}$ and $f'(0) = 1/2$,

$$f'(x) = \frac{1}{\pi}\arcsin(x) + \frac{1}{2}.$$

Using the fact that $\int \arcsin = x\arcsin + \sqrt{1-x^2}$ and $f(1) = 1$, we conclude that for $x \in [0,1]$, $f(x) = \frac{1}{\pi}x\arcsin(x) + \frac{1}{\pi}\sqrt{1-x^2} + \frac{1}{2}x$.

For the residual network, we have $\overline{q}_a^l = \overline{q}_a^{l-1} + \overline{\sigma}_w^2\mathbb{E}[\phi(\sqrt{\overline{q}_a^{l-1}}Z)^2] = (1 + \frac{\overline{\sigma}_w^2}{2})\overline{q}_a^{l-1}$.

Let $\delta = \frac{1}{1+\frac{\overline{\sigma}_w^2}{2}}$. We have

$$\begin{aligned}\overline{c}_{ab}^l &= \delta\overline{c}_{ab}^{l-1} + \delta\overline{\sigma}_w^2\mathbb{E}[\phi(Z_1)\phi(U_2(\overline{c}_{ab}^{l-1}))] \\ &= \overline{c}_{ab}^{l-1} + \delta\frac{\overline{\sigma}_w^2}{2}(f(\overline{c}_{ab}^{l-1}) - \overline{c}_{ab}^{l-1})\end{aligned}$$

Now, we use Taylor expansion near to conclude. However, since $f$ is not differentiable in 1 for all orders, we use a change of variable $x = 1 - t^2$ with $t$ close to 0, then

$$\arcsin(1-t^2) = \frac{\pi}{2} - \sqrt{2}t - \frac{\sqrt{2}}{12}t^3 + O(t^5),$$

so that

$$\arcsin(x) = \frac{\pi}{2} - \sqrt{2}(1-x)^{1/2} - \frac{\sqrt{2}}{12}(1-x)^{3/2} + O((1-x)^{5/2}),$$

and

$$x\arcsin(x) = \frac{\pi}{2}x - \sqrt{2}(1-x)^{1/2} + \frac{11\sqrt{2}}{12}(1-x)^{3/2} + O((1-x)^{5/2}).$$

Since

$$\sqrt{1-x^2} = \sqrt{2}(1-x)^{1/2} - \frac{\sqrt{2}}{4}(1-x)^{3/2} + O((1-x)^{5/2}),$$

we obtain that

$$f(x) \underset{x\to 1-}{=} x + \frac{2\sqrt{2}}{3\pi}(1-x)^{3/2} + O((1-x)^{5/2}). \tag{6}$$

Since $(f(x)-x)' = \frac{1}{\pi}(\arcsin(x) - \frac{\pi}{2}) < 0$ and $f(1) = 1$, for all $x \in [0,1)$, $f(x) > x$. If $c^l < c^{l+1}$ then by taking the image by $f$ (which is increasing because $f' \geq 0$) we have that $c^{l+1} < c^{l+2}$, and we know that $c^1 = f(c^0) \geq c^0$, so by induction the sequence $c^l$ is increasing, and therefore it converges to the fixed point of $f$ which is 1.

Using a Taylor expansion of $f$ near 1, we have

$$\overline{c}_{ab}^l = \overline{c}_{ab}^{l-1} + \delta\frac{2\sqrt{2}}{3\pi}(1-\overline{c}_{ab}^{l-1})^{3/2} + O((1-\overline{c}_{ab}^{l-1})^{5/2})$$

and

$$c_{ab}^l = c_{ab}^{l-1} + \frac{2\sqrt{2}}{3\pi}(1-c_{ab}^{l-1})^{3/2} + O((1-c_{ab}^{l-1})^{5/2}).$$

Now let $\gamma_l := 1 - c_{ab}^l$ for $a, b$ fixed. We note $s = \frac{2\sqrt{2}}{3\pi}$, from the series expansion we have that $\gamma_{l+1} = \gamma_l - s\gamma_l^{3/2} + O(\gamma_l^{5/2})$ so that

$$\begin{aligned}\gamma_{l+1}^{-1/2} &= \gamma_l^{-1/2}(1 - s\gamma_l^{1/2} + O(\gamma_l^{3/2}))^{-1/2} = \gamma_l^{-1/2}(1 + \frac{s}{2}\gamma_l^{1/2} + O(\gamma_l^{3/2})) \\ &= \gamma_l^{-1/2} + \frac{s}{2} + O(\gamma_l).\end{aligned}$$

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

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

Therefore, we have $1 - c_{ab}^l \sim \frac{9\pi^2}{2l^2}$. Using the same argument for $\bar{c}_{al}^l$, we conclude.

□

**Proposition 2.** *Let $\phi \in \mathcal{D}_g^1$ be non ReLU-like function. Assume $V[\phi]$ is non-decreasing and $V[\phi']$ is non-increasing. Let $\sigma_{max} := \sqrt{\sup_{x \geq 0} |x - \frac{V[\phi](x)}{V[\phi'](x)}|}$ and for $\sigma_b < \sigma_{max}$ let $q_{\sigma_b}$ be the smallest fixed point of the function $\sigma_b^2 + \frac{V[\phi]}{V[\phi']}$. Then we have $EOC = \{(\sigma_b, \frac{1}{\sqrt{\mathbb{E}[\phi'(\sqrt{q}Z)^2]}}) : \sigma_b < \sigma_{max}\}$.*

To prove Proposition 2, we need to introduce some lemmas. The next lemma gives a characterization of ReLU-like activation functions.

**Lemma 1.1** (A Characterization of ReLU-like activations)**.** *Let $\phi \in \mathcal{D}^1(\mathbb{R}, \mathbb{R})$ such that $\phi(0) = 0$ and $\phi'$ non-identically zero. We define the function $e$ for non-negative real numbers by*

$$e(x) = \frac{V[\phi](x)}{V[\phi'](x)} = \frac{\mathbb{E}[\phi(\sqrt{x}Z)^2]}{\mathbb{E}[\phi'(\sqrt{x}Z)^2]}$$

*Then, for all $x \geq 0$, $e(x) \leq x$.*
*Moreover, the following statements are equivalent*

- *There exists $x_0 > 0$ such that $e(x_0) = x_0$.*
- *$\phi$ is ReLU-like, i.e. there exists $\lambda, \beta \in \mathbb{R}$ such that $\phi(x) = \lambda x$ if $x > 0$ and $\phi(x) = \beta x$ if $x \leq 0$.*

*Proof.* Let $x > 0$. We have for all $z \in \mathbb{R}$, $\phi(\sqrt{x}z) = \sqrt{x}\int_0^z \phi'(\sqrt{x}u)du$. This yields

$$\begin{aligned}\mathbb{E}[\phi(\sqrt{x}Z)^2] &= x\mathbb{E}[\Big(\int_0^Z \phi'(\sqrt{x}u)du\Big)^2] \\ &\leq x\mathbb{E}[|Z|\int_0^{|Z|} \phi'(\sqrt{x}u)^2 du] \\ &= x\mathbb{E}[Z\int_0^Z \phi'(\sqrt{x}u)^2 du] \\ &= x\mathbb{E}[\phi'(\sqrt{x}Z)^2 du]\end{aligned}$$

where we have used Cauchy-Schwartz inequality and Gaussian integration by parts. Therefore $e(x) \leq x$.

Now assume there exists $x_0 > 0$ such that $e(x_0) = x_0$. We have

$$\begin{aligned}\mathbb{E}[\phi(\sqrt{x_0}Z)^2] &= x_0\mathbb{E}[\Big(\int_0^Z \phi'(\sqrt{x_0}u)du\Big)^2] \\ &= x_0\mathbb{E}[1_{Z>0}\Big(\int_0^Z \phi'(\sqrt{x_0}u)du\Big)^2] + x_0\mathbb{E}[1_{Z\leq 0}\Big(\int_Z^0 \phi'(\sqrt{x_0}u)du\Big)^2] \\ &\leq x_0\mathbb{E}[1_{Z>0}\int_0^Z 1du\int_0^Z \phi'(\sqrt{x_0}u)^2du] + x_0\mathbb{E}[1_{Z\leq 0}\int_Z^0 1du\int_Z^0 \phi'(\sqrt{x_0}u)^2du].\end{aligned}$$

The equality in Cauchy-Schwartz inequality implies that
- For almost every $z > 0$, there exists $\lambda_z$ such that $\phi'(\sqrt{x_0}u) = \lambda_z$ for all $u \in [0, z]$.
- For almost every $z < 0$, there exists $\beta_z$ such that $\phi'(\sqrt{x_0}u) = \beta_z$ for all $u \in [z, 0]$.
Therefore, $\lambda_z, \beta_z$ are independent of $z$, and $\phi$ is ReLU-like.

It is easy to see that for ReLU-like activations, $e(x) = x$ for all $x \geq 0$. □

The next trivial lemma provides a sufficient condition for the existence of a fixed point of a shifted function.

**Lemma 1.2.** *Let $g \in \mathcal{C}^0(\mathbb{R}^+, \mathbb{R})$ such that $g(0) = 0$ and $g(x) \leq x$ for all $x \in \mathbb{R}^+$. Let $t_{max} := \sup_{x\geq 0} |x - g(x)|$ ($t_{max}$ may be infinite). Then, for all $t \in [0, t_{max})$, the shifted function $t + g(.)$ has a fixed point.*

*Proof.* Let $t \in [0, t_{max})$. There exists $x_0 > 0$ such that $t + g(.) < x_0 - g(x_0) + g(.)$. So we have $t + g(0) = t$ and $t + g(x_0) < x_0$, which means that $t + g(.)$ crosses the identity line, therefore the fixed point exists. □

**Corollary 1.1.** *Let $\phi \in \mathcal{D}^1(\mathbb{R}, \mathbb{R})$ such that $\phi$ is non ReLU-like. Let $t_{max} = \sup_{x\geq 0} |x - \frac{V[\phi](x)}{V[\phi'](x)}|$. Then, For any $\sigma_b^2 \in [0, t_{max})$, the shifted function $\sigma_b^2 + \frac{V[\phi]}{V[\phi']}$ has a fixed point $q$. Moreover, by taking $q$ to be the greatest fixed point, we have $\lim_{\sigma_b\to 0} q = 0$.*

The limit of $q$ is zero because it is a fixed point of the function $\frac{V[\phi](x)}{V[\phi'](x)}$ which has only 0 as a fixed point for non ReLU-like functions.

Corollary 1.1 proves the existence of a fixed point for the shifted function $\sigma_b^2 + \frac{V[\phi]}{V[\phi']}$, which is a necessary condition for $(\sigma_b, 1/\sqrt{V[\phi'](q)})$ to be in the EOC where $q$ is the smallest fixed point. It is not a sufficient condition because $q$ may not be the smallest fixed point of $\sigma_b^2 + \frac{1}{V[\phi'](q)}V[\phi]$. We further analyse this problem hereafter.

**Definition 6** (Permissible couples)**.** *Let $g, h \in \mathcal{C}(\mathbb{R}^+, \mathbb{R}^+)$ and $c > 0$. Define the function $k(x) = c + \frac{g(x)}{h(x)}$ for $x \geq 0$ and let $q = \inf\{x : k(x) = x\}$. We say that $(g, h)$ is permissible if for any $c \geq 0$ such that $q < \infty$, $q$ is the smallest fixed point of the function $c + \frac{g(.)}{h(q)}$.*

**Lemma 1.3.** *Let $g, h \in \mathcal{C}(\mathbb{R}^+, \mathbb{R}^+)$. Then the following statements are equivalent*

1. *$(g, h)$ is permissible.*
2. *For any $c > 0$ such that $q$ is finite, we have $g(q) - g(x) < (q - x)h(q)$ for $x \in [0, q)$.*

*Proof.* If $q$ is a fixed point of $c + \frac{g(.)}{h(.)}$, then $q$ is clearly a fixed point of $I(x) = c + \frac{1}{h(q)}g(x)$. Having $q$ is the smallest fixed point of $c + \frac{g(.)}{h(q)}$ is equivalent to $c + \frac{g(x)}{h(q)} > x$ for all $x \in [0, q)$. Since $c = q - \frac{g(q)}{h(q)}$, we conclude. □

**Corollary 1.2.** *Let $g, h \in \mathcal{C}(\mathbb{R}^+, \mathbb{R}^+)$. Assume $h$ is non-increasing, then $(g, h)$ is permissible.*

*Proof.* Since $h$ is non-increasing, we have for $x \in [0, q)$, $g(q) - g(x) \leq h(q)(q - c) - \frac{h(q)}{h(x)}g(x) = h(q)(q - (c + \frac{g(x)}{h(x)}))$. We conclude using the fact that $c + \frac{g(x)}{h(x)} > x$ for $x \in [0, q)$. □

**Corollary 1.3.** *Let $\phi$ be a non ReLU-like function. Assume $V[\phi]$ is non-decreasing and $(V[\phi], V[\phi'])$ is permissible. Then, for any $\sigma_b^2 < t_{max} := \sup_{x\geq 0} |x - e(x)|$, by taking $\sigma_w^2 = \frac{1}{\mathbb{E}[\phi'(\sqrt{q}Z)^2]}$, we have $(\sigma_b, \sigma_w) \in EOC$. Moreover, we have $\lim_{\sigma_b \to 0} q = 0$.*

We can omit the condition '$V[\phi]$ is non-decreasing' by choosing a small $t_{max}$. Indeed, by taking a small $\sigma_b$, the limiting variance $q$ is small, and we know that $V[\phi]$ is increasing near 0 because $V[\phi]'(0) = \phi'(0)^2 > 0$.

The proof of Proposition 2 is straightforward from corollary A.3.

**Lemma 4.** *Let $\phi$ be a Tanh-like activation function, then $\phi$ satisfies all conditions of Proposition 2 and $EOC = \{(\sigma_b, \frac{1}{\sqrt{\mathbb{E}[\phi'(\sqrt{q}Z)^2]}}) : \sigma_b \in \mathbb{R}^+\}$.*

*Proof.* For $x \geq 0$, we have $V[\phi]'(x) = \frac{1}{x}\mathbb{E}[\sqrt{x}Z\phi'(\sqrt{x}Z)\phi(\sqrt{x}Z)] \geq 0$, so $V[\phi]$ is non-decreasing. Moreover, $V[\phi']'(x) = \frac{1}{x}\mathbb{E}[\sqrt{x}Z\phi''(\sqrt{x}Z)\phi'(\sqrt{x}Z)] \leq 0$, therefore $V[\phi']$ is non-increasing. To conclude, we still have to show that $t_{max} = \infty$.

Using the second condition on $\phi$, there exists $M > 0$ such that $|\phi'(y)|^2 \geq Me^{-2\alpha|y|}$. Let $x > 0$. we have

$$\begin{aligned}
\mathbb{E}[\phi'(\sqrt{x}Z)^2] &\geq M\mathbb{E}[e^{-2\alpha|\sqrt{x}Z|}] \\
&= 2M\int_0^\infty e^{-2\alpha\sqrt{x}Z}\frac{e^{-z^2/2}}{\sqrt{2\pi}}dz \\
&= 2Me^{2\alpha^2 x}\Psi(2\alpha\sqrt{x}) \\
&\sim \frac{2M}{2\alpha\sqrt{x}}
\end{aligned}$$

where $\Psi$ is the Gaussian cumulative function and where we used the asymptotic approximation $\Psi(x) \sim \frac{e^{-x^2/2}}{x}$ for large $x$. Using this lower bound and the upper bound on $\phi$, there exists $x_0, k > 0$ such that for $x > x_0$, we have $x - \frac{V[\phi](x)}{V[\phi'](x)} \geq x - k\sqrt{x} \to \infty$ which concludes the proof. □

**Proposition 3** (Convergence rate for smooth activations)**.** *Let $\phi \in \mathcal{A}$ such that $\phi$ non-linear (i.e. $\phi^{(2)}$ is non-identically zero). Then, on the EOC, we have $1 - c^l \sim \frac{\beta_q}{l}$ where $\beta_q = \frac{2\mathbb{E}[\phi'(\sqrt{q}Z)^2]}{q\mathbb{E}[\phi''(\sqrt{q}Z)^2]}$.*

*Proof.* We first prove that $\lim_{l\to\infty} c^l = 1$ on the EOC. Let $x \in [0, 1)$ and $u_2(x) := xZ_1 + \sqrt{1-x^2}Z_2$, we have

$$\begin{aligned}
f'(x) &= \sigma_w^2\mathbb{E}[\phi'(\sqrt{q}Z_1)\phi'(\sqrt{q}u_2(x))] \\
&\leq \sigma_w^2(\mathbb{E}[\phi'(\sqrt{q}Z_1)^2])^{1/2}(\mathbb{E}[\phi'(\sqrt{q}u_2(x))^2])^{1/2} \\
&= 1
\end{aligned}$$

where we have used Cauchy Schwartz inequality and the fact the $\sigma_w^2 = \frac{1}{\mathbb{E}[\phi'(\sqrt{q}Z)^2]}$. Moreover, the equality holds if and only if there exists a constant $s$ such that $\phi'(\sqrt{q}(xz_1 + \sqrt{1-x^2}z_2)) = s\phi'(\sqrt{q}z_1)$ for almost any $z_1, z_2 \in \mathbb{R}$, which is equivalent to having $\phi'$ equal to a constant almost everywhere on $\mathbb{R}$, hence $\phi$ is linear and $q$ does not exists. This proves that for all $x \in [0, 1)$, $f'(x) < 1$. Integrating both sides between $x$ and 1 yields $f(x) > x$ for all $x \in [0, 1)$. Therefore $c^l$ is non-decreasing and converges to the fixed point of $f$ which is 1.

Now we want to prove that $f$ admits a Taylor expansion near 1. It is easy to do that if $\phi \in \mathcal{D}_g^3$. Indeed, using the conditions on $\phi$, we can easily see that $f$ has a third derivative at 1 and we have

$$\begin{aligned}
f'(1) &= \sigma_w^2\mathbb{E}[\phi'(\sqrt{q}Z)^2] \\
f''(1) &= \sigma_w^2 q\mathbb{E}[\phi''(\sqrt{q}Z)^2].
\end{aligned}$$

A Taylor expansion near 1 yields

$$\begin{aligned} f(x) &= 1 + f'(1)(x-1) + \frac{(x-1)^2}{2} f''(1) + O((x-1)^3) \\ &= x + \frac{(x-1)^2}{\beta_q} + O((x-1)^3). \end{aligned}$$

The proof is a bit more complicated for general $\phi \in \mathcal{A}$. We prove the result when $\phi^{(2)}(x) = 1_{x<0} g_1(x) + 1_{x \geq 0} g_2(x)$. The generalization to the whole class is straightforward. Let us first show that there exists $g \in \mathcal{C}^1$ such that $f^{(3)}(x) = \frac{1}{\sqrt{1-x^2}} g(x)$. We have

$$\begin{aligned} f''(x) &= \sigma_w^2 q \mathbb{E}[\phi''(\sqrt{q} Z_1) \phi''(\sqrt{q} U_2(x))] \\ &= \sigma_w^2 q \mathbb{E}[\phi''(\sqrt{q} Z_1) 1_{U_2(x)<0} g_1(\sqrt{q} U_2(x))] + \sigma_w^2 q \mathbb{E}[\phi''(\sqrt{q} Z_1) 1_{U_2(x)>0} g_2(\sqrt{q} U_2(x))]. \end{aligned}$$

Let $G(x) = \mathbb{E}[\phi''(\sqrt{q} Z_1) 1_{U_2(x)<0} g_1(\sqrt{q} U_2(x))]$ then

$$\begin{aligned} G'(x) = \mathbb{E}[\phi''(\sqrt{q} Z_1)(Z_1 - \frac{x}{\sqrt{1-x^2}} Z_2) \delta_{U_2(x)=0} \frac{1}{\sqrt{1-x^2}} g_1(\sqrt{q} U_2(x))] \\ + \mathbb{E}[\phi''(\sqrt{q} Z_1) 1_{U_2(x)<0} \sqrt{q} (Z_1 - \frac{x}{\sqrt{1-x^2}} Z_2) g_1'(\sqrt{q} U_2(x))]. \end{aligned}$$

After simplification, it is easy to see that $G'(x) = \frac{1}{\sqrt{1-x^2}} G_1(x)$ where $G_1 \in \mathcal{C}^1$. By extending the same analysis to the second term of $f''$, we conclude that there exists $g \in \mathcal{C}^1$ such that $f^{(3)}(x) = \frac{1}{\sqrt{1-x^2}} g(x)$.

Let us now derive a Taylor expansion of $f$ near 1. Since $f^{(3)}$ is potentially non defined at 1, we use the change of variable $x = 1 - t^2$ to compensate this effect. Simple algebra shows that the function $t \to f(1-t^2)$ has a Taylor expansion near 0

$$f(1-t^2) = 1 - t^2 f'(1) + \frac{t^4}{2} f''(1) + O(t^5).$$

Therefore,

$$f(x) = 1 + (x-1) f'(1) + \frac{(x-1)^2}{2} f''(1) + O((x-1)^{5/2}).$$

Note that this expansion is weaker than the expansion when $\phi \in \mathcal{D}_g^3$.

Denote $\lambda_l := 1 - c^l$, we have

$$\lambda_{l+1} = \lambda_l - \frac{\lambda_l^2}{\beta_q} + O(\lambda_l^{5/2})$$

therefore,

$$\begin{aligned} \lambda_{l+1}^{-1} &= \lambda_l^{-1} (1 - \frac{\lambda_l}{\beta_q} + O(\lambda_l^{3/2}))^{-1} \\ &= \lambda_l^{-1} (1 + \frac{\lambda_l}{\beta_q} + O(\lambda_l^{3/2})) \\ &= \lambda_l^{-1} + \frac{1}{\beta_q} + O(\lambda_l^{1/2}). \end{aligned}$$

By summing (divergent series), we conclude that $\lambda_l^{-1} \sim \frac{l}{\beta_q}$. □

**Proposition 4.** *Let $\phi \in \mathcal{A}$ be a non-linear activation function such that $\phi(0) = 0$, $\phi'(0) \neq 0$. Assume that $V[\phi]$ is non-decreasing and $V[\phi'])$ is non-increasing, and let $\sigma_{max} > 0$ be defined as in Proposition 2. Define the gradient with respect to the $l^{th}$ layer by $\frac{\partial E}{\partial y^l} = (\frac{\partial E}{\partial y_i^l})_{1 \leq i \leq N_l}$ and let $\tilde{Q}_{ab}^l = \mathbb{E}[\frac{\partial E}{\partial y_a^l}^T \frac{\partial E}{\partial y_b^l}]$ denote the covariance matrix of the gradients during backpropagation. Recall that $\beta_q = \frac{2\mathbb{E}[\phi'(\sqrt{q}Z)^2]}{q\mathbb{E}[\phi''(\sqrt{q}Z)^2]}$.*

*Then, for any $\sigma_b < \sigma_{max}$, by taking $(\sigma_b, \sigma_w) \in EOC$ we have*

- $\sup_{x \in [0,1]} |f(x) - x| \leq \frac{1}{\beta_q}$
- *For $l \geq 1$, $|\frac{\mathrm{Tr}(\tilde{Q}_{ab}^l)}{\mathrm{Tr}(\tilde{Q}_{ab}^{l+1})} - 1| \leq \frac{2}{\beta_q}$*

*Moreover, we have*

$$\lim_{\substack{\sigma_b \to 0 \\ (\sigma_b, \sigma_w) \in EOC}} \beta_q = \infty.$$

To prove this result, let us first prove a more general result.

**Proposition 5** (How close is $f$ to the identity function?)**.** *Let $\phi \in \mathcal{D}^2(\mathbb{R}, \mathbb{R}) - \{0\}$ and $(\sigma_b, \sigma_w) \in D_{\phi,var}$ with $q$ the corresponding limiting variance. Then,*

$$\sup_{x \in [0,1]} |f(x) - x| \leq |\sigma_w^2 \mathbb{E}[\phi'(\sqrt{q}Z)^2] - 1| + \frac{\sigma_w^2}{2} q\mathbb{E}[\phi''(\sqrt{q}Z)^2]$$

*Proof.* Using a second order Taylor expansion, we have for all $s \in [0,1]$

$$|f(x) - f(1) - f'(1)(x-1)| \leq \frac{(1-x)^2}{2} \sup_{\theta \in [0,1]} |f''(\theta)|.$$

We have $f(1) = 1$. Therefore $|f(x) - x| \leq (1-x)|f'(1) - 1| + \frac{(1-x)^2}{2} \sup_{\theta \in [0,1]} |f''(\theta)|$.
For $\theta \in [0,1]$, we have

$$\begin{aligned} f''(\theta) &= \sigma_w^2 q\mathbb{E}[\phi''(\sqrt{q}Z_1)\phi''(\sqrt{q}U_2(\theta))] \\ &\leq \sigma_w^2 q\mathbb{E}[\phi''(\sqrt{q}Z)^2] \\ &= \frac{\sigma_w^2}{2} q\mathbb{E}[\phi''(\sqrt{q}Z)^2] \end{aligned}$$

using Cauchy-Schwartz inequality. □

As a result, for $\phi \in \mathcal{D}^2(\mathbb{R}, \mathbb{R}) - \{0\}$ and $(\sigma_b, \sigma_w) \in EOC$ with $q$ the corresponding limiting variance, we have

$$\sup_{x \in [0,1]} |f(x) - x| \leq \frac{q\mathbb{E}[\phi''(\sqrt{q}Z)^2]}{2\mathbb{E}[\phi'(\sqrt{q}Z)^2]} = \frac{1}{\beta_q}$$

which is the first result of Proposition 4.

Now let us prove the second result for gradient backpropagation, we show that under some assumptions, our results of forward information propagation generalize to the back-propagation of the gradients. Let us first recall the results in (Schoenholz et al., 2017) (we use similar notations hereafter).

Let $E$ be the loss we want to optimize. The backpropagation process is given by the equations

$$\frac{\partial E}{\partial W_{ij}^l} = \delta_i^l \phi(y_j^{l-1})$$

$$\delta_i^l = \frac{\partial E}{\partial y_i^l} = \phi'(y_i^l) \sum_{j=1}^{N_{l+1}} \delta_j^{l+1} W_{ji}^{l+1}.$$

Although $\delta_i^l$ is non Gaussian (unlike $y_i^l$), knowing how $\tilde{q}_a^l = \mathbb{E}[(\delta_i^l)^2]$ changes back through the network will give us an idea about how the norm of the gradient changes. Indeed, following this approach, and using the approximation that the weights used during forward propagation are independent from those used for backpropagation, (Schoenholz et al., 2017) showed that

$$\tilde{q}_a^l = \tilde{q}_a^{l+1} \frac{N_{l+1}}{N_l} \chi_1$$

where $\chi_1 = \sigma_w^2 \mathbb{E}[\phi'(\sqrt{q}Z)^2]$.

Considering a constant width network, authors concluded that $\chi_1$ controls also the depth scales of the gradient norm, i.e. $\tilde{q}_a^l = \tilde{q}_a^L e^{-(L-l/\xi_\Delta)}$ where $\xi_\Delta^{-1} = -\log(\chi_1)$. So in the ordered phase, gradients can propagate to a depth of $\xi_\Delta$ without being exponentially small, while in the chaotic phase, gradient explode exponentially. On the EOC ($\chi_1 = 1$), the depth scale is infinite so the gradient information can also propagate deeper without being exponentially small.
The following result shows that our previous analysis on the EOC extends to the backpropagation of gradients, and that we can make this propagation better by choosing a suitable activation function and an initialization on the EOC. We use the following approximation to ease the calculations: the weights used in forward propagation are independent from those used in backward propagation.

**Proposition 6** (Better propagation for the gradient)**.** *Let $a$ and $b$ be two inputs and $(\sigma_b, \sigma_w) \in D_{\phi,var}$ with $q$ the limiting variance. We define the covariance between the gradients with respect to layer $l$ by $\tilde{q}_{ab}^l = \mathbb{E}[\delta_i^l(a)\delta_i^l(b)]$. Then, we have*

$$|\frac{\tilde{q}_{ab}^l}{\tilde{q}_{ab}^{l+1}} \times \frac{N_l}{N_{l+1}} - 1| \leq |\sigma_w^2 \mathbb{E}[\phi'(\sqrt{q}Z)^2] - 1| + (1 - c_{ab}^l)\sigma_w^2 q \mathbb{E}[\phi''(\sqrt{q}Z)^2] \to_{\sigma_b \to 0} 0.$$

*Proof.* We have

$$\begin{aligned}
\tilde{q}_{ab}^l &= \mathbb{E}[\delta_i^l(a)\delta_i^l(b)] \\
&= \mathbb{E}[\phi'(y_i^l(a))\phi'(y_i^l(b)) \sum_{j=1}^{N_{l+1}} \delta_j^{l+1}(a) W_{ji}^{l+1} \sum_{j=1}^{N_{l+1}} \delta_j^{l+1}(b) W_{ji}^{l+1}] \\
&= \mathbb{E}[\phi'(y_i^l(a))\phi'(y_i^l(b))] \times \mathbb{E}[\delta_j^{l+1}(a)\delta_j^{l+1}(b)] \times \mathbb{E}[\sum_{j=1}^{N_{l+1}} (W_{ji}^{l+1})^2] \\
&\approx \tilde{q}_{ab}^{l+1} \frac{N_{l+1}}{N_l} \sigma_w^2 \mathbb{E}[\phi'(\sqrt{q}Z_1)\phi'(\sqrt{q}U_2(c_{ab}^l))] \\
&= \tilde{q}_{ab}^{l+1} \frac{N_{l+1}}{N_l} f'(c_{ab}^l).
\end{aligned}$$

We conclude using the fact that $|f'(x) - 1| \leq |f'(1) - 1| + (1 - x)f''(1)$ □

The dependence in the width of the layer is natural since it acts as a scale for the covariance. We define the gradient with respect to the $l^{th}$ layer by $\frac{\partial E}{\partial y^l} = (\frac{\partial E}{\partial y_i^l})_{1 \leq i \leq N_l}$ and let $\tilde{Q}_{ab}^l = \mathbb{E}[\frac{\partial E}{\partial y_a^l}^T \frac{\partial E}{\partial y_b^l}]$ denote the covariance matrix of the gradients during backpropagation. Then, on the EOC, we have

$$|\frac{\mathrm{Tr}(\tilde{Q}_{ab}^l)}{\mathrm{Tr}(\tilde{Q}_{ab}^{l+1})} - 1| \leq (1 - c_{ab}^l) \frac{q\mathbb{E}[\phi''(\sqrt{q}Z)^2]}{\mathbb{E}[\phi'(\sqrt{q}Z)^2]} \leq \frac{2}{\beta_q}.$$

So again, the quantity $|\phi|_{EOC}$ controls the vanishing of the covariance of the gradients during backpropagation. This was expected because linear activation functions do not change the covariance of the gradients.

## B. Further theoretical results

### B.1. Results on the Edge of Chaos

The next lemma shows that under some conditions, the EOC does not include couples $(\sigma_b, \sigma_w)$ with small $\sigma_b > 0$.

**Lemma 5** (Trivial EOC)**.** *Assume there exists $M > 0$ such that $\mathbb{E}[\phi''(xZ)\phi(xZ)] > 0$ for all $x \in ]0, M[$. Then, there exists $\sigma > 0$ such that $EOC \cap ([0, \sigma) \times \mathbb{R}^+) = \{(0, \frac{1}{|\phi'(0)|})\}$. Moreover, if $M = \infty$ then $EOC = \{(0, \frac{1}{|\phi'(0)|})\}$.*

Activation functions that satisfy the conditions of Lemma 5 cannot be used with small $\sigma_b > 0$ (note that using $\sigma_b = 0$ would lead to $q = 0$ which is not practical for the training), therefore, the result of Proposition 4 do not apply in this case. However, as we will see hereafter, SiLU (a.k.a Swish) has a partial EOC, and still allows better information propagation (Proposition 3) compared to ReLU even if $\sigma_b$ not very small.

*Proof.* It is clear that $(0, \frac{1}{|\phi'(0)|}) \in EOC$. For $\sigma_b > 0$ we denote by $q$ the smallest fixed point of the function $\sigma_b^2 + \frac{V[\phi]}{V[\phi']}$ (which is supposed to be the limiting variance on the EOC). Using the condition on $\phi$ and the fact that $\lim_{\sigma_b \to 0} q = 0$, there exists $\sigma > 0$ such that for $\sigma_b < \sigma$ we have $\mathbb{E}[\phi''(\sqrt{q}Z)\phi(\sqrt{q}Z)] > 0$. Now let us prove that for $\sigma_b \in ]0, \sigma[$, the limiting variance does not satisfy the EOC equation.
Let $t_{max} = \sqrt{\sup_{x>0} |x - \frac{V[\phi]}{V[\phi']}|}$ and $\sigma_b \in ]0, \min(t_{max}, \sigma)[$. Recall that for all $x \geq 0$ we have that

$$F'(x) = \sigma_w^2(\mathbb{E}[\phi'(\sqrt{x}Z)^2] + \mathbb{E}[\phi''(\sqrt{x}Z)\phi(\sqrt{x}Z)])$$

Using $\sigma_w^2 = 1/V[\phi'](q)$ (EOC equation) we have that $F'(q) = 1 + \sigma_w^2\mathbb{E}[\phi''(\sqrt{q}Z)\phi(\sqrt{q}Z)]) > 1$. Therefore, the function $\sigma_b^2 + \frac{1}{V[\phi'](q)}V[\phi]$ crosses the identity in a point $\hat{q} < q$, hence $(\sigma_b, \sigma_w) \notin D_{\phi,var}$. Therefore, for any $\sigma_b \in ]0, \sigma[$, there is no $\sigma_w$ such that $(\sigma_b, \sigma_w) \in EOC$.

If $M = \infty$, the previous analysis is true for any $\sigma > 0$, by taking the limit $\sigma \to \infty$, we conclude. □

This is true for activations such as Shifted Softplus (a shifted version of Softplus in order to have $\phi(0) = 0$) and SiLU (a.k.a Swish).

**Corollary 1.** *$EOC_{SSoftplus} = \{(0, 2)\}$ and there exists $\sigma > 0$ such that $EOC_{SiLU} \cap ([0, \sigma[\times\mathbb{R}^+) = \{(0, 2)\}$*

*Proof.* let $s(x) = \frac{1}{1+e^{-x}}$ for all $x \in \mathbb{R}$ (sigmoid function).

1. Let $sp(x) = \log(1 + e^x) - \log(2)$ for $x \in \mathbb{R}$ (Shifted Softplus). We have $sp'(x) = s(x)$ and $sp''(x) = s(x)(1 - s(x))$. For $x > 0$ we have

$$\begin{aligned}
\mathbb{E}[sp''(xZ)sp(xZ)] &= \mathbb{E}[s(xZ)(1 - s(xZ))sp(xZ)] \\
&= \mathbb{E}[1_{Z>0}(s(xZ)(1 - s(xZ))sp(xZ))] + \mathbb{E}[1_{Z<0}(s(xZ)(1 - s(xZ))sp(xZ))] \\
&= \mathbb{E}[1_{Z>0}(s(xZ)(1 - s(xZ))sp(xZ))] + \mathbb{E}[1_{Z<0}(s(xZ)(1 - s(xZ))sp(-xZ))] \\
&= \mathbb{E}[1_{Z>0}(s(xZ)(1 - s(xZ))(sp(xZ) + sp(-xZ)))] > 0,
\end{aligned}$$

   where we have used the fact that $sp(y) + sp(-y) = \log(\frac{2+e^y+e^{-y}}{4}) > 0$ for all $y > 0$. We conclude using Lemma 5.

2. Let $si(x) = xs(x)$ (SiLU activation function, known also as Swish). We have $si'(x) = s(x) + xs(x)(1 - s(x))$ and $si''(x) = s(x)(1 - s(x))(2 + x(1 - 2s(x)))$. Using the same technique as for SSoftplus, we have for $x > 0$

$$\begin{aligned}
\mathbb{E}[si''(xZ)si(xZ)] &= \mathbb{E}[xZ \times s(xZ)^2 \times (1 - s(xZ))(2 + xZ(1 - 2))] \\
&= \mathbb{E}[1_{Z>0}G(xZ)],
\end{aligned}$$

   where $G(y) = ys(y)(1 - s(y))(2 + y(1 - 2s(y)))(2s(y) - 1)$. The only term that changes sign is $(2 + y(1 - 2s(y)))$. It is positive for small $y$ and negative for large $y$. We conclude that there $M > 0$ such that $\mathbb{E}[si''(xZ)si(xZ)] > 0$ for $x \in ]0, M[$.

□

### B.2. Beyond the Edge of Chaos

Can we make the distance between $f$ and the identity function small independently from the choice of $\sigma_b$? The answer is yes if we select the right activation function. Let us first define a semi-norm on $\mathcal{D}^2(\mathbb{R},\mathbb{R})$.

**Definition 7** (EOC semi-norm)**.** *The semi-norm $|.|_{EOC}$ is defined on $\mathcal{D}^2(\mathbb{R},\mathbb{R})$ by $|\phi|_{EOC} = \sup_{y\in\mathbb{R}^+} \frac{y\mathbb{E}[\phi''(\sqrt{y}Z)^2]}{\mathbb{E}[\phi'(\sqrt{y}Z)^2]}$. $|.|_{EOC}$ is a norm on the quotient space $\mathcal{D}^2(\mathbb{R},\mathbb{R})/\mathcal{L}(\mathbb{R})$ where $\mathcal{L}(\mathbb{R})$ is the space of linear functions.*

When $|\phi|_{EOC}$ is small, $\phi$ is close to a linear function, which implies that the function $\frac{V[\phi]}{V[\phi']}$ defined on $\mathbb{R}^+$ is close to the identity function. Thus, for a fixed $\sigma_b$, we expect $q$ to become arbitrarily big when $|\phi|_{EOC}$ goes to zero.

**Lemma 2.1.** *Let $(\phi_n)_{n\in\mathbb{N}}$ be a sequence of functions such that $\lim_{n\to\infty}|\phi_n|_{EOC} = 0$. Let $\sigma_b > 0$ and assume that for all $n \in \mathbb{N}$ there exists $\sigma_{w,n}$ such that $(\sigma_w, \sigma_{w,n}) \in EOC$. Let $q_n$ be the limiting variance. Then $\lim_{n\to\infty} q_n = \infty$*

*Proof.* The proof is straightforward knowing that $f(0) \leq \frac{1}{2}|\phi_n|_{EOC}$, which implies that $\frac{\sigma_b^2}{q} \leq \frac{1}{2}|\phi_n|_{EOC}$. □

**Corollary 2.1.** *Let $\phi \in \mathcal{D}^2(\mathbb{R},\mathbb{R}) - \{0\}$ and $(\sigma_b, \sigma_w) \in EOC$ with $q$ the corresponding limiting variance. Then,*

$$\sup_{x\in[0,1]} |f(x) - x| \leq \frac{1}{2}|\phi|_{EOC}.$$

Corollary 2.1 shows that by taking an activation function $\phi$ such that $|\phi|_{EOC}$ is small and by initializing the network on the EOC, the correlation function is close to the identity function, i.e., the signal propagates deeper through the network. However, note that there is a trade-off to take in account here: we loose expressiveness by taking $|\phi|_{EOC}$ too small, because this would imply that $\phi$ is close to a linear function. So there is a trade-off between signal propagation and expressiveness We check this finding with activation functions of the form $\phi_\alpha(x) = x + \alpha\text{Tanh}(x)$. Indeed, we have $|\phi_\alpha|_{EOC} \leq \alpha^2 \sup_{y\in\mathbb{R}^+} \mathbb{E}[\text{Tanh}''(\sqrt{x}Z)^2] \to_{\alpha\to 0} 0$. So by taking small $\alpha$, we would theoretically provide deeper signal propagation. However, note that we loose expressiveness as $\alpha$ goes to zero because $\phi_\alpha$ becomes closer to the identity function. So There is also a trade-off here. The difference with Proposition 4 is that here we can compensate the expressiveness issue by adding more layers (see e.g. (Montufar et al., 2014) who showed that expressiveness grows exponentially with depth).

## C. Experiments

### C.1. Training with RMSProp

For RMSProp, the learning rate $10^{-5}$ is nearly optimal for networks with depth $L \leq 200$ (for deeper networks, $10^{-6}$ gives better results). This learning rate was found by a grid search with exponential step of size 10.
Figure 7 shows the training curves of ELU, ReLU and Tanh on MNIST for a network with depth 200 and width 300. Here also, ELU and Tanh perform better than ReLU. This confirms that the result of Proposition 3 is independent of the training algorithm.
ELU has faster convergence than Tanh. This could be explained by the saturation problem of Tanh.

### C.2. Training with activation $\phi_\alpha(x) = x + \alpha\textbf{Tanh}(x)$

As we have already mentioned, $\phi_\alpha$ satisfies all conditions of Proposition 3. Therefore, we expect it to perform at least better than ReLU for deep neural networks. Figure 8 shows the training curve for width 300 and depth 200 with different activation functions. $\phi_{0.5}$ has approximately similar performance as ELU and better than Tanh and ReLU. Note that $\phi_\alpha$ does not suffer form saturation of the gradient, which could explain why it performs better than Tanh.

### C.3. Impact of $\phi''(0)$

Since we usually take $\sigma_b$ small on the EOC, then having $\phi''(0) = 0$ would make the coefficient $\beta_q$ even bigger. We test this result on SiLU (a.k.a Swish) for depth 70. SiLU is defined by

$$\phi_{SiLU}(x) = x\ \text{sigmoid}(x)$$

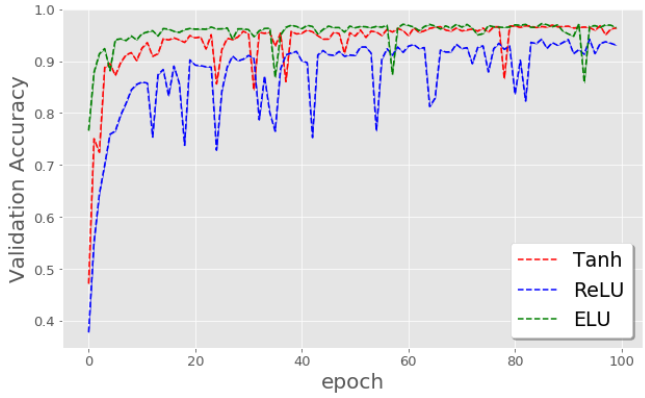



*Figure 7.* 100 epochs of the training curves of ELU, ReLU and Tanh networks of depth 200 and width 300 on MNIST with RMSProp

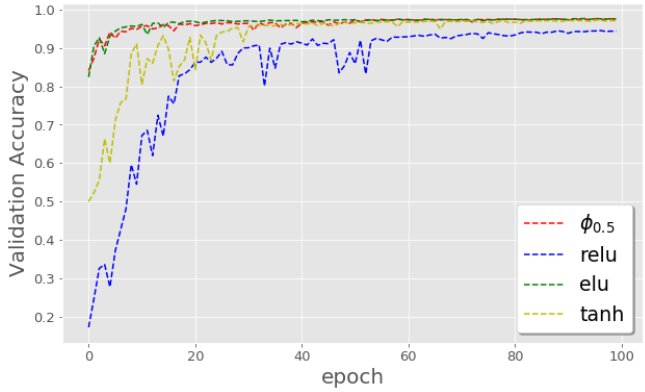



*Figure 8.* 100 epochs of the training curves of ELU, ReLU, Tanh and $\phi_{0.5}$ networks of depth 200 and width 300 on MNIST with SGD

we have $\phi''(0) = 1/2$. consider a modified SiLU (MSiLU) defined by

$$\phi_{MSiLU}(x) = x \text{ sigmoid}(x) + (e^{-x^2} - 1)/4$$

We have $\phi''_{MSiLU}(0) = 0$.
Figure 9 shows the the training curves (test accuracy) of SiLU and MSiLU on MNIST with SGD. MSiLU performs better than SiLU, expecially at the beginning of the training.

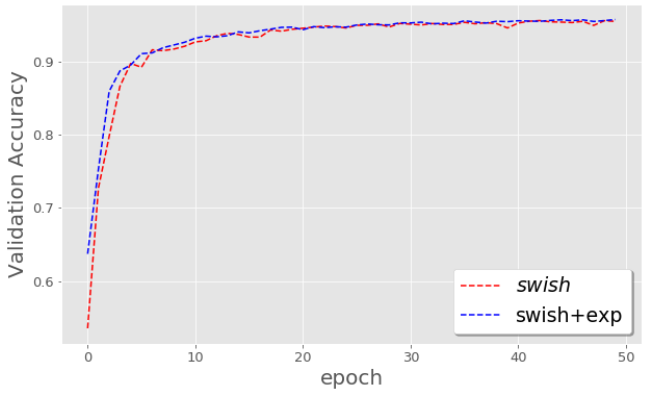



*Figure 9.* 50 epochs of the training curves of SiLU and MSiLU on MNIST with SGD

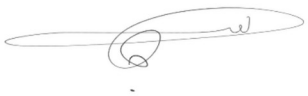
## Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

| | |
|---|---|
| Title of Paper | On the Impact of the Activation Function on Deep Neural Networks Training |
| Publication Status | ■ Published □ Accepted for Publication<br>□ Submitted for Publication □ Unpublished and unsubmitted work written in a manuscript style |
| Publication Details | Hayou, S., Doucet, A., and Rousseau, J. (2019). On the Impact of the Activation Function on Deep Neural Networks Training. Proceedings of the 36th International Conference of Machine Learning (ICML 2019). |

### Student Confirmation

| | | | |
|---|---|---|---|
| Student Name: | Soufiane Hayou | | |
| Contribution to the Paper | I worked on the theory and proofs behind this paper. I also worked on the experiments. During our weekly meetings, my supervisors contributed to this work by providing valuable insights and helpful remarks. They also contributed a lot to the writing of the draft, checking the proofs, and proof-reading. | | |
| Signature |  | Date | 21/01/2021 |

### Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

| | | |
|---|---|---|
| Supervisor name and title: Arnaud Doucet | | |
| Supervisor comments | | |
| Signature | Date | |

This completed form should be included in the thesis, at the end of the relevant chapter.

# 3

# Neural Tangent Kernel for Deep Neural Networks

# 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.

[1]Department of Statistics, University of Oxford, Oxford, United kingdom. Correspondence to: Soufiane Hayou <soufiane.hayou@stats.ox.ac.uk>.

**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

*Table 1.* "Does the model learn?". We train a FeedForward Neural Network on MNIST using both standard SGD training and NTK training defined in section 2. For Shallow networks, both SGD and NTK yield good performance (See section 5). However, for Deep networks, the NTK training yields trivial accuracy of around $\sim 10\%$ for any initialization scheme.

| | | Initialization on the Edge of Chaos | Other Initialization |
|---|---|---|---|
| Shallow Network (depth $L = 3$) | NTK | ✓ | ✓ |
| | SGD | ✓ | ✓ |
| Medium Network (depth $L = 30$) | NTK | ✓ | ✗ |
| | SGD | ✓ | ✗ |
| Deep Network (depth $L = 300$) | NTK | ✗ | ✗ |
| | SGD | ✓ | ✗ |

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 $s : \mathbb{R}^{n_L} \to \mathbb{R}^o$ 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 $\theta_t$ the value of $\theta$ at time $t$ and $f_t(x) = f(x, \theta_t)$. Let $\mathcal{D} = (x_i, z_i)_{1\leq i\leq N}$ be the data set, and let $\mathcal{X} = (x_i)_{1\leq i\leq N}$, $\mathcal{Z} = (z_j)_{1\leq j\leq N}$ 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 $x, x' \in \mathbb{R}^d$

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

• **The NTK regime (Infinite width):** In the case of an FFNN, Jacot et al. (2018) proved that, with GD, the kernel $K^L_{\theta_t}$ 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

$$f_t(\mathcal{X}) = e^{-\frac{t}{N}\hat{K}^L} f_0(\mathcal{X}) + (I - e^{-\frac{t}{N}\hat{K}^L})\mathcal{Z}, \quad (1)$$

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

$$f_t(x) = f_0(x) + \gamma(x, \mathcal{X})(I - e^{-\frac{1}{N}\hat{K}^L t})(\mathcal{Z} - f_0(\mathcal{X})), \quad (2)$$

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 $\hat{K}^L$. 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 $\mathcal{O}(L\sqrt{z^T(\hat{K}^L)^{-1}z/N})$ 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 $a_1, ..., a_N \in \mathbb{R}$ such that for all $x \in \mathbb{R}^d$, $f_t(x) - f_0(x) = \sum_{i=1}^N a_i K^L(x_i, x)$, 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

$$y_i^1(x) = \frac{\sigma_w}{\sqrt{d}}\sum_{j=1}^{d} w_{ij}^1 x_j + \sigma_b b_i^1$$
$$y_i^l(x) = \frac{\sigma_w}{\sqrt{n_{l-1}}}\sum_{j=1}^{n_{l-1}} w_{ij}^l \phi(y_j^{l-1}(x)) + \sigma_b b_i^l, \quad l \geq 2. \tag{3}$$

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

$$y_{i,\alpha}^1(x) = \frac{\sigma_w}{\sqrt{v_1}}\sum_{j=1}^{n_0}\sum_{\beta\in ker_1} w_{i,j,\beta}^1 x_{j,\alpha+\beta} + \sigma_b b_i^1$$
$$y_{i,\alpha}^l(x) = \frac{\sigma_w}{\sqrt{v_l}}\sum_{j=1}^{n_{l-1}}\sum_{\beta\in ker_l} w_{i,j,\beta}^l \phi(y_{j,\alpha+\beta}^{l-1}(x)) + \sigma_b b_i^l, \tag{4}$$

where $i \in [1 : n_l]$ 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, $ker_l = [-k : k]$ is a filter with size $2k+1$ and $v_l = n_{l-1}(2k+1)$. Here, $w^l \in \mathbb{R}^{n_l \times n_{l-1} \times (2k+1)}$. We assume periodic boundary conditions, which result in having $y_{i,\alpha}^l = y_{i,\alpha+M}^l = y_{i,\alpha-M}^l$, and similarly for $l = 0$, $x_{i,\alpha+M_0} = x_{i,\alpha} = x_{i,\alpha-M_0}$. For the sake of simplification, we only consider the case of 1D CNN, the generalization to a $m$D CNN for $m \in \mathbb{N}$ 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 $[x, x']_{\alpha,\alpha'} = \sum_{j=1}^{n_0}\sum_{\beta\in ker_0} x_{j,\alpha+\beta} x'_{j,\alpha'+\beta}$.

We initialize the model randomly with $w_{ij}^l, b_i^l \overset{\text{iid}}{\sim} \mathcal{N}(0, 1)$, 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 $(y_i^l(.))_{i,l}$ 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. $q_{\alpha,\alpha'}^l(x, x')$ the covariance between $y_1^l(x)$ and $y_1^l(x')$ resp. $y_{1,\alpha}^l(x)$ and $y_{1,\alpha'}^l(x')$. We define the correlations $c^l(x, x')$ and $c_{\alpha,\alpha'}^l(x, x')$ similarly. For FFNN, we have that

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

and similarly for CNN we have

$$q_{\alpha,\alpha'}^1(x, x') = \sigma_b^2 + \frac{\sigma_w^2}{n_0(2k+1)}[x, x']_{\alpha,\alpha'}.$$

For $\epsilon \in (0,1)$, we define the set $B_\epsilon$ by:

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

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*

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

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

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 $\alpha, \alpha' \in [0 : M-1]$*

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

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

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

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

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 $K^L$ 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 $\hat{K}^L$ 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 $x, x' \in \mathcal{X}$, $q^1_{\alpha,\alpha'}(x,x')$ 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 $\epsilon \in (0,1)$*

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

*where*
*• if $\phi = ReLU$, then $AK^\infty(x,x') = \frac{\sigma_w^2 \|x\|\, \|x'\|}{d}(1 - (1-\lambda)\mathbb{1}_{x \neq x'})$ with $\lambda = 1/4$.*
*• if $\phi = Tanh$, then $AK^\infty(x,x') = q(1-(1-\lambda)\mathbb{1}_{x \neq x'})$ 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 $\mathcal{O}(\frac{\text{polylog}L}{L})$, 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 $x, x' \in \mathbb{R}^d$*

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

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

$$K_{res}^l(x,x') = K_{res}^{l-1}(x,x')(\dot{q}^l(x,x') + 1) + q^l(x,x').$$

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 $x, x' \in \mathbb{R}^d$*

$$K_{(i,\alpha),(i',\alpha')}^{res,1}(x,x') = \delta_{ii'}\Big(\frac{\sigma_w^2}{n_0(2k+1)}[x,x']_{\alpha,\alpha'} + \sigma_b^2\Big).$$

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

$$K_{\alpha,\alpha'}^{res,l} = K_{\alpha,\alpha'}^{res,l-1} + \frac{1}{2k+1}\sum_{\beta} \Psi_{\alpha+\beta,\alpha'+\beta}^{l-1},$$

*where $\Psi_{\alpha,\alpha'}^l = \dot{q}_{\alpha,\alpha'}^l K_{\alpha,\alpha'}^{res,l} + \hat{q}_{\alpha,\alpha'}^l$.*

The additional terms $K_{res}^{l-1}(x,x')$ (resp. $K_{\alpha,\alpha'}^{res,l-1}$) 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 $\bar{K}^L = K^L/\alpha_L$ 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*

$$y^l(x) = y^{l-1}(x) + \mathcal{F}(w^l, y^{l-1}(x)), \quad l \geq 2, \qquad (5)$$

*where $\mathcal{F}$ is either a convolutional or dense layer (equations (3) and (4)) with ReLU activation. Let $K^L_{res}$ be the corresponding NTK, and $\bar{K}^L_{res} = K^L_{res}/\alpha_L$ (Normalized NTK) with $\alpha_L = L(1+\frac{\sigma_w^2}{2})^{L-1}$. Then, we have*

$$\sup_{x\in E} |\bar{K}^L_{res}(x,x) - \bar{K}^\infty_{res}(x,x)| = \Theta(L^{-1}).$$

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

$$\sup_{x,x'\in B_\epsilon} \left|\bar{K}^L_{res}(x,x') - \bar{K}^\infty_{res}(x,x')\right| = \Theta(\log(L)L^{-1}),$$

*where $\bar{K}^\infty_{res}(x,x') = \frac{\sigma_w^2\|x\|\,\|x'\|}{d}(1-(1-\lambda)\mathbb{1}_{x\neq x'})$.*
*All results hold for ResNet with Convolutional layers under Assumption 1.*

The proof techniques used in theorem 2 are similar to those used in the proof of theorem 1. Details are provided in the appendix.

Theorem 2 shows that the NTK of a ReLU ResNet explodes exponentially w.r.t $L$. However, the normalized kernel $\bar{K}^L_{res} = K^L_{res}(x,x')/\alpha_L$ converges to a limiting kernel $\bar{K}^\infty_{res}$ 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 the normalized NTK converges to a trivial kernel, which means that, even with ResNet, NTK training would fail at some point as we increase the depth.

The term $\alpha_L$ in the residual NTK might cause numerical stability issues for NTK training, and the triviality of the limiting kernel yields a trivial NTK regime solution (recall that $f_t - f_0$ belongs to the RKHS of the NTK; see section 2). It turns out that we can improve the performance of NTK training of ResNets with a simple scaling of the ResNet blocks.

**Proposition 2** (Scaled ResNet)**.** *Consider a ResNet satisfying*

$$y^l(x) = y^{l-1}(x) + \frac{1}{\sqrt{l}}\mathcal{F}(w^l, y^{l-1}(x)), \quad l \geq 2, \qquad (6)$$

*where $\mathcal{F}$ is either a convolutional or dense layer ((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 $\Theta(\log(L)^{-1})$.*

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 $L^{1+\frac{\sigma_w^2}{2}}$ instead of $L(1+\frac{\sigma_w^2}{2})^{L-1}$; second, it drastically slows down the convergence rate to the limiting (trivial) $\bar{K}^\infty_{res}$. 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 ResNet. 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 $\mathbb{S}^{d-1} = \{x \in \mathbb{R}^d : \|x\|_2 = 1\}$. On the sphere $\mathbb{S}^{d-1}$, 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 $x, x' \in \mathbb{S}^{d-1}$. This kernel type is known to be diagonalizable on the sphere $\mathbb{S}^{d-1}$ and the eigenfunctions are the so-called Spherical Harmonics of $\mathbb{S}^{d-1}$. 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 $\mathbb{S}^{d-1}$)**.** *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 ($\bar{K}^L_{res}$) for a ResNet with $L$ layers (Fully Connected). Then, for all $L \geq 1$, there exists $(\mu^L_k)_{k\geq}$ 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 $0 < \mu^\infty_0 = \lim_{L\to\infty}\mu^L_0 < \infty$, and for all $k \geq 1$, $\lim_{L\to\infty}\mu^L_k = 0$.*

The proof of theorem 3 is based on a result from spectral theory analysis. The limiting eigenvalues are obtained by a simple application of the dominated convergence theorem.

Theorem 3 shows that in the limit of large $L$, the kernel $\kappa^L$ becomes close to the trivial kernel $\kappa^\infty\ (x,x') \mapsto \mu^\infty_0 Y_{0,0}(x)Y_{0,0}(x')$, where $Y_{0,0}$ is the constant function in

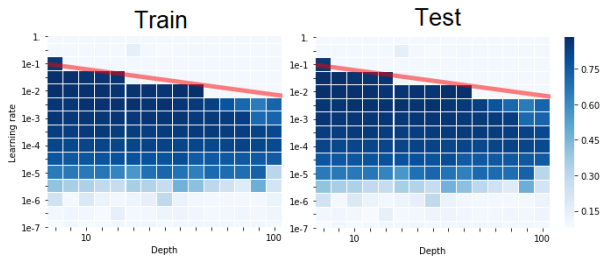



*Figure 1.* Train/Test accuracy of an FFNN with ReLU activation on Fashion MNIST dataset for different depths and learning rates, trained for 10 epochs. The plot is in log-log scale.

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

$$df_t(\mathcal{X}) = -\frac{1}{N}\hat{K}^L(f_t(\mathcal{X}) - \mathcal{Z})dt. \tag{7}$$

The GD update with learning rate $\eta$ is given by

$$f_{t+1}(\mathcal{X}) = (I - \frac{\eta}{N}\hat{K}^L)f_t(\mathcal{X}) - \frac{\eta}{N}\hat{K}^L\mathcal{Z}). \tag{8}$$

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

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

where $\mu_{\max}$ is the largest eigenvalue.

For an FFNN (or a CNN with Assumption 1) initialized on the EOC, as $L$ grows we have that $\hat{K}^L = qL((1-\lambda)I + \lambda U) + \mathcal{O}(\log(L))$ (Theorem 1). Therefore, for large $L$ and $N$, we have that $\mu_{\max}(\frac{1}{N}\hat{K}^L) \sim q\lambda L$. 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 LRs 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 and depths are almost uncorrelated.[1]

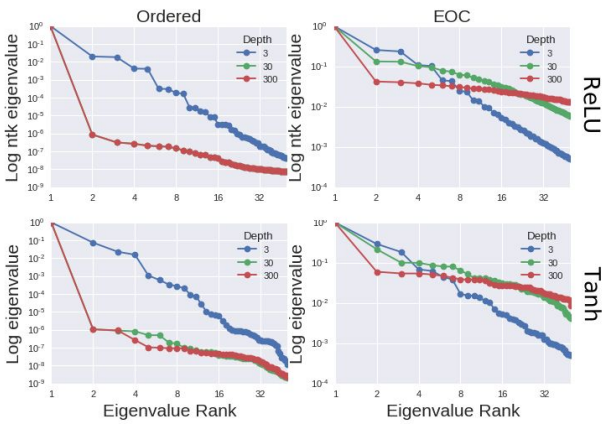



*Figure 2.* Normalized eigenvalues of $K_L$ on the 2D sphere for an FFNN with different initializations, activations, and depths.

## 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 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 intialization. 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.

*Table 2.* Test accuracy for varying architectures and depths on MNIST and CIFAR10 dataset. We show test accuracy after 100 training epochs for $L \in \{3, 30\}$ and 160 epochs for $L = 300$.

| | MNIST | | | | CIFAR10 | | | |
|---|---|---|---|---|---|---|---|---|
| | NTK Training | | SGD Training | | NTK Training | | SGD Training | |
| | EOC | Ordered | EOC | Ordered | EOC | Ordered | EOC | Ordered |
| **L=3** | | | | | | | | |
| FFNN-ReLU | $96.64_{\pm 0.11}$ | $96.57_{\pm 0.12}$ | $97.05_{\pm 0.27}$ | $97.11_{\pm 0.31}$ | $48.13_{\pm 0.10}$ | $48.45_{\pm 0.14}$ | $55.13_{\pm 0.23}$ | $54.10_{\pm 0.12}$ |
| FFNN-Tanh | $95.34_{\pm 1.04}$ | $96.32_{\pm 0.41}$ | $97.19_{\pm 0.11}$ | $97.03_{\pm 0.29}$ | $48.32_{\pm 0.15}$ | $48.10_{\pm 0.10}$ | $56.13_{\pm 0.34}$ | $54.10_{\pm 0.23}$ |
| CNN-ReLU | $97.13_{\pm 0.31}$ | $97.23_{\pm 0.22}$ | $98.95_{\pm 0.12}$ | $98.89_{\pm 0.18}$ | $49.11_{\pm 0.16}$ | $42.76_{\pm 3.32}$ | $60.23_{\pm 0.45}$ | $59.05_{\pm 0.15}$ |
| V-ResNet | $96.73_{\pm 0.05}$ | $96.71_{\pm 0.16}$ | $97.19_{\pm 0.23}$ | $97.12_{\pm 0.14}$ | $47.82_{\pm 0.73}$ | $48.01_{\pm 0.20}$ | $54.40_{\pm 0.24}$ | $54.28_{\pm 0.33}$ |
| **L=30** | | | | | | | | |
| FFNN-ReLU | $96.95_{\pm 0.22}$ | — | $97.55_{\pm 0.09}$ | — | $48.32_{\pm 0.10}$ | — | $56.10_{\pm 0.41}$ | — |
| FFNN-Tanh | $97.30_{\pm 0.15}$ | — | $97.87_{\pm 0.17}$ | — | $48.40_{\pm 0.12}$ | — | $57.39_{\pm 0.08}$ | — |
| CNN-ReLU | $98.60_{\pm 0.13}$ | — | $99.02_{\pm 0.07}$ | — | $48.42_{\pm 0.10}$ | — | $75.39_{\pm 0.31}$ | — |
| V-ResNet | — | — | $98.17_{\pm 0.03}$ | $98.13_{\pm 0.08}$ | — | — | $57.09_{\pm 0.47}$ | $58.13_{\pm 0.18}$ |
| S-ResNet | $97.01_{\pm 0.10}$ | $97.11_{\pm 0.10}$ | $98.33_{\pm 0.10}$ | $98.26_{\pm 0.14}$ | $49.10_{\pm 0.15}$ | $50.01_{\pm 0.12}$ | $57.21_{\pm 0.43}$ | $57.51_{\pm 0.11}$ |
| **L=300** | | | | | | | | |
| FFNN-ReLU | — | — | $98.14_{\pm 0.12}$ | — | — | — | $30.25_{\pm 3.23}$ | — |
| FFNN-Tanh | — | — | $98.54_{\pm 0.18}$ | — | — | — | $58.25_{\pm 0.43}$ | — |
| CNN-ReLU | — | — | $99.43_{\pm 0.04}$ | — | — | — | $76.25_{\pm 0.21}$ | — |
| V-ResNet | — | — | $98.23_{\pm 0.09}$ | $98.19_{\pm 0.06}$ | — | — | $58.87_{\pm 0.44}$ | $59.25_{\pm 0.10}$ |
| S-ResNet | — | — | $98.40_{\pm 0.07}$ | $98.51_{\pm 0.08}$ | — | — | $60.86_{\pm 0.24}$ | $61.51_{\pm 0.18}$ |

**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 $L = 30$, 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±3.23 | 74.77±0.29 |
| | scaled | **56.38±2.54** | **76.14±0.98** |

**Does Scaled ResNet outperforms ResNet with SGD?** We train standard ResNet with depths 32, 50, and 104 on CIFAR100 with SGD. We use a decaying learning rate schedule; we start with 0.1 and divide by 10 after $n_e/2$ epochs, where $n_e$ is the total number of epochs; we scale again, by 10, after $n_e/4$ epochs. We use a batch size of 128, and we train the model with 160 epochs. Proposition 2 shows that the NTK of Scaled ResNet is more stable compared to the NTK of standard ResNet. Although this result is limited to NTK training, we investigate the impact of scaling on SGD training. Table 3 displays test accuracy for standard ResNet and scaled ResNet after 10 and 160 epochs; Scaled ResNet outperforms ResNet and converges faster. However, it is not clear whether this is linked to the NTK, or caused by something else. We leave this for future work.

## 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.

## 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. 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 $y^l : \mathbb{R}^{n_{l-1}} \to \mathbb{R}^{n_l}$, $n_0 = d$ 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 $s : \mathbb{R}^{n_L} \to \mathbb{R}^o$ 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 $\theta_t$ the value of $\theta$ at time $t$ and $f_t(x) = f(x, \theta_t) = (f_j(x, \theta_t), j \leq o)$. Let $D = (x_i, z_i)_{1\leq i\leq N}$ be the data set and let $\mathcal{X} = (x_i)_{1\leq i\leq N}$, $\mathcal{Z} = (z_j)_{1\leq j\leq N}$ be the matrices of input and output respectively, with dimension $d \times N$ and $o\times N$. For any function $g : \mathbb{R}^{d\times o} \to \mathbb{R}^k$, $k \geq 1$, we denote by $g(\mathcal{X}, \mathcal{Z})$ 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

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

where $\eta > 0$ is the learning rate.
Let $T > 0$ be the training time and $N_s = T/\eta$ 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

$$d\theta_t = -\nabla_\theta\mathcal{L}(\theta_t)dt. \tag{2}$$

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 $\mathcal{O}(\eta)$ (see Appendix). Equation (2) can be re-written as

$$d\theta_t = -\frac{1}{N}\nabla_\theta f(\mathcal{X}, \theta_t)^T\nabla_{z'}\ell(f(\mathcal{X}, \theta_t), \mathcal{Z})dt.$$

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

$$df_t(x) = -\frac{1}{N}\nabla_\theta f(x, \theta_t)\nabla_\theta f(\mathcal{X}, \theta_t)^T\nabla_{z'}\ell(f_t(\mathcal{X}), \mathcal{Z})dt. \tag{3}$$

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

$$\begin{aligned} K^L_{\theta_t}(x, x') &= \nabla_\theta f(x, \theta_t)\nabla_\theta f(x', \theta_t)^T \in \mathbb{R}^{o\times o} \\ &= \sum_{l=1}^L \nabla_{\theta^l} f(x, \theta_t)\nabla_{\theta^l} f(x', \theta_t)^T. \end{aligned} \tag{4}$$

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

$$K^L_{\theta_t}(\mathcal{X}, \mathcal{X}) = \begin{pmatrix} K^L_{\theta_t}(x_1, x_1) & \cdots & K^L_{\theta_t}(x_1, x_N) \\ K^L_{\theta_t}(x_2, x_1) & \cdots & K^L_{\theta_t}(x_2, x_N) \\ \vdots & \ddots & \vdots \\ K^L_{\theta_t}(x_N, x_1) & \cdots & K^L_{\theta_t}(x_N, x_N) \end{pmatrix}.$$

By applying (3) to the vector $\mathcal{X}$, one obtains

$$df_t(\mathcal{X}) = -\frac{1}{N} K^L_{\theta_t}(\mathcal{X}, \mathcal{X}) \nabla_{z'} \ell(f_t(\mathcal{X}), \mathcal{Z}) dt, \tag{5}$$

meaning that for all $j \leq N$

$$df_t(x_j) = -\frac{1}{N} K^L_{\theta_t}(x_j, \mathcal{X}) \nabla_{z'} \ell(f_t(\mathcal{X}), \mathcal{Z}) dt.$$

**Infinite width dynamics.** In the case of an FFNN, (Jacot et al., 2018) proved that, with GD, the kernel $K^L_{\theta_t}$ converges to a kernel $K^L$ which depends only on $L$ (number of layers) for all $t < T$ when $n_1, n_2, ..., n_L \to \infty$, where $T$ is an upper bound on the training time, under the technical assumption $\int_0^T ||\nabla_z \ell(f_t(\mathcal{X}, \mathcal{Z}))||_2 dt < \infty$ a.s. with respect to the initialization weights. The infinite width limit of the training dynamics is given by

$$df_t(\mathcal{X}) = -\frac{1}{N} K^L(\mathcal{X}, \mathcal{X}) \nabla_{z'} \ell(f_t(\mathcal{X}), \mathcal{Z}) dt, \tag{6}$$

We note hereafter $\hat{K}^L = K^L(\mathcal{X}, \mathcal{X})$. As an example, with the quadratic loss $\ell(z', z) = \frac{1}{2}||z' - z||^2$, (6) is equivalent to

$$df_t(\mathcal{X}) = -\frac{1}{N} \hat{K}^L (f_t(\mathcal{X}) - \mathcal{Z}) dt, \tag{7}$$

which is a simple linear model that has a closed-form solution given by

$$f_t(\mathcal{X}) = e^{-\frac{1}{N}\hat{K}^L t} f_0(\mathcal{X}) + (I - e^{-\frac{1}{N}\hat{K}^L t})\mathcal{Z}. \tag{8}$$

For general input $x \in \mathbb{R}^d$, we have

$$f_t(x) = f_0(x) + \gamma(x, \mathcal{X})(I - e^{-\frac{1}{N}\hat{K}^L t})(\mathcal{Z} - f_0(\mathcal{X})). \tag{9}$$

where $\gamma(x) = K^L(x, \mathcal{X}) K^L(\mathcal{X}, \mathcal{X})^{-1}$.

### 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 $(n_l)_{1 \leq l \leq L}$, 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

$$\begin{aligned} y^1_i(x) &= \frac{\sigma_w}{\sqrt{d}} \sum_{j=1}^{d} w^1_{ij} x_j + \sigma_b b^1_i \\ y^l_i(x) &= \frac{\sigma_w}{\sqrt{n_{l-1}}} \sum_{j=1}^{n_{l-1}} w^l_{ij} \phi(y^{l-1}_j(x)) + \sigma_b b^l_i, \quad l \geq 2. \end{aligned} \tag{10}$$

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

$$\begin{aligned} y^1_{i,\alpha}(x) &= \frac{\sigma_w}{\sqrt{v_1}} \sum_{j=1}^{n_0} \sum_{\beta \in ker_1} w^1_{i,j,\beta} x_{j,\alpha+\beta} + \sigma_b b^1_i \\ y^l_{i,\alpha}(x) &= \frac{\sigma_w}{\sqrt{v_l}} \sum_{j=1}^{n_{l-1}} \sum_{\beta \in ker_l} w^l_{i,j,\beta} \phi(y^{l-1}_{j,\alpha+\beta}(x)) + \sigma_b b^l_i, \end{aligned} \tag{11}$$

  where $i \in [1 : n_l]$ 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, $ker_l = [-k : k]$ is a filter with size $2k+1$ and $v_l = n_{l-1}(2k+1)$. Here, $w^l \in \mathbb{R}^{n_l \times n_{l-1} \times (2k+1)}$. We assume periodic boundary conditions, which results in having $y^l_{i,\alpha} = y^l_{i,\alpha+M} = y^l_{i,\alpha-M}$ and similarly for $l = 0$, $x_{i,\alpha+M_0} = x_{i,\alpha} = x_{i,\alpha-M_0}$. 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, controling 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

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

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 $y_i^l(x)$ is a Gaussian variable for any input $x$. This gives an error of order $\mathcal{O}(1/\sqrt{n_{l-1}})$ (standard Monte Carlo error). More generally, an approximation of the random process $y_i^l(.)$ 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 $x, x' \in \mathbb{R}^d$

$$\begin{aligned}\kappa^l(x, x') &= \mathbb{E}[y_i^l(x) y_i^l(x')] \\ &= \sigma_b^2 + \sigma_w^2 \mathbb{E}[\phi(y_i^{l-1}(x))\phi(y_i^{l-1}(x'))] \\ &= \sigma_b^2 + \sigma_w^2 \Psi_\phi(\kappa^{l-1}(x,x), \kappa^{l-1}(x,x'), \kappa^{l-1}(x',x')),\end{aligned}$$

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 $\mathcal{O}(1/\sqrt{n_{l-1}})$. The kernel in this case depends on the choice of the neurons in the channel and is given by

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

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

$$\hat{\kappa}^l(x,x') = \sigma_b^2 + \sigma_w^2\frac{1}{N^2}\sum_{\alpha,\alpha'} F_\phi(\kappa^{l-1}_{\alpha,\alpha'}(x,x), \kappa^{l-1}_{\alpha,\alpha'}(x,x'), \kappa^{l-1}_{\alpha,\alpha'}(x',x')).$$

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

Note that our analysis only requires the approximation that, in the infinite width limit, for any two inputs $x, x'$, the variables $y^l_i(x)$ and $y^l_i(x')$ are Gaussian with covariance $\kappa^l(x,x')$ for FFNN, and $y^l_{i,\alpha}(x)$ and $y^l_{i,\alpha'}(x')$ are Gaussian with covariance $\kappa^l_{\alpha,\alpha'}(x,x')$ for CNN. We do not need the much stronger approximation that the process $y^l_i(x)$ ($y^l_{i,\alpha}(x)$ for CNN) is a Gaussian process.

**Residual Neural Networks.** The infinite width limit approximation for ResNet yields similar results with an additional residual terms. It is straighforward 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

$$\begin{aligned}\kappa^l_{\alpha,\alpha'}(x,x') &= \kappa^{l-1}_{\alpha,\alpha'}(x,x') + \sigma_b^2 \\ &+ \frac{\sigma_w^2}{2k+1}\sum_{\beta\in ker} F_\phi(\kappa^{l-1}_{\alpha+\beta,\alpha'+\beta}(x,x), \kappa^{l-1}_{\alpha+\beta,\alpha'+\beta}(x,x'), \kappa^{l-1}_{\alpha+\beta,\alpha'+\beta}(x',x')).\end{aligned}$$

### 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 $x, x' \in \mathcal{X}$, $q^1_{\alpha,\alpha'}(x,x')$ is independent of $\alpha, \alpha'$.*

Assumption 1 implies that, there exists some function $e : (x, x') \mapsto e(x, x')$ such that for all $\alpha, \alpha', x, x'$

$$\sum_j \sum_{\beta \in ker_0} x_{j,\alpha+\beta} x'_{j,\alpha'+\beta} = e(x, x')$$

This system has $N^2M^2$ 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* $\sup_{(x,x')\in E\cap\mathcal{X}^2} G(x, x')$.

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 $(y^l_i(x))_{i\in[1:N_l]}$ are iid). Let $q^l(x, x')$, resp. $c^l_1(x, x')$ be the covariance, resp. the correlation between $y^l_1(x)$ and $y^l_1(x')$. For Gradient back-propagation, let $\tilde{q}^l(x, x')$ be the Gradient covariance defined by $\tilde{q}^l(x, x') = \mathbb{E}\left[\frac{\partial \mathcal{L}}{\partial y^l_1}(x)\frac{\partial \mathcal{L}}{\partial y^l_1}(x')\right]$ where $\mathcal{L}$ is some loss function. Similarly, let $\tilde{q}^l(x)$ be the Gradient variance at point $x$. We also define $\dot{q}^l(x, x') = \sigma_w^2\mathbb{E}[\phi'(y^{l-1}_1(x))\phi'(y^{l-1}_1(x'))]$.
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 $(y^l_{i,\alpha}(x))_{i\in[1:N_l]}$ are iid). Let $q^l_{\alpha,\alpha'}(x, x')$ the covariance between $y^l_{1,\alpha}(x)$ and $y^l_{1,\alpha'}(x')$, and $c^l_{\alpha,\alpha'}(x, x')$ the corresponding correlation. We also define the pseudo-covariance $\hat{q}^l_{\alpha,\alpha'}(x, x') = \sigma_b^2 + \sigma_w^2\mathbb{E}[\phi(y^{l-1}_{1,\alpha}(x))\phi(y^{l-1}_{1,\alpha'}(x'))]$ and $\dot{q}^l_{\alpha,\alpha'}(x, x') = \sigma_w^2\mathbb{E}[\phi(y^{l-1}_{1,\alpha}(x))\phi(y^{l-1}_{1,\alpha'}(x'))]$.
The Gradient covariance is defined by $\tilde{q}^l_{\alpha,\alpha'}(x, x') = \mathbb{E}\left[\frac{\partial \mathcal{L}}{\partial y^l_{1,\alpha}}(x)\frac{\partial \mathcal{L}}{\partial y^l_{1,\alpha'}}(x')\right]$.

#### 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

$$q^l(x, x') = \sigma_b^2 + \sigma_w^2\mathbb{E}[\phi(y^{l-1}_i(x))\phi(y^{l-1}_i(x'))] \tag{12}$$

this can be written as

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

with $c^{l-1} := c^{l-1}(x, x')$.
With ReLU, and since ReLU is positively homogeneous (i.e. $\phi(\lambda x) = \lambda\phi(x)$ for $\lambda \geq 0$), we have that

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

where $f$ is the ReLU correlation function given by (Hayou et al., 2019)

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

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

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

Observe that

$$q^l_{\alpha,\alpha'}(x, x') = \frac{1}{2k+1}\sum_{\beta inker}\hat{q}^l_{\alpha+\beta,\alpha'+\beta}(x, x') \tag{13}$$

With ReLU, we have

$$q^l_{\alpha,\alpha'}(x,x') = \sigma_b^2 + \frac{\sigma_w^2}{2k+1}\sum_{\beta\in ker}\sqrt{q^l_{\alpha+\beta}(x)}\sqrt{q^l_{\alpha'+\beta}(x')}f(c^{l-1}_{\alpha+\beta,\alpha'+\beta}(x,x')).$$

**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

$$q^l(x,x') = q^{l-1}(x,x') + \sigma_b^2 + \frac{\sigma_w^2}{2}\sqrt{q^l(x)}\sqrt{q^l(x')}f(c^{l-1}) \tag{14}$$

with CNN layers, we have instead

$$q^l_{\alpha,\alpha'}(x,x') = q^{l-1}_{\alpha,\alpha'}(x,x') + \sigma_b^2 + \frac{\sigma_w^2}{2k+1}\sum_{\beta\in ker}\sqrt{q^l_{\alpha+\beta}(x)}\sqrt{q^l_{\alpha'+\beta}(x')}f(c^{l-1}_{\alpha+\beta,\alpha'+\beta}(x,x')) \tag{15}$$

#### 4.1.2. Gradient Covariance back-propagation

**Gradient back-propagation for FFNN.** The gradient back-propagation is given by

$$\frac{\partial\mathcal{L}}{\partial y^l_i} = \phi'(y^l_i)\sum_{j=1}^{N_{l+1}}\frac{\partial\mathcal{L}}{\partial y^{l+1}_j}W^{l+1}_{ji}.$$

where $\mathcal{L}$ is some loss function. Using the Gradient Independence 2.2, we have as in (Schoenholz et al., 2017)

$$\tilde{q}^l(x) = \tilde{q}^{l+1}(x)\frac{N_{l+1}}{N_l}\chi(q^l(x)).$$

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

$$\frac{\partial\mathcal{L}}{\partial W^l_{i,j,\beta}} = \sum_\alpha\frac{\partial\mathcal{L}}{\partial y^l_{i,\alpha}}\phi(y^{l-1}_{j,\alpha+\beta})$$

Moreover,

$$\frac{\partial\mathcal{L}}{\partial y^l_{i,\alpha}} = \sum_{j=1}^n\sum_{\beta\in ker}\frac{\partial\mathcal{L}}{\partial y^{l+1}_{j,\alpha-\beta}}W^{l+1}_{i,j,\beta}\phi'(y^l_{i,\alpha}).$$

Using the Gradient Independence 2.2, and taking the average over the number of channels we have that

$$\mathbb{E}\left[\frac{\partial\mathcal{L}}{\partial y^l_{i,\alpha}}^2\right] = \frac{\sigma_w^2\mathbb{E}\left[\phi'(\sqrt{q^l_\alpha(x)}Z)^2\right]}{2k+1}\sum_{\beta\in ker}\mathbb{E}\left[\frac{\partial\mathcal{L}}{\partial y^{l+1}_{i,\alpha-\beta}}^2\right].$$

We can get similar recursion to that of the FFNN case by summing over $\alpha$ and using the periodic boundary condition, this yields

$$\sum_\alpha\mathbb{E}\left[\frac{\partial\mathcal{L}}{\partial y^l_{i,\alpha}}^2\right] = \chi(q^l_\alpha(x))\sum_\alpha\mathbb{E}\left[\frac{\partial\mathcal{L}}{\partial y^{l+1}_{i,\alpha}}^2\right].$$

#### 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 $c^l(x,x')$ between $y^l(x)$ and $y^l(x')$ for any two

inputs $x$ and $x'$ is also driven by $(\sigma_b, \sigma_w)$: 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 $\sigma_w^2 \mathbb{E}[\phi'(\sqrt{q(\sigma_b, \sigma_w)}Z)^2] = 1$ where $Z \sim \mathcal{N}(0,1)$. Similarly the Ordered, resp. Chaotic, phase is defined by $\sigma_w^2 \mathbb{E}[\phi'(\sqrt{q(\sigma_b, \sigma_w)}Z)^2] < 1$, resp. $\sigma_w^2 \mathbb{E}[\phi'(\sqrt{q(\sigma_b, \sigma_w)}Z)^2] > 1$. On the Ordered phase, the gradient will vanish as it backpropagates through the network, and the correlation $c^l(x, x')$ converges exponentially to 1. 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 $w_{ij}^l, b_i^l \overset{iid}{\sim} \mathcal{N}(0,1)$, 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 $l \geq 1$, $\sup_{x \in \mathbb{R}^d} |q^l(x,x) - q| \leq e^{-\lambda l}$. (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 $(\sigma_b, \sigma_w) \in$ EOC. Using the same notation as in fact 4, we have that $\sup_{(x,x') \in B_\epsilon} |1 - c^l(x,x')| = \mathcal{O}(l^{-1})$. (Proposition 3 in (Hayou et al., 2019)).*

**Fact 4.** *Let $B_\epsilon = \{(x,x') \in \mathbb{R}^d : c^1(x,x') < 1 - \epsilon\}$. 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 $f(x) = \frac{\sigma_b^2 + \sigma_w^2 \mathbb{E}[\phi(\sqrt{q}Z_1)\phi(\sqrt{q}(xZ_1 + \sqrt{1-x^2}Z_2))]}{q}$ 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*

$$f^{(j)}(x) = \sigma_w^2 q^{j-1} \mathbb{E}[\phi^{(j)}(Z_1)\phi^{(j)}(xZ_1 + \sqrt{1-x^2}Z_2)], \quad \forall x \in [-1,1]$$

*As a result, we have that $f^{(j)}(1) = \sigma_w^2 q^{j-1} \mathbb{E}[\phi^{(j)}(Z_1)^2] > 0$ 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 $(\sigma_b, \sigma_w) \in$ 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 Ord $= \{(\sigma_b, \sigma_w) \in (\mathbb{R}^+)^2 : \sigma_w < \sqrt{2}\}$. Moreover, for any $(\sigma_b, \sigma_w) \in$ 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 $q = \frac{\sigma_b^2}{1 - \sigma_w^2/2}$.*

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 $l \geq 1$, $\sup_{(x,x')\in\mathbb{R}^d} |c^l(x,x') - 1| \leq e^{-\lambda l}$.*

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 $l \geq 1$, $x \in \mathbb{R}^d, q^l(x,x) \gtrsim (\sigma_w^2/2)^l$.*

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 $q^l(x,x) = \frac{\sigma_w^2}{d}||x||^2$ 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*

$$\sup_{(x,x')\in B_\epsilon} |1 - c^l(x,x')| = \mathcal{O}(l^{-2})$$

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

**Fact 14.** *We have that*

$$f(c) = c + s(1-c)^{3/2} + b(1-c)^{5/2} + O((1-c)^{7/2} \tag{16}$$

*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 $l \geq 1$, $\sup_{\alpha,\alpha'} \sup_{x\in\mathbb{R}^d} |q^l_{\alpha,\alpha'}(x,x) - q| \leq e^{-\lambda l}$. (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*

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

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

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

where $\zeta_l = \sup_{t\in(\min_{(x,x')\in B_\epsilon} c^l(x,x'),1)} |f''(t)| \to |f''(1)|$. 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 $\tilde{f}'(c) = \frac{\sigma_w^2}{2} f'(c)$ where $f$ is given in fact 12. $f'(x) = 1 - \frac{\sqrt{2}}{\pi}(1-x)^{1/2} + \mathcal{O}((1-x)^{3/2})$. Therefore, there exists $l_0, \zeta > 0$ such that for $l > l_0$,

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

We conclude using fact 9. □

**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*

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

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

$$\sup_{(x,x')\in B_\epsilon} \left| f'(c^l(x,x')) - 1 + \frac{3}{l} - \frac{9}{2\sqrt{\kappa}}\frac{\log(l)}{l^2} \right| = \mathcal{O}(l^{-2}).$$

*Proof.* Let $(x,x') \in B_\epsilon$ and $s = \frac{2\sqrt{2}}{3\pi}$. From the preliminary results, we have that $\lim_{l\to\infty} \sup_{x,x'\in\mathbb{R}^d} 1 - c^l(x,x') = 0$ (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 $\gamma_{l+1}^{-1/2} = \gamma_l^{-1/2} + \frac{s}{2} + \frac{3s^2}{8}\gamma_l^{1/2} + O(\gamma_l^{3/2})$, using the same argument multiple times and inverting the formula yields

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

Note that, by Appendix Lemma 5 (section 5), the $\mathcal{O}$ bound can be chosen in a way that it does not depend on $(x, x')$, it depends only on $\epsilon$; this concludes the proof for the first part of the result.
Using fact 12, we have that

$$\begin{aligned} f'(x) &= \frac{1}{\pi}\arcsin(x) + \frac{1}{2} \\ &= 1 - \frac{\sqrt{2}}{\pi}(1-x)^{1/2} + O((1-x)^{3/2}). \end{aligned}$$

Thus, it follows that

$$f'(c^l(x,x')) = 1 - \frac{3}{l} + \frac{9\sqrt{2}}{4}\frac{\log(l)}{l^2} + \mathcal{O}(l^{-2}).$$

uniformly over the set $B_\epsilon$, which concludes the proof.

□

We prove a similar result for an FFNN with Tanh activation.

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

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

*where $\kappa = \frac{2}{f''(1)} > 0$ and $\zeta = \frac{f^3(1)}{6} > 0$. Moreover, we have that*

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

*Proof.* Let $(x, x') \in B_\epsilon$ and $\lambda_l := 1 - c^l(x, x')$. 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

$$\begin{aligned} \lambda_{l+1}^{-1} &= \lambda_l^{-1}(1 - \alpha\lambda_l + \zeta\lambda^2 + O(\lambda_l^3))^{-1} = \lambda_l^{-1}(1 + \alpha\lambda_l + (\alpha^2 - \zeta)\lambda_l^2 + O(\lambda_l^3)) \\ &= \lambda_l^{-1} + \alpha + (\alpha^2 - \zeta)\lambda_l + O(\lambda_l^2). \end{aligned}$$

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 $\mathcal{O}$ 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

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

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, $\hat{q}^l_{\alpha,\alpha'}(x,x') = \sigma_b^2 + \sigma_w^2 \mathbb{E}[\phi(y^{l-1}_{1,\alpha}(x))\phi(y^{l-1}_{1,\alpha'}(x'))]$ and $q^l_{\alpha,\alpha'}(x,x') = \mathbb{E}[y^l_{i,\alpha}(x) y^l_{i,\alpha'}(x')]$.

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

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

Let $c^l_{\alpha,\alpha'}(x,x')$ be the corresponding correlation. Since $q^l_{\alpha,\alpha}(x,x)$ 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

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

where $f(c) = \frac{\sigma_w^2 \mathbb{E}[\phi(\sqrt{q}Z_1)\phi(\sqrt{q}(cZ_1 + \sqrt{1-c^2}Z_2))] + \sigma_b^2}{q}$ and $Z_1, Z_2$ are independent standard normal variables. The dynamics of $c^l_{\alpha,\alpha'}$ 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 $c^l_{\alpha,\alpha'}(x,x)$ (same input $x$), however, they do not study $c^l_{\alpha,\alpha'}(x,x')$ 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 $B_\epsilon = \{(x,x') \in \mathbb{R}^d : \sup_{\alpha,\alpha'} c^1_{\alpha,\alpha'}(x,x') < 1 - \epsilon\}$. The following statements hold*

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

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

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

$$\sup_{(x,x') \in B_\epsilon} \sup_{\alpha,\alpha'} |c^l_{\alpha,\alpha'}(x,x') - c| = \mathcal{O}(e^{-\beta l})$$

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

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

*where $\kappa = \frac{2}{f''(1)} > 0$, $\zeta = \frac{f^3(1)}{6} > 0$, 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

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

Writing this in matrix form yields

$$C_l = \frac{1}{2k+1} U f(C_{l-1})$$

where $C_l = ((c^l_{\alpha,\alpha+\beta}(x,x'))_{\alpha\in[0:N-1]})_{\beta\in[0:N-1]}$ is a vector in $\mathbb{R}^{N^2}$, $U$ is a convolution matrix and $f$ is applied element-wise. As an example, for $k=1$, $U$ is given by

$$U = \begin{bmatrix} 1 & 1 & 0 & ... & 0 & 1 \\ 1 & 1 & 1 & 0 & \ddots & 0 \\ 0 & 1 & 1 & 1 & \ddots & 0 \\ 0 & 0 & 1 & 1 & \ddots & 0 \\ & \ddots & \ddots & \ddots & \ddots & \\ 1 & 0 & \dots & 0 & 1 & 1 \end{bmatrix}$$

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

$$(1+2k)^{-l}U^l = e_1 e_1^T + O(e^{-\beta l})$$

where $\beta = \log(\frac{\lambda_1}{\lambda_2})$.
This provides another justification to Assumption 1; as $l$ grows, and assuming that $C_l \to e_1$ (which we show in the remainder of this proof), $C_l$ exhibits a self-averaging property since $C_l \approx \frac{1}{2k+1}UC_{l-1}$. 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 $(\sigma_b, \sigma_w)$ be in the Ordered phase, $(x,x') \in \mathbb{R}^d$ and $c^l_m = \min_{\alpha,\alpha'} c^l_{\alpha,\alpha'}(x,x')$. Using the fact that $f$ is non-decreasing, we have that $c^l_{\alpha,\alpha'}(x,x') \geq \frac{1}{2k+1}\sum_{\beta\in ker} c^{l-1}_{\alpha+\beta,\alpha'+\beta}(x,x')) \geq c^{l-1}_m$. Taking the minimum again over $\alpha,\alpha'$, we have $c^l_m \geq c^{l-1}_m$, therefore $c^l_m$ is non-decreasing and converges to the unique fixed point of $f$ which is $c=1$. This proves that $\sup_{\alpha,\alpha'} |c^l_{\alpha,\alpha'}(x,x') - 1| \to 0$. Moreover, the convergence rate is exponential using the fact that (fact 16) $0 < f'(1) < 1$. To see this, observe that

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

   Knowing that $\sup_{\zeta\in[c^{l-1}_m,1]} f'(\zeta) \to f'(1) < 1$, 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 $B_\epsilon$ to avoid points where $c^1(x,x') = 1$. 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 $(\sigma_b, \sigma_w) \in$ EOC. Using the same argument of monotony as in the previous cases and that $f$ has 1 as unique fixed point, we have that $\lim_{l\to\infty} \sup_{x,x'} \sup_{\alpha,\alpha'} |1 - c^l_{\alpha,\alpha'}(x,x')| = 0$.
   From fact 7, the Taylor expansion of $f$ near 1 is given by

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

   where $\alpha = \frac{f''(1)}{2}$ and $\zeta = \frac{f^{(3)}(1)}{6}$. Using fact 6, we know that $f^{(k)}(1) = \sigma_w^2 q^{k-1}\mathbb{E}[\phi^{(k)}(\sqrt{q}Z)^2]$. 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*
$$\sup_{(x,x')\in\mathbb{R}^d} \sup_{\alpha,\alpha'} |c^l_{\alpha,\alpha'}(x,x') - 1| = \mathcal{O}(e^{-\beta l})$$
2. *If $(\sigma_b, \sigma_w)$ are on the Chaotic phase, then there exists $\beta > 0$ and $c \in (0,1)$ such that*
$$\sup_{(x,x')\in B_\epsilon} \sup_{\alpha,\alpha'} |c^l_{\alpha,\alpha'}(x,x') - c| = \mathcal{O}(e^{-\beta l})$$
3. *Under Assumption 1, if $(\sigma_b, \sigma_w) \in EOC$, then*
$$\sup_{(x,x')\in B_\epsilon} \sup_{\alpha,\alpha'} \left| c^l(x,x') - 1 + \frac{\kappa}{l^2} - 3\sqrt{\kappa}\frac{\log(l)}{l^3} \right| = \mathcal{O}(l^{-3})$$
*where $\kappa = \frac{9\pi^2}{2}$.*

*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. □

## 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 $B_\epsilon$. 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 $(x,x') \in B_\epsilon$. 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 $\zeta_0 \in A$,*
$$\zeta_l = \alpha_l + \mathcal{O}_{\zeta_0}(\beta_l)$$
*where $\mathcal{O}_{\zeta_0}$ means that the $\mathcal{O}$ bound depends on $\zeta_0$. Then, we have that*
$$\sup_{\zeta_0\in A} |\zeta_l - \alpha_l| = \mathcal{O}(\beta_l)$$
*i.e. we can choose the bound $\mathcal{O}$ 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. □

Note that Appendix Lemma 5 can be easily extended to Taylor expansions with '$o$' instead of '$\mathcal{O}$'. 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 $x, x' \in \mathbb{R}^d$, $i, i' \leq n_L$, $K^L_{ii'}(x, x') = \delta_{ii'} K^L(x, x')$, where $K^L(x, x')$ is given by the recursive formula*

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

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

*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 $i \in [1 : n_l]$

$$\partial_{\theta_{1:l}} y_i^{l+1}(x) = \frac{\sigma_w}{\sqrt{n_l}} \sum_{j=1}^{n_l} w_{ij}^{l+1} \phi'(y_j^l(x)) \partial_{\theta_{1:l}} y_j^l(x).$$

Therefore,

$$(\partial_{\theta_{1:l}} y_i^{l+1}(x))(\partial_{\theta_{1:l}} y_i^{l+1}(x'))^t = \frac{\sigma_w^2}{n_l} \sum_{j,j'}^{n_l} w_{ij}^{l+1} w_{ij'}^{l+1} \phi'(y_j^l(x))\phi'(y_{j'}^l(x')) \partial_{\theta_{1:l}} y_j^l(x) (\partial_{\theta_{1:l}} y_{j'}^l(x'))^t$$

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 $x, x'$

$$\partial_{\theta_{1:l}} y_j^l(x) (\partial_{\theta_{1:l}} y_{j'}^l(x'))^t \to K^l(x, x') \mathbf{1}_{j=j'}$$

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

$$\frac{\sigma_w^2}{n_l} \sum_{j,j'}^{n_l} w_{ij}^{l+1} w_{ij'}^{l+1} \phi'(y_j^l(x))\phi'(y_{j'}^l(x')) \partial_{\theta_{1:l}} y_j^l(x) (\partial_{\theta_{1:l}} y_{j'}^l(x'))^t \to \quad \frac{\sigma_w^2}{n_l} \sum_{j}^{n_l} (w_{ij}^{l+1})^2 \phi'(y_j^l(x))\phi'(y_j^l(x')) K^l(x, x')$$

and letting $n_l$ go to infinity, the law of large numbers, implies that

$$\frac{\sigma_w^2}{n_l} \sum_{j}^{n_l} (w_{ij}^{l+1})^2 \phi'(y_j^l(x))\phi'(y_j^l(x')) K^l(x, x') \to \dot{q}^{l+1}(x, x') K^l(x, x').$$

Moreover, we have that

$$\begin{aligned}(\partial_{w^{l+1}} y_i^{l+1}(x))(\partial_{w^{l+1}} y_i^{l+1}(x'))^t + (\partial_{b^{l+1}} y_i^{l+1}(x))(\partial_{b^{l+1}} y_i^{l+1}(x'))^t &= \frac{\sigma_w^2}{n_l} \sum_j \phi(y_j^l(x))\phi(y_j^l(x')) + \sigma_b^2 \\ &\underset{n_l \to \infty}{\to} \sigma_w^2 \mathbb{E}[\phi(y_i^l(x))\phi(y_i^l(x'))] + \sigma_b^2 = q^{l+1}(x, x').\end{aligned}$$

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 $\alpha, \alpha' \in [0 : M-1]$*

$$K^1_{(i,\alpha),(i',\alpha')}(x,x') = \delta_{ii'}\Big(\frac{\sigma_w^2}{n_0(2k+1)}[x,x']_{\alpha,\alpha'} + \sigma_b^2\Big)$$

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

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

*where $\Psi^{l-1}_{\alpha,\alpha'} = \dot{q}^l_{\alpha,\alpha'}K^{l-1}_{\alpha,\alpha'} + \hat{q}^l_{\alpha,\alpha'}$, and $\hat{q}^l_{\alpha,\alpha}, \dot{q}^l_{\alpha,\alpha'}$ are defined in Lemma 1, with $y^{l-1}_{1,\alpha}(x), y^{l-1}_{1,\alpha'}(x')$ in place of $y^{l-1}_1(x), y^{l-1}_1(x')$.*

*Proof.* Let $x, x'$ be two inputs. We have that

$$y^1_{i,\alpha}(x) = \frac{\sigma_w}{\sqrt{v_1}}\sum_{j=1}^{n_0}\sum_{\beta \in ker_1} w^1_{i,j,\beta}x_{j,\alpha+\beta} + \sigma_b b^1_i$$
$$y^l_{i,\alpha}(x) = \frac{\sigma_w}{\sqrt{v_l}}\sum_{j=1}^{n_{l-1}}\sum_{\beta \in ker_l} w^l_{i,j,\beta}\phi(y^{l-1}_{j,\alpha+\beta}(x)) + \sigma_b b^l_i$$

therefore

$$\begin{aligned} K^1_{(i,\alpha),(i',\alpha')}(x,x') &= \sum_r \left(\sum_j\sum_\beta \frac{\partial y^1_{i,\alpha}(x)}{\partial w^1_{r,j,\beta}}\frac{\partial y^1_{i',\alpha'}(x)}{\partial w^1_{r,j,\beta}}\right) + \frac{\partial y^1_{i,\alpha}(x)}{\partial b^1_r}\frac{\partial y^1_{i',\alpha'}(x)}{\partial b^1_r} \\ &= \delta_{ii'}\left(\frac{\sigma_w^2}{n_0(2k+1)}\sum_j\sum_\beta x_{j,\alpha+\beta}x_{j,\alpha'+\beta} + \sigma_b^2\right)\end{aligned}$$

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 $\partial_{\theta_{1:l-1}}y^l_{i,\alpha}(x) = \frac{\partial y^l_{i,\alpha}(x)}{\partial \theta_{1:l-1}}$. We have that

$$\partial_{\theta_{1:l-1}}y^l_{i,\alpha}(x) = \frac{\sigma_w}{\sqrt{n_{l-1}(2k+1)}}\sum_j\sum_\beta w^l_{i,j,\beta}\phi'(y^{l-1}_{j,\alpha+\beta})\partial_{\theta_{1:l-1}}y^{l-1}_{i,\alpha+\beta}(x)$$

this yields

$$\begin{aligned}&\partial_{\theta_{1:l-1}}y^l_{i,\alpha}(x)\partial_{\theta_{1:l-1}}y^l_{i',\alpha'}(x)^T = \\ &\frac{\sigma_w^2}{n_{l-1}(2k+1)}\sum_{j,j'}\sum_{\beta,\beta'} w^l_{i,j,\beta}w^l_{i',j',\beta'}\phi'(y^{l-1}_{j,\alpha+\beta})\phi'(y^{l-1}_{j',\alpha'+\beta})\partial_{\theta_{1:l-1}}y^{l-1}_{j,\alpha+\beta}(x)\partial_{\theta_{1:l-1}}y^{l-1}_{j',\alpha'+\beta}(x)^T\end{aligned}$$

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

$$\begin{aligned}&\partial_{\theta_{1:l-1}}y^l_{i,\alpha}(x)\partial_{\theta_{1:l-1}}y^l_{i',\alpha'}(x)^T \to \\ &\frac{\sigma_w^2}{n_{l-1}(2k+1)}\sum_j\sum_{\beta,\beta'} w^l_{i,j,\beta}w^l_{i',j,\beta'}\phi'(y^{l-1}_{j,\alpha+\beta})\phi'(y^{l-1}_{j,\alpha'+\beta})K^{l-1}_{(j,\alpha+\beta),(j,\alpha'+\beta)}(x,x')\end{aligned}$$

note 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

$$\partial_{\theta_{1:l-1}} y^l_{i,\alpha}(x) \partial_{\theta_{1:l-1}} y^l_{i',\alpha'}(x)^T \to$$
$$\delta_{ii'}\Big(\frac{1}{(2k+1)} \sum_{\beta,\beta'} \dot{q}^l_{\alpha+\beta,\alpha'+\beta} K^{l-1}_{(1,\alpha+\beta),(1,\alpha'+\beta)}(x,x')\Big)$$

We conclude using the fact that

$$\partial_{\theta_l} y^l_{i,\alpha}(x) \partial_{\theta_l} y^l_{i',\alpha'}(x)^T \to \delta_{ii'}\Big(\frac{\sigma_w^2}{2k+1} \sum_{\beta} \mathbb{E}[\phi(y^{l-1}_{\alpha+\beta}(x))\phi(y^{l-1}_{\alpha'+\beta}(x'))] + \sigma_b^2\Big)$$

□

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 $\forall l \geq 0, a_{l+1} \leq \alpha a_l + k e^{-\beta l}$, 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

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

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

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

where $q^1(x,x') = \sigma_b^2 + \frac{\sigma_w^2}{d} x^T x'$ and $q^l(x,x') = \sigma_b^2 + \sigma_w^2 \mathbb{E}_{f\sim\mathcal{N}(0,q^{l-1})}[\phi(f(x))\phi(f(x'))]$ and $\dot{q}^l(x,x') = \sigma_w^2 \mathbb{E}_{f\sim\mathcal{N}(0,q^{l-1})}[\phi'(f(x))\phi'(f(x'))]$. 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

$$\sup_{(x,x')\in B_\epsilon} |\dot{q}^l(x,x') - \alpha| \leq e^{-\eta l}.$$

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 $r_l = \sup_{(x,x')\in B_\epsilon} |K^l(x,x') - \frac{k}{1-\alpha}|$, 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 $(a_0, b_0, \lambda_0) \in A \times B \times \Lambda$*

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

*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 '$\mathcal{O}$' and '$o$' depend only on $A, B, \Lambda \subset \mathbb{R}$. Then, we have*

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

*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 $l \geq 2$

$$\begin{aligned} r_l &= r_{l-1}(1 - \frac{1}{l})(1 - \frac{\alpha}{l} + \mathcal{O}(l^{-1-\beta})) + \frac{q}{l} + o(l^{-2}) \\ &= r_{l-1}(1 - \frac{1+\alpha}{l}) + \frac{q}{l} + \mathcal{O}(l^{-1-\beta}). \end{aligned}$$

where $\mathcal{O}$ bound depends only on $A, B, \Lambda$. Letting $x_l = r_l - \frac{q}{1+\alpha}$, 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

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

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

$$\prod_{k=l_0}^{l} (1 - \frac{1+\alpha}{k}) = \Theta(l^{-1-\alpha}).$$

where the bound $\Theta$ does not depend on $l_0$. For the second part, observe that

$$\prod_{j=k+1}^{l}(1-\frac{1+\alpha}{j}) = \frac{(l-\alpha-1)!}{l!}\frac{k!}{(k-\alpha-1)!}$$

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

$$\begin{aligned}\sum_{k=l_0}^{l}\frac{k!}{(k-\alpha-1)!}\frac{1}{k^2} &\sim \sum_{k=1}^{l} k^{\alpha-\beta} \\ &\sim \int_1^l t^{\alpha-\beta}dt \\ &\sim \frac{1}{\alpha-\beta+1}l^{\alpha-\beta+1}.\end{aligned}$$

Therefore, it follows that

$$\begin{aligned}\sum_{k=l_0}^{l}\prod_{j=k+1}^{l}(1-\frac{1+\alpha}{j})\frac{1}{k^{1+\beta}} &= \frac{(l-\alpha-1)!}{l!}\sum_{k=l_0}^{l}\frac{k!}{(k-\alpha-1)!}\frac{1}{k^{1+\beta}} \\ &\sim \frac{1}{\alpha}l^{-\beta}.\end{aligned}$$

This proves that

$$x_l \leq \frac{M}{\alpha}l^{-\beta} + o(l^{-\beta}).$$

where the '$o$' bound depends only on $A, B, \Lambda$. Using the same approach for the left-hand inequality, we prove that

$$x_l \geq -\frac{M}{\alpha}l^{-\beta} + o(l^{-\beta}).$$

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 $(a_0, b_0, \lambda_0) \in A \times B \times \Lambda$*

$$\begin{aligned}a_l &= a_{l-1}\lambda_l + b_l, \quad b_l = q(b_0) + \mathcal{O}(l^{-1}), \\ \lambda_l &= 1 - \frac{\alpha}{l} + \kappa\frac{\log(l)}{l^2} + \mathcal{O}(l^{-2}),\end{aligned}$$

*where $\alpha \in \mathbb{N}^*, \kappa \neq 0$ both do not depend on $a_0, b_0, \Lambda_0$, $q(b_o) \in \mathbb{R}^+$ 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

$$\begin{aligned}r_l &= r_{l-1}(1-\frac{1}{l})(1-\frac{\alpha}{l}+\kappa\frac{\log(l)}{l^2}+O(l^{-1-\beta})) + \frac{q}{l} + \mathcal{O}(l^{-2}) \\ &= r_{l-1}(1-\frac{1+\alpha}{l}) + r_{l-1}\kappa\frac{\log(l)}{l^2} + \frac{q}{l} + \mathcal{O}(l^{-2}).\end{aligned}$$

Let $x_l = r_l - \frac{q}{1+\alpha}$. It is clear that $\lambda_l = 1 - \alpha/l + \mathcal{O}(l^{-3/2})$. 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

$$x_l \leq x_{l_0} \prod_{k=l_0}^{l} (1 - \frac{1+\alpha}{k}) + \sum_{k=l_0}^{l} \prod_{j=k+1}^{l} (1 - \frac{1+\alpha}{j}) \frac{\kappa_2 \log(k) + M}{k^2}$$

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

$$\prod_{j=k+1}^{l} (1 - \frac{1+\alpha}{j}) = \frac{(l-\alpha-1)!}{l!} \frac{k!}{(k-\alpha-1)!}$$

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

$$\begin{aligned} \sum_{k=l_0}^{l} \frac{k!}{(k-\alpha-1)!} \frac{\kappa_1 \log(k) - M}{k^2} &\sim \sum_{k=1}^{l} \log(k) k^{\alpha-1} \\ &\sim \int_1^l \log(t) t^{\alpha-1} dt \\ &\sim C_1 l^\alpha \log(l), \end{aligned}$$

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 $\frac{(l-\alpha-1)!}{l!} \sim l^{-1-\alpha}$ 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.

□

**Theorem 1** (Neural Tangent Kernel on the Edge of Chaos). *Let $\phi$ be ReLU or Tanh, $(\sigma_b, \sigma_w) \in EOC$ and $AK^L = K^L/L$. We have that*

$$\sup_{x\in E} |AK^L(x,x) - AK^\infty(x,x)| = \mathcal{O}(L^{-1})$$

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

$$\sup_{(x,x')\in B_\epsilon} \left|AK^L(x,x') - AK^\infty(x,x')\right| = \Theta(\log(L)L^{-1}).$$

*where*
- *if $\phi$ is ReLU-like, then $AK^\infty(x,x') = \frac{\sigma_w^2 \|x\| \, \|x'\|}{d}(1-(1-\lambda)\mathbb{1}_{x\neq x'})$.*
- *if $\phi$ is Tanh, then $AK^\infty(x,x') = q(1-(1-\lambda)\mathbb{1}_{x\neq x'})$ 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 $\epsilon \in (0,1)$, $(\sigma_b,\sigma_w) \in$ EOC, $x, x' \in \mathbb{R}^d$ and recall $c^l(x,x') = \frac{q^l(x,x')}{\sqrt{q^l(x,x)q^l(x',x')}}$. 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 $\dot{q}^l(x,x) = f'(c^{l-1}(x,x'))$. We first prove the result for ReLU, then we extend it to Tanh.

- $\phi =$*ReLU*: From fact 11, we know that, on the EOC for ReLU, the variance $q^l(x,x)$ is constant wrt $l$ and given by $q^l(x,x) = q^1(x,x) = \frac{\sigma_w^2}{d}||x||^2$, and from fact 16 that $\dot{q}^l(x,x) = 1$. Therefore

$$K^l(x,x) = K^{l-1}(x,x) + \frac{\sigma_w^2}{d}||x||^2 = l\frac{\sigma_w^2}{d}||x||^2 = lAK^\infty(x,x)$$

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

$$\sup_{(x,x')\in B_\epsilon} \left|\frac{K^{l}(x,x')}{l} - \frac{1}{4}\frac{\sigma_w^2}{d}\|x\|\|x'\|\right| = \Theta(\log(l)l^{-1})$$

- $\phi = Tanh$: 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

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

where $\kappa = \frac{2}{f''(1)} > 0$ and $\zeta = \frac{f^3(1)}{6} > 0$. Moreover, we have that

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

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 $K^l_{\alpha,\alpha'}$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 layers in the infinite width limit)**.** *Let $x, x'$ be two inputs and $K^{res,1}$ 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 $x, x' \in \mathbb{R}^d$*

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

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

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

$$K^l_{res}(x,x') = K^{l-1}_{res}(x,x')(\dot{q}^l(x,x') + 1) + \hat{q}^l(x,x').$$

*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

$$K^1_{res}(x,x') = \sum_j \frac{\partial y^1_1(x)}{\partial w^1_{1j}} \frac{\partial y^1_1(x)}{\partial w^1_{1j}} + \frac{\partial y^1_1(x)}{\partial b^1_1} \frac{\partial y^1_1(x)}{\partial b^1_1} = \frac{\sigma_w^2}{d} x \cdot x' + \sigma_b^2.$$

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

$$\partial_{\theta_{1:l}} y^{l+1}_i(x) = \partial_{\theta_{1:l}} y^l_i(x) + \frac{\sigma_w}{\sqrt{n_l}} \sum_{j=1}^{n_l} w^{l+1}_{ij} \phi'(y^l_j(x)) \partial_{\theta_{1:l}} y^l_j(x).$$

  Therefore, we obtain

$$\begin{aligned}(\partial_{\theta_{1:l}} y^{l+1}_i(x))(\partial_{\theta_{1:l}} y^{l+1}_i(x'))^t &= (\partial_{\theta_{1:l}} y^l_i(x))(\partial_{\theta_{1:l}} y^l_i(x'))^t \\ &+ \frac{\sigma_w^2}{n_l} \sum_{j,j'}^{n_l} w^{l+1}_{ij} w^{l+1}_{ij'} \phi'(y^l_j(x)) \phi'(y^l_{j'}(x')) \partial_{\theta_{1:l}} y^l_j(x) (\partial_{\theta_{1:l}} y^l_{j'}(x'))^t + I\end{aligned}$$

  where

$$I = \frac{\sigma_w}{\sqrt{n_l}} \sum_{j=1}^{n_l} w^{l+1}_{ij} (\phi'(y^l_j(x)) \partial_{\theta_{1:l}} y^l_i(x) (\partial_{\theta_{1:l}} y^l_j(x'))^t + \phi'(y^l_j(x')) \partial_{\theta_{1:l}} y^l_j(x) (\partial_{\theta_{1:l}} y^l_i(x'))^t).$$

Using the induction hypothesis, as $n_0, n_1, ..., n_{l-1} \to \infty$, we have that

$$(\partial_{\theta_{1:l}} y_i^{l+1}(x))(\partial_{\theta_{1:l}} y_i^{l+1}(x'))^t + \frac{\sigma_w^2}{n_l}\sum_{j,j'}^{n_l} w_{ij}^{l+1} w_{ij'}^{l+1} \phi'(y_j^l(x))\phi'(y_{j'}^l(x'))\partial_{\theta_{1:l}} y_j^l(x)(\partial_{\theta_{1:l}} y_{j'}^l(x'))^t + I$$

$$\to K_{res}^l(x,x') + \frac{\sigma_w^2}{n_l}\sum_{j}^{n_l}(w_{ij}^{l+1})^2 \phi'(y_j^l(x))\phi'(y_j^l(x'))K_{res}^l(x,x') + I',$$

where $I' = \frac{\sigma_w^2}{n_l} w_{ii}^{l+1}(\phi'(y_i^l(x)) + \phi'(y_i^l(x')))K_{res}^l(x,x')$.

As $n_l \to \infty$, we have that $I' \to 0$. Using the law of large numbers, as $n_l \to \infty$

$$\frac{\sigma_w^2}{n_l}\sum_{j}^{n_l}(w_{ij}^{l+1})^2 \phi'(y_j^l(x))\phi'(y_j^l(x'))K_{res}^l(x,x') \to \dot{q}^{l+1}(x,x')K_{res}^l(x,x').$$

Moreover, we have that

$$(\partial_{w^{l+1}} y_i^{l+1}(x))(\partial_{w^{l+1}} y_i^{l+1}(x'))^t + (\partial_{b^{l+1}} y_i^{l+1}(x))(\partial_{b^{l+1}} y_i^{l+1}(x'))^t = \frac{\sigma_w^2}{n_l}\sum_j \phi(y_j^l(x))\phi(y_j^l(x')) + \sigma_b^2$$

$$\underset{n_l\to\infty}{\to} \sigma_w^2 \mathbb{E}[\phi(y_i^l(x))\phi(y_i^l(x'))] + \sigma_b^2 = q^{l+1}(x,x').$$

□

Now we proof the recursive formula for ResNets with Convolutional layers.

**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 $x, x' \in \mathbb{R}^d$*

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

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

$$K^{res,l}_{\alpha,\alpha'} = K^{res,l-1}_{\alpha,\alpha'} + \frac{1}{2k+1}\sum_{\beta}\Psi^{l-1}_{\alpha+\beta,\alpha'+\beta}.$$

*where $\Psi^l_{\alpha,\alpha'} = \dot{q}^l_{\alpha,\alpha'}K^{res,l}_{\alpha,\alpha'} + \hat{q}^l_{\alpha,\alpha'}$.*

*Proof.* Let $x, x'$ be two inputs. We have that

$$K^1_{(i,\alpha),(i',\alpha')}(x,x') = \sum_j(\sum_\beta \frac{\partial y^1_{i,\alpha}(x)}{\partial w^1_{i,j,\beta}}\frac{\partial y^1_{i',\alpha'}(x)}{\partial w^1_{i',j,\beta}} + \frac{\partial y^1_{i,\alpha}(x)}{\partial b^1_j}\frac{\partial y^1_{i',\alpha'}(x)}{\partial b^1_j})$$

$$= \delta_{ii'}\big(\frac{\sigma_w^2}{n_0(2k+1)}\sum_j\sum_\beta x_{j,\alpha+\beta}x_{j,\alpha'+\beta} + \sigma_b^2\big).$$

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 $\partial_{\theta_{1:l-1}} y^l_{i,\alpha}(x) = \frac{\partial y^l_{i,\alpha}(x)}{\partial \theta_{1:l-1}}$. We have that

$$\partial_{\theta_{1:l-1}} y^l_{i,\alpha}(x) = \partial_{\theta_{1:l-1}} y^{l-1}_{i,\alpha}(x) + \frac{\sigma_w}{\sqrt{n_{l-1}(2k+1)}}\sum_j\sum_\beta w^l_{i,j,\beta}\phi'(y^{l-1}_{j,\alpha+\beta})\partial_{\theta_{1:l-1}} y^{l-1}_{i,\alpha+\beta}(x)$$

this yields

$$\partial_{\theta_{1:l-1}} y^l_{i,\alpha}(x)\partial_{\theta_{1:l-1}} y^l_{i',\alpha'}(x)^T = \partial_{\theta_{1:l-1}} y^{l-1}_{i,\alpha}(x)\partial_{\theta_{1:l-1}} y^{l-1}_{i',\alpha'}(x)^T + \frac{\sigma_w^2}{n_{l-1}(2k+1)} \sum_{j,j'}\sum_{\beta,\beta'} w^l_{i,j,\beta} w^l_{i',j',\beta'} \phi'(y^{l-1}_{j,\alpha+\beta})\phi'(y^{l-1}_{j',\alpha'+\beta})\partial_{\theta_{1:l-1}} y^{l-1}_{j,\alpha+\beta}(x)\partial_{\theta_{1:l-1}} y^{l-1}_{j',\alpha'+\beta}(x)^T + I,$$

where

$$I = \frac{\sigma_w}{\sqrt{n_{l-1}(2k+1)}} \sum_{j,\beta} w^l_{i,j,\beta}\phi'(y^{l-1}_{j,\alpha+\beta})(\partial_{\theta_{1:l-1}} y^{l-1}_{i,\alpha}(x)\partial_{\theta_{1:l-1}} y^{l-1}_{i,\alpha+\beta}(x)^T + \partial_{\theta_{1:l-1}} y^{l-1}_{i,\alpha+\beta}(x)\partial_{\theta_{1:l-1}} y^{l-1}_{i,\alpha}(x)^T).$$

As $n_1, n_2, ..., n_{l-2} \to \infty$ and using the induction hypothesis, we have

$$\partial_{\theta_{1:l-1}} y^l_{i,\alpha}(x)\partial_{\theta_{1:l-1}} y^l_{i',\alpha'}(x)^T \to \delta_{ii'} K^{l-1}_{\alpha,\alpha'}(x,x') + \frac{\sigma_w^2}{n_{l-1}(2k+1)} \sum_{j}\sum_{\beta,\beta'} w^l_{i,j,\beta} w^l_{i',j,\beta'} \phi'(y^{l-1}_{j,\alpha+\beta})\phi'(y^{l-1}_{j,\alpha'+\beta}) K^{l-1}_{(j,\alpha+\beta),(j,\alpha'+\beta)}(x,x').$$

Note 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

$$\partial_{\theta_{1:l-1}} y^l_{i,\alpha}(x)\partial_{\theta_{1:l-1}} y^l_{i',\alpha'}(x)^T \to \delta_{ii'} K^{l-1}_{\alpha,\alpha'}(x,x') + \delta_{ii'}\Big(\frac{1}{(2k+1)} \sum_{\beta,\beta'} f'(c^{l-1}_{\alpha+\beta,\alpha'+\beta}(x,x')) K^{l-1}_{(1,\alpha+\beta),(1,\alpha'+\beta)}(x,x')\Big),$$

where $f'(c^{l-1}_{\alpha+\beta,\alpha'+\beta}(x,x')) = \sigma_w^2 \mathbb{E}[\phi'(y^{l-1}_{j,\alpha+\beta})\phi'(y^{l-1}_{j,\alpha'+\beta})]$.
We conclude using the fact that

$$\partial_{\theta_l} y^l_{i,\alpha}(x)\partial_{\theta_l} y^l_{i',\alpha'}(x)^T \to \delta_{ii'}\Big(\frac{\sigma_w^2}{2k+1} \sum_{\beta} \mathbb{E}[\phi(y^{l-1}_{\alpha+\beta}(x))\phi(y^{l-1}_{\alpha'+\beta}(x'))] + \sigma_b^2\Big).$$

□

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*

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

*where $\kappa = \frac{9\pi^2}{2}(1+\frac{2}{\sigma_w^2})^2$. Moreover, we have that*

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

*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* $\sup_{(x,x')\in B_\epsilon}\sup_{\alpha,\alpha'}$.

*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

$$q^l(x,x) = q^{l-1}(x,x) + \sigma_w^2/2 q^{l-1}(x,x) = (1+\sigma_w^2/2)^{l-1} q^1(x,x)$$

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 $\bar{K}_{res}^{\infty}$.

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

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

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

$$\sup_{x\in E}|\bar{K}_{res}^L(x,x) - \bar{K}_{res}^{\infty}(x,x)| = \Theta(L^{-1})$$

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

$$\sup_{x,x'\in B_\epsilon}\left|\bar{K}_{res}^L(x,x') - \bar{K}_{res}^{\infty}(x,x')\right| = \Theta(L^{-1}\log(L)),$$

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

**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 $K_{res}^L(x,x)$ then $K_{res}^L(x,x')$.

*Proof.* • Diagonal terms: using properties on the correlation function $f$ (fact 12), we have that $\dot{q}^l(x,x) = \frac{\sigma_w^2}{2}f(1) = \frac{\sigma_w^2}{2}$. 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

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

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

$$K_{res}^l(x,x') = K_{res}^{l-1}(x,x')(\dot{q}^l(x,x')+1) + q^l(x,x')$$

Hence, for the diagonal terms we obtain

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

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

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

Therefore, $\bar{K}^l_{res}(x,x) = \frac{\hat{K}^1_{res}(x,x)}{l} + (1-1/l)\frac{\sigma_w^2}{d}\|x\|^2$, the conclusion is straightforward since $E$ is compact and ${K'_{res}}^1(x,x)$ 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

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

where $\kappa = \frac{9\pi^2}{2}(1+\frac{2}{\sigma_w^2})^2$. Moreover, we have that

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

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

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

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 $a_l = \frac{K^{l+1}_{res}(x,x')}{(1+\alpha)^l}$. Using the recursive formula of the NTK, we obtain

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

where $\lambda_l = 1 - 3l^{-1} + \zeta\frac{\log(l)}{l^2} + \mathcal{O}(l^{-3})$, $b_l = \frac{\sigma_w^2}{d}\sqrt{\|x\|\|x'\|}f(c^l(x,x')) = q(x,x') + \mathcal{O}(l^{-2})$ with $q(x,x') = \frac{\sigma_w^2}{d}\sqrt{\|x\|\|x'\|}$ 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

$$\sup_{x,x'\in B_\epsilon}\left|K^{L+1}_{res}(x,x')/L(1+\alpha)^L - \bar{K}^\infty_{res}(x,x')\right| = \Theta(L^{-1}\log(L)),$$

which can also be written as

$$\sup_{x,x'\in B_\epsilon}\left|K^{L}_{res}(x,x')/(L-1)(1+\alpha)^{L-1} - \bar{K}^\infty_{res}(x,x')\right| = \Theta(L^{-1}\log(L)),$$

We conclude by observing that $K^L_{res}(x,x')/(L-1)(1+\alpha)^{L-1} = K^L_{res}(x,x')/L(1+\alpha)^{L-1} + \mathcal{O}(L^{-1})$ 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*

$$y^l(x) = y^{l-1}(x) + \frac{1}{\sqrt{l}}\mathcal{F}(w^l, y^{l-1}(x)), \quad l \geq 2. \tag{19}$$

*where $\mathcal{F}$ is either a convolutional or dense layer (equations 3 and 4) with ReLU activation. Then there exists $\zeta, \nabla > 0$ such that for all $\epsilon \in (0,1)$*

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

*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.*

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

*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 $\alpha_l = \frac{\sigma_w^2}{2l}$. 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

$$c^l - c = c^{l-1} - c + \frac{f(c) - c}{l} + \mathcal{O}(\frac{c^l - c}{l}) + \mathcal{O}(l^{-2})$$

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

$$f(c) \underset{x\to 1-}{=} c + s(1-c)^{3/2} + b(1-c)^{5/2} + \mathcal{O}((1-c)^{7/2}) \tag{20}$$

where $s = \frac{2\sqrt{2}}{3\pi}$ and $b = \frac{\sqrt{2}}{30\pi}$. Letting $\gamma_l = 1 - c^l$, we obtain

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

which yields

$$\gamma_l^{-1/2} = \gamma_{l-1}^{-1/2} + \frac{s}{2}\delta_l + \frac{3}{8}s^2\delta_l^2\gamma_{l-1}^{1/2} + \frac{b}{2}\delta_l\gamma_{l-1} + O(\delta_l\gamma_{l-1}^{3/2}). \tag{21}$$

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

$$1 - c^l = \frac{\zeta}{\log(l)^2} - \frac{\nabla}{\log(l)^3} + o(\frac{1}{\log(l)^3})$$

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.

□

**Proposition 2** (Scaled Resnet). *Consider a Residual Neural Network with the following forward propagation equations*

$$y^l(x) = y^{l-1}(x) + \frac{1}{\sqrt{l}}\mathcal{F}(w^l, y^{l-1}(x)), \quad l \geq 2. \tag{22}$$

*where $\mathcal{F}$ is either a convolutional or dense layer (equations 3 and 4) with ReLU activation. Then the scaling factor $\alpha_L$ in Theorem 2 becomes $\alpha_L = L^{1+\sigma_w^2/2}$ and the convergence rate is $\Theta(\log(L)^{-1})$.*

*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 $\epsilon \in (0,1)$ and $x, x' \in B_\epsilon$ be two inputs. We first prove the result for the diagonal term $K^L_{res}(x,x)$ then $K^L_{res}(x,x')$.

- We have that $\dot{q}^l(x,x) = \frac{\sigma_w^2}{2l} f(1) = \frac{\sigma_w^2}{2l}$. Moreover, we have $q^l(x,x) = q^{l-1}(x,x) + \sigma_w^2/2l \times q^{l-1}(x,x) = [\prod_{k=1}^l (1+\sigma_w^2/2k)]\frac{\sigma_w^2}{d}\|x\|^2$. Recall that

$$K^l_{res}(x,x) = K^{l-1}_{res}(x,x)(1+\frac{\sigma_w^2}{2l}) + q^l(x,x)$$

  letting $k'_l = \frac{K^l_{res}(x,x)}{\prod_{k=1}^l (1+\sigma_w^2/2k)}$,we have that

$$k'_l = k'_{l-1} + \frac{\sigma_w^2}{d}\|x\|$$

  using the fact that $\prod_{k=1}^l (1+\sigma_w^2/2k) = \Theta(l^{\sigma_w^2/2})$, we conclude for $K^l_{res}(x,x)$.

- Recall that

$$K^l_{res}(x,x') = K^{l-1}_{res}(x,x')(\dot{q}^l(x,x')+1) + q^l(x,x')$$

  Let $c^l := c^l(x,x')$. From Appendix Lemma 10 we have that

$$1 - c^l = \frac{\zeta}{\log(l)^2} - \frac{\nabla}{\log(l)^3} + o(\frac{1}{\log(l)^3})$$

  $\zeta = \frac{16}{s^2\sigma_w^4}$ and $\nabla > 0$. Using the Taylor expansion of $f'$ as in Appendix Lemma 1, it follows that

$$f'(c^l(x,x')) = 1 - \frac{6}{\sigma_w^2}\log(l)^{-1} + \zeta' \log(l)^{-2} + \mathcal{O}(\log(l)^{-3})$$

  where $\zeta' = \frac{\nabla}{\sqrt{2\pi\zeta}}$. We obtain

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

  where $\zeta'' = \frac{\sigma_w^2}{2}\zeta'$. Letting $a_l = \frac{K^{l+1}_{res}(x,x')}{\prod_{k=1}^l (1+\sigma_w^2/2k)}$, we obtain

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

  where $\lambda_l = 1 - l^{-1} - 3l^{-1}\log(l)^{-1} + \mathcal{O}(l^{-1}\log(l)^{-2})$, $b_l = \sqrt{q^1(x,x)}\sqrt{q^1(x',x')}f(c^l(x,x')) = q(x,x') + \mathcal{O}(\log(l)^{-2})$ with $q = \sqrt{q^1(x,x)}\sqrt{q^1(x',x')}$ and where we used the fact that $c^l = 1 + \mathcal{O}(\log(l)^{-2})$ (Appendix Lemma 10).

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

$$x_{l-1}(1-\frac{1}{l}) - M_1 l^{-1}\log(l)^{-1} \leq x_l \leq x_{l-1}(1-\frac{1}{l}) - M_2 l^{-1}\log(l)^{-1}$$

therefore, there exists $l_0$ independent of $(x, x')$ such that for all $l \geq l_0$

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

and

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

after simplification, we have that

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

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

we conclude that $\alpha_L = L \times \prod_{k=1}^{l} (1 + \sigma_w^2/2k) \sim L^{1 + \frac{\sigma_w^2}{2}}$ 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 $B_\epsilon$.

In the limit of large $L$, the matrix NTK of the scaled resnet has the following form

$$A\hat{K}^{l}_{res} = qU + \log(L)^{-1} \Theta(M_L)$$

where $U$ is the matrix of ones, and $M_L$ has all elements but the diagonal equal to 1 and the diagonal terms are $\mathcal{O}(L^{-1} \log(L)) \to 0$. Therefore, $M_L$ is inversible for large $L$ which makes $\hat{K}^{l}_{res}$ 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.

□

### 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 $N(d, k) = \frac{2k+d-2}{k} \binom{k+d-3}{d-2}$.

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

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

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

$$P_k^d(t) = \big(-\frac{1}{2}\big)^k \frac{\Gamma(\frac{d-1}{2})}{\Gamma(k + \frac{d-1}{2})} (1 - t^2)^{\frac{3-d}{2}} \big(\frac{d}{dt}\big)^k (1 - t^2)^{k + \frac{d-3}{2}}$$

$(P_k^d)_{k \geq 0}$ 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

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

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

$$\kappa(x,x') = \sum_{k\geq 0}\sum_{j=1}^{N(d,k)}\left[\int_{\mathbb{S}^{d-1}} g(\langle w, x'\rangle)Y_{k,j}(w)d\nu_{d-1}(w)\right]Y_{k,j}(x)$$

Using the Hecke-Funk formula yields

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

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

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 ($AK^L$) for an FFNN with $L$ layers initialized on the EOC, or the Normalized NTK ($\bar{K}^L_{res}$) for a ResNet with $L$ layers (Fully Connected). Then, for all $L \geq 1$, there exists $(\mu_k^L)_{k\geq}$ such that for all $x, x' \in \mathbb{S}^{d-1}$*

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

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

*Moreover, we have that $0 < \mu_0^\infty = \lim_{L\to\infty}\mu_0^L < \infty$, and for all $k \geq 1$, $\lim_{L\to\infty}\mu_k^L = 0$.*

*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 $x, x' \in \mathbb{S}^{d-1}$

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

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

For $k = 0$, we have that for all $L \geq 1$, $\mu_0^L = \frac{\Omega_{d-1}}{\Omega_d}\int_{-1}^{1} g^L(t)(1-t^2)^{(d-3)/2}dt$. By a simple dominated convergence argument, we have that $\lim_{L\to\infty}\mu_0^L = q\lambda\frac{\Omega_{d-1}}{\Omega_d}\int_{-1}^{1}(1-t^2)^{(d-3)/2}dt > 0$, 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 $k \geq 1$, $\lim_{L\to\infty}\mu_k^L = q\lambda\frac{\Omega_{d-1}}{\Omega_d}\int_{-1}^{1} P_k^d(t)(1-t^2)^{(d-3)/2}dt = q\lambda\frac{\Omega_{d-1}}{\Omega_d}\langle P_0^d, P_k^d\rangle_{L^2([-1,1],(1-t^2)^{\frac{d-3}{2}}dt)} = 0$.

□

## 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. 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 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*.

## Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

| | |
|---|---|
| Title of Paper | Mean-field Behaviour of Neural Tangent Kernel for Deep Neural Networks |
| Publication Status | □ Published □ Accepted for Publication<br>■ Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style |
| Publication Details | Hayou, S., Doucet, A., and Rousseau, J. (2020). Mean-field Behaviour of Neural Tangent Kernel for Deep Neural Networks. arXiv preprint arXiv:1905.13654v8. |

### Student Confirmation

| | | | |
|---|---|---|---|
| Student Name: | Soufiane Hayou | | |
| Contribution to the Paper | I worked on the theory and proofs behind this paper. I also worked on the experiments. During our weekly meetings, my supervisors contributed to this work by providing valuable insights and helpful remarks. They also contributed a lot to the writing of the draft, checking the proofs, and proof-reading. | | |
| Signature  | | Date | 21/01/2021 |

### Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

| | | |
|---|---|---|
| Supervisor name and title: Arnaud Doucet | | |
| Supervisor comments<br>I am in agreement with the description of the contributions | | |
| Signature  | Date | 23/01/2021 |

This completed form should be included in the thesis, at the end of the relevant chapter.

# 4

# Stable Residual Neural Networks

# Stable ResNet

**Soufiane Hayou***[1] **Eugenio Clerico***[1] **Bobby He***[1] **George Deligiannidis**[1]

**Arnaud Doucet**[1] **Judith Rousseau**[1]

## Abstract

Deep ResNet architectures have achieved state of the art performance on many tasks. While they solve the problem of gradient vanishing, they might suffer from gradient exploding as the depth becomes large. Moreover, recent results have shown that ResNet might lose expressivity as the depth goes to infinity [Yang and Schoenholz, 2017, Hayou et al., 2019a]. To resolve these issues, we introduce a new class of ResNet architectures, called Stable ResNet, that have the property of stabilizing the gradient while ensuring expressivity in the infinite depth limit.

## 1 INTRODUCTION

The limit of infinite width has been the focus of many theoretical studies on Neural Networks (NNs) [Neal, 1995, Poole et al., 2016, Schoenholz et al., 2017, Yang and Schoenholz, 2017, Hayou et al., 2019a, Lee et al., 2019]. Although unachievable in practice, it features many interesting properties which can help grasp the complex behaviour of large networks.

Infinitely wide 1-layer random NNs behave like Gaussian Processes (GPs) at initialization [Neal, 1995]. This was recently extended to multilayer NNs, where each layer can be associated to its own GP [Matthews et al., 2018, Lee et al., 2018, Yang, 2019a]. From a theoretical point of view, GPs have the advantage that their behaviour is fully captured by the mean function and the covariance kernel. Moreover, when dealing with GPs that are equivalent to infinite width NNs, these processes are usually centered, and hence fully determined by their covariance kernel. For multilayer networks, these kernels can be computed recursively, layer by layer [Lee et al., 2018]. Interestingly, in apparent contradiction with the naive idea "the deeper, the more expressive", it was shown in [Schoenholz et al., 2017] that the GP becomes trivial as the number of layers goes to infinity, that is the output completely forgets about the input and hence lacks expressive power. This loss of input information during the forward propagation through the network might be exponential in depth and could lead to trainability issues for extremely deep nets [Schoenholz et al., 2017, Hayou et al., 2019a].

One natural way to prevent this last issue is the introduction of skip connections, commonly known as the ResNet architecture. However, in the regime of large width and depth, the output of standard ResNets becomes inexpressive and the network may suffer from gradient exploding [Yang and Schoenholz, 2017].

In the present work, we propose a new class of residual neural networks, the Stable ResNet, which, in the limit of infinite width and depth, is shown to stabilize the gradient (no gradient vanishing or exploding) and to preserve expressivity in the limit of large depth. The main idea is the introduction of layer/depth dependent scaling factors to the ResNet blocks.

For ReLU networks, we provide a comprehensive analysis of two different scalings: a uniform one, where the scaling factor is the same for all the layers, and a decreasing one, where the scaling factor decreases as we go deeper inside the network. We also show that Stable ResNet solve the problem of Neural Tangent kernel (NTK) degeneracy in the limit of large depth [Hayou et al., 2019b]; indeed, with our scalings, the NTK is universal in the limit of infinite depth, which ensures that any continuous function can be approximated to an arbitrary precision by the features of the infinite depth NTK on a compact set.

All theoretical results are substantiated with numerical experiments in Section 7, where we demonstrate the benefits of Stable ResNet scalings both for the corresponding infinite width GP kernels as well as trained ResNets, over a range of moderate and large-scale image classification tasks: MNIST, CIFAR-10, CIFAR-100 and TinyImageNet.

*Equal contribution [1]Department of Statistics, University of Oxford. Correspondence to: <soufiane.hayou;eugenio.clerico;bobby.he@stats.ox.ac.uk>.

Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS) 2021, San Diego, California, USA. PMLR: Volume 130. 

## 2 RESNET

### 2.1 Setup and Notations

Consider a standard ResNet architecture with $L+1$ layers, labelled with $l \in [0:L]$[1], of dimensions $\{N_l\}_{l\in[0:L]}$.

$$\begin{aligned} &y_0(x) = W_0\, x + B_0\,; \\ &y_l(x) = y_{l-1}(x) + \mathcal{F}((W_l, B_l), y_{l-1}(x)) \quad \text{for } l \in [1:L]\,, \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^d$ is an input, $y_l(x) = \{y_l^i(x)\}_{i\in[1:N_l]}$ is the vector of pre-activations, $W_l$ and $B_l$ are respectively the weights and bias of the $l^{th}$ layer, and $\mathcal{F}$ is a mapping that defines the nature of the layer. In general, the mapping $\mathcal{F}$ consists of successive applications of simple linear maps (including convolutional layers), normalization layers [Ioffe and Szegedy, 2015] and activation functions. In this work, for the sake of simplicity, we consider Fully Connected blocks with ReLU activation function:

$$\mathcal{F}((W, B), x) = W\phi(x) + B\,,$$

where $\phi$ is the activation function. The weights and bias are initialized with $W_l \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_w^2/N_{l-1})$, and $B_l \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_b^2)$, where $\sigma_w > 0$, $\sigma_b \geq 0$, $N_{-1} = d$, and $\mathcal{N}(\mu, \sigma^2)$ is the normal law of mean $\mu$ and variance $\sigma^2$.

Recent results by [Hayou et al., 2021] suggest that scaling the residual blocks with $L^{-1/2}$ might have some beneficial properties on model pruning at initialization. This results from the stabilization effect on the gradient due to the scaling.

More generally, we introduce the residual architecture:

$$\begin{aligned} &y_0(x) = W_0\, x + B_0\,; \\ &y_l(x) = y_{l-1}(x) + \lambda_{l,L}\, \mathcal{F}((W_l, B_l), y_{l-1})\,, \quad l \in [1:L]\,, \end{aligned} \tag{2}$$

where $\{\lambda_{l,L}\}_{l\in[1:L]}$ is a sequence of scaling factors. We assume hereafter that there exists $\lambda_{\max} \in (0, \infty)$ such that $\lambda_{l,L} \in (0, \lambda_{\max}]$ for all $L \geq 1$ and $l \in [1:L]$.
In the next proposition, we give a necessary and sufficient condition for the gradient to remain bounded as the depth $L$ goes to infinity.

**Proposition 1** (Stable Gradient). *Consider a ResNet of type (2), and let $\mathcal{L}_y(x) := \ell(y_L^1(x), y)$ for some $(x, y) \in \mathbb{R}^d \times \mathbb{R}$, where $\ell : (z, y) \mapsto \ell(z, y)$ is a loss function satisfying $\sup_{K_1\times K_2} \left|\frac{\partial \ell(z,y)}{\partial z}\right| < \infty$, for all compacts $K_1, K_2 \subset \mathbb{R}$. Then, in the limit of infinite width, for any compacts $K \subset \mathbb{R}^d$, $K' \subset \mathbb{R}$, there exists a constant $C > 0$ such that for all $(x, y) \in K \times K'$*

$$\sup_{l\in[0:L]} \mathbb{E}\left[\left|\frac{\partial \mathcal{L}_y(x)}{\partial W_l^{11}}\right|^2\right] \leq C \exp\left(\frac{\sigma_w^2}{2} \sum_{l=1}^L \lambda_{l,L}^2\right).$$

*Moreover, if there exists $\lambda_{\min} > 0$ such that for all $L \geq 1$ and $l \in [1:L]$ we have $\lambda_{l,L} \geq \lambda_{\min}$, then, for all $(x, y) \in (\mathbb{R}^d \setminus \{0\}) \times \mathbb{R}$ such that $\left|\frac{\partial \ell(z,y)}{\partial z}\right| \neq 0$, there exists $\kappa > 0$ such that for all $l \in [1:L]$*

$$\mathbb{E}\left[\left|\frac{\partial \mathcal{L}_y(x)}{\partial W_l^{11}}\right|^2\right] \geq \kappa \left(1 + \frac{\lambda_{\min}^2 \sigma_w^2}{2}\right)^L.$$

Proposition 1 shows that in order to stabilize the gradient, we have to scale the blocks of the ResNet with scalars $\{\lambda_{l,L}\}_{l\in[1:L]}$ such that $\sum_{l=1}^L \lambda_{l,L}^2$ remains bounded as the depth $L$ goes to infinity. Taking $\lambda_{\min} = 1$, Proposition 1 shows that the standard ResNet architecture (1) suffers from gradient exploding at initialization,[2] which may cause instability during the first step of gradient based optimization algorithms such as Stochastic Gradient Descent (SGD). This motivates the following definition of Stable ResNet.

**Definition 1** (Stable ResNet). *A ResNet of type (2) is called a Stable ResNet if and only if $\lim_{L\to\infty} \sum_{l=1}^L \lambda_{l,L}^2 < \infty$.*

The condition on the scaling factors is satisfied by a wide range of sequences $\{\lambda_{l,L}\}_{l\in[1:L],L\geq 1}$. However, it is natural to consider the two categories:
**Uniform scaling.** The scaling factors have similar magnitude and tend to zero at the same time. A simple example is the uniform scaling $\lambda_{l,L} = 1/\sqrt{L}$.
**Decreasing scaling.** The sequence is decreasing and tends to zero. To be clearer, we consider a general sequence $\{\lambda_l\}_{l\in[1:L]}$ such that $\sum_{l\geq 1} \lambda_l^2 < \infty$, and let $\lambda_{l,L} = \lambda_l$ for all $L \geq 1$, all $l \in [1:L]$.

Note that our theoretical analyses will hold for any decreasing scaling $\{\lambda_l\}_{l\geq 1}$ that is square summable, but for simplicity in all empirical results we consider the decreasing scaling:

$$\lambda_l^{-1} = l^{1/2} \times \log(l+1)\,.$$

We study theoretical properties of both ResNets with uniform and decreasing scaling. We show that, in addition to stabilizing the gradient, both scalings ensure that the ResNet is expressive in the infinite depth limit. For this purpose, we use a tool known as Neural Network Gaussian Process (NNGP) [Lee et al., 2018] which is the equivalent Gaussian Process of a Neural Network in limit of infinite width.

### 2.2 On Gaussian Process approximation of Neural Networks

Consider a ResNet of type (2). Neurons $\{y_0^i(x)\}_{i\in[1:N_1]}$ are iid since the weights with which they are connected

[1] Notation: $[m:n] = \{m, m+1 \dots n\}$ for integers $n \geq m$.

[2] In [Yang and Schoenholz, 2017], authors show a similar result with a slightly different ResNet architecture.

to the inputs are iid. Using the Central Limit Theorem, as $N_0 \to \infty$, $y_1^i(x)$ is a Gaussian variable for any input $x$ and index $i \in [1 : N_1]$. Moreover, the variables $\{y_1^i(x)\}_{i\in[1:N_1]}$ are iid. Therefore, the processes $y_1^i(.)$ can be seen as independent (across $i$) centred Gaussian processes with covariance kernel $Q_1$. This is an idealized version of the true process corresponding to letting width $N_0 \to \infty$. Doing this recursively over $l$ leads to similar approximations for $y_l^i(.)$ where $l \in [1 : L]$, and we write accordingly $y_l^i \overset{ind}{\sim} \mathcal{GP}(0, Q_l)$. The approximation of $y_l^i(.)$ by a Gaussian process was first proposed by [Neal, 1995] in the single layer case and was extended to multiple feedforward layers by [Lee et al., 2019] and [Matthews et al., 2018]. More recently, a powerful framework, known as Tensor Programs, was proposed by [Yang, 2019b], confirming the large-width NNGP association for nearly all NN architectures.

For any input $x \in \mathbb{R}^d$, we have $\mathbb{E}[y_l^i(x)] = 0$, so that the covariance $Q_l(x, x') = \mathbb{E}[y_l^1(x)y_l^1(x')]$ satisfies for all $x, x' \in \mathbb{R}^d$ (see Appendix A1)

$$Q_l(x, x') = Q_{l-1}(x, x') + \lambda_{l,L}^2 \Psi_{l-1}(x, x'),$$

where $\Psi_{l-1}(x, x') = \sigma_b^2 + \sigma_w^2 \mathbb{E}[\phi(y_{l-1}^1(x))\phi(y_{l-1}^1(x'))]$.

For the ReLU activation function $\phi : x \mapsto \max(0, x)$, the recurrence relation can be written more explicitly as in [Daniely et al., 2016]. Let $C_l$ be the correlation kernel, defined as

$$C_l(x, x') = \frac{Q_l(x,x')}{\sqrt{Q_l(x,x)Q_l(x',x')}} \tag{3}$$

and let $f : [-1, 1] \to \mathbb{R}$ be given by

$$f : \gamma \mapsto \tfrac{1}{\pi}(\sqrt{1-\gamma^2} - \gamma \arccos \gamma). \tag{4}$$

The recurrence relation reads (see Appendix A1)

$$\begin{aligned} Q_l &= Q_{l-1} + \lambda_{l,L}^2 \left[\sigma_b{}^2 + \tfrac{\sigma_w{}^2}{2}\left(1 + \tfrac{f(C_{l-1})}{C_{l-1}}\right) Q_{l-1}\right], \\ Q_0(x, x') &= \sigma_b^2 + \sigma_w^2 \tfrac{x\cdot x'}{d}. \end{aligned} \tag{5}$$

This recursion leads to divergent diagonal terms $Q_L(x, x)$. This was proven in [Yang and Schoenholz, 2017] for a slightly different ResNet architecture. In the next Lemma, we extend this result to the ResNet defined by (1).

**Lemma 1** (Exploding kernel with standard ResNet)**.** *Consider a ResNet of type (1). Then, for all $x \in \mathbb{R}^d$,*

$$Q_L(x, x) \geq \left(1 + \tfrac{\sigma_w^2}{2}\right)^L \left(\sigma_b^2\left(1 + \tfrac{2}{\sigma_w^2}\right) + \tfrac{\sigma_w^2}{d}\|x\|^2\right).$$

Figure 1 plots the diagonal NNGP and NTK (introduced in Section 5) values for a point on the sphere,

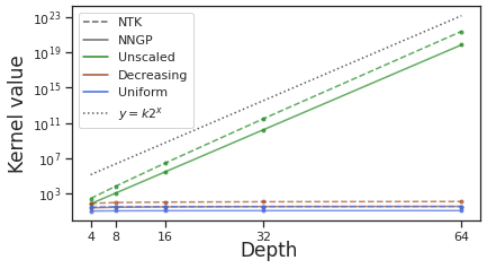



**Figure 1:** NNGP/NTK for unscaled ResNets explode exponentially (with base 2 if $\sigma_w^2 = 2$) in depth, unlike (both uniform and decreasing scaled) Stable ResNets.

highlighting the exploding kernel problem for standard ResNets. Stable ResNets do not suffer from this problem.

We now introduce further notation and definitions. Hereafter, unless specified otherwise, $K$ will denote a compact set in $\mathbb{R}^d$ ($d \geq 1$) and $x, x'$ denote two arbitrary elements of $K$.
Let us start with a formal definition of a kernel.[3]

**Definition 2** (Kernel)**.** *A kernel $Q$ on $K$ is a symmetric continuous function $K^2 \to \mathbb{R}$ such that, for all $n \in \mathbb{N}$, for any finite subset $\{x_1 \dots x_n\} \subset K$, the matrix $\{Q(x_i, x_j)\}_{i,j}$ is non-negative definite.*

The symmetry in the above definition has to be understood as $Q(x, x') = Q(x', x)$ for all $x, x' \in K$.

Kernels induce non-negative integral operators [Paulsen and Raghupathi, 2016].

**Lemma 2.** *Given a continuous and symmetric function $Q : K^2 \to \mathbb{R}$, we can define the induced integral operator $T(Q)$ on $L^2(K)$ via its action $T(Q)\varphi(x) = \int_K Q(x, y)\varphi(y)\,\mathrm{d}y$, for $\varphi \in L^2(K)$.[4] Moreover, $T(Q)$ is a bounded, compact, non-negative definite self-adjoint operator.*

Each kernel induces a centred Gaussian Process on $K$ [Dudley, 2002], that is a random function $F$ on $K$ such that, for any finite $\hat{K} \subset K$, $\{F(x)\}_{x\in\hat{K}}$ is a centred Gaussian vector. We recall that the law of a centred GP is fully determined by its covariance function $(x, x') \mapsto \mathbb{E}[F(x)F(x')]$, defined on $K^2$.

**Definition 3** (Induced GP)**.** *Given a kernel $Q$ on $K$, the Gaussian Process induced by $Q$ is a centred GP on $K$ whose covariance function is $Q$.*

We will sometimes use the notation $\mathcal{GP}(0, Q)$ for the law of the GP induced by a kernel $Q$. With our definition

[3] Our definition is not the standard definition of a kernel, which is more general and does not require the continuity, [Paulsen and Raghupathi, 2016].

[4] Naturally, we should write $L^2(K, \mu)$, specifying a measure $\mu$ on $K$. In the present work, unless otherwise specified, the notation $L^2(K)$ will imply the choice of any arbitrary finite Borel measure on $K$ (cf Appendix A0).

of a kernel, the samples from the induced GP lies in $L^2(K)$ with probability 1 [Steinwart, 2019].

From now on we will assume that $0 \notin K$ if $\sigma_b = 0$.[5] For all ResNets, it is straightforward to check that $Q_L$ is a kernel, in the sense of Definition 2 (see Appendix A1 or [Daniely et al., 2016]). The induced Gaussian Process is what we refer to as NNGP.

We denote by $\mathcal{H}_Q(K)$ the Reproducing Kernel Hilbert Space (RKHS)[6] induced by the kernel $Q$ on the set $K$. The following hierarchical result holds.

**Proposition 2.** *For all $L \geq 1$, $l \in [0, L-1]$, $\mathcal{H}_{Q_l}(K) \subseteq \mathcal{H}_{Q_{l+1}}(K)$.*

Proposition 2 shows that, as we go deeper, the RKHS cannot become poorer. However, increasing $L$ might introduce stability issues as illustrated in Proposition 1. We show in Sections 3 and 4 that Stable ResNets resolve this problem.

By Lemma 2, $T(Q_L)$ is a bounded, compact, self-adjoint operator and hence can be written as the sum of the projections on its eigenspaces [Lang, 2012]. By Mercer's Theorem [Paulsen and Raghupathi, 2016], all the eigenfunctions of $T(Q_L)$ are continuous. Finally, it is possible to link the eigen-decomposition of $T(Q_L)$ with the distribution of the GP induced by $Q_L$. Denoting respectively by $\mu_k$ and $\psi_k$ the eigenvalues and eigenfunctions of the operator $T(Q_L)$, we have the equivalence in law:

$$y_L^1 \sim \sum_{k \in \mathbb{N}} \sqrt{\mu_k} Z_k \psi_k \sim \mathcal{GP}(0, Q_L)\,, \tag{6}$$

where $\{Z_k\}_{k \geq 0}$ are i.i.d. standard Gaussian random variables [Grenander, 1950]. The expressivity, that is the capacity to approximate a large class of function, of the network at initialization is then closely linked to the eigendecomposition of $Q_L$ [Yang and Salman, 2019].

### 2.3 Universal kernels and expressive GPs

In this section, we provide a comprehensive study of the kernel $Q_L$. We start with a formal definition of universality ($c$-universality in [Sriperumbudur et al., 2011]). Again, unless otherwise stated, let $K$ be a compact in $\mathbb{R}^d$.

**Definition 4** (Universal Kernel)**.** *Let $Q$ be a kernel on $K$, and $\mathcal{H}_Q(K)$ its RKHS*[7]*. We say that $Q$ is universal on $K$ if for any $\varepsilon > 0$ and any continuous function $g$ on $K$, there exists $h \in \mathcal{H}_Q(K)$ such that $\|h - g\|_\infty < \varepsilon$.*

The universality of a kernel $Q$ on a compact set implies that the kernel is strictly positive definite, i.e. for all non-zero $\varphi \in L^2(K)$, $\langle T(Q)\varphi, \varphi \rangle > 0$ [Sriperumbudur et al., 2011]. Moreover, universality also implies the full expressivity of the induced GP, as expressed in the following.

**Definition 5** (Expressive GP)**.** *A Gaussian Process on $K$ is said to be expressive on $L^2(K)$ if, denoting by $\psi$ a random realisation $\psi$ of the process, for all $\varphi \in L^2(K)$, for all $\varepsilon > 0$,*

$$\mathbb{P}(\|\psi - \varphi\|_2 \leq \varepsilon) > 0\,.$$

**Lemma 3.** *A universal kernel $Q$ on $K$ induces an expressive GP on $L^2(K)$.*

By definition, universal kernels are characterized by the property that their associated RKHS is dense (w.r.t the uniform norm $\|.\|_\infty$) in the space of continuous functions on $K$. This is crucial for Kernel regression and Gaussian Process inference [Kanagawa et al., 2018].[8] By Proposition 2, it suffices to prove that $Q_{L_0}$ is universal for some $L_0$ in order to conclude for all $L \geq L_0$. It turns out this is true for $L_0 = 2$.

**Proposition 3.** *If $\sigma_b > 0$, then $Q_2$ is universal on $K$. From Proposition 2, $Q_L$ is universal for all $L \geq 2$.*

Note that the presence of biases is essential to achieve universality in the case of a general $K$, since the output of a ReLU ResNet with no bias is always a positive homogeneous function of its input, i.e., a map $F$ such that $F(\alpha x) = \alpha F(x)$ for all $\alpha \geq 0$. However, in the particular case of $K = \mathbb{S}^{d-1}$, the unit sphere in $\mathbb{R}^d$, the kernel $Q_L$ is universal (for $L \geq 2$), even when $\sigma_b = 0$.

**Proposition 4.** *Assume $\sigma_b = 0$. Then for all $L \geq 2$, $Q_L$ is universal on $\mathbb{S}^{d-1}$ for $d \geq 2$.*

Another interesting fact of the case $K = \mathbb{S}^{d-1}$ is that the eigendecomposition of the kernel $Q_L$ has a simple structure. Indeed, on $\mathbb{S}^{d-1}$, $Q_L(x, x')$ depends only on the scalar product $x \cdot x'$. These kernels (zonal kernel) admit Spherical Harmonics as an eigenbasis [Yang and Salman, 2019].

**Proposition 5** (Spectral decomposition on $\mathbb{S}^{d-1}$)**.** *Let $Q$ be a zonal kernel on $\mathbb{S}^{d-1}$, that is $Q(x, x') = p(x \cdot x')$ for a continuous function $p : [-1, 1] \to \mathbb{R}$. Then, there is a sequence $\{\mu_k \geq 0\}_{k \in \mathbb{N}}$ such that for all $x, x' \in \mathbb{S}^{d-1}$*

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

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

[5] We exclude 0 since for $\sigma_b = 0$ $C_0$ is discontinuous in 0 and can't be a kernel on $K$ as in Definition 2, if $0 \in K$.

[6] See Appendix A0 for a definition.

[7] See Appendix A0.

[8] The closure of the set of functions described by the mean function of the posterior of a GP regression is exactly the RKHS of the kernel of the GP prior.

*k. With respect to the standard spherical measure, the spherical harmonics form an orthonormal basis of $L^2(\mathbb{S}^{d-1})$ and $T(Q)$ is diagonal on this basis.*

Although the kernel is universal for fixed depth $L$, it is not guaranteed that as $L \to \infty$, $Q_L$ remains universal. Indeed, for the standard ResNet architecture, the variance $Q_L(x,x)$ grows exponentially with $L$ [Yang and Schoenholz, 2017], and therefore, the kernel diverges. In order to analyse the expressivity of the kernel of a standard ResNet in the limit of large depth, we can study the correlation kernel $C_L$, defined in (3), instead. We show in the following Lemma that, as $L$ goes to infinity, the kernel $C_L$ converges to a constant (which has a 1D RKHS).

**Lemma 4.** *Consider a standard ResNet of type (1) and let $K \subset \mathbb{R}^d \setminus \{0\}$ be a compact set. We have that*

$$\lim_{L\to\infty} \sup_{x,x'\in K} |1 - C_L(x,x')| = 0\,.$$

*Moreover, if $\sigma_b = 0$, then,*

$$\sup_{x,x'\in K} |1 - C_L(x,x')| = \mathcal{O}(L^{-2})\,.$$

*Therefore, $\mathcal{H}_{C_\infty}(K)$ is the space of constant functions.*

Lemma 4 shows that in the limit of infinite depth $L$, the RKHS of the correlation kernel is trivial, meaning that the NNGP cannot be expressive. On the contrary, we will show in the next sections that Stable ResNets achieve a universal kernel for infinite depth $L$.

# 3 UNIFORM SCALING

Consider a Stable ResNet with layers $[0:L]$. Under uniform scaling, the recurrence relation in (5) reads:

$$Q_l = Q_{l-1} + \tfrac{1}{L}\left({\sigma_b}^2 + \tfrac{{\sigma_w}^2}{2}\left(1 + \tfrac{f(C_{l-1})}{C_{l-1}}\right) Q_{l-1}\right). \quad (7)$$

In the limit $L \to \infty$, (7) converges uniformly to a continuous ODE. Studying the solution of the corresponding Cauchy problem, we show that the covariance kernel remains universal in the limit of infinite depth.

## 3.1 Continuous formulation

The layer index $l$ in (7) can be rescaled as $l \mapsto t(l) = l/L$. Clearly $t(0) = 0$ and $t(L) = 1$, so the image of $t$ is contained in $[0,1]$. In the limit $L \to \infty$ it is natural to consider $t$ as a continuous variable spanning the interval $[0,1]$. With this in mind, it makes sense to look at the continuous version of (7).
Let $K \subset \mathbb{R}^d$ be a compact set and $x, x' \in K$. If $\sigma_b = 0$ assume that $0 \notin K$.

$$\begin{aligned} \dot{q}_t(x,x') &= {\sigma_b}^2 + \tfrac{{\sigma_w}^2}{2}\left(1 + \tfrac{f(c_t(x,x'))}{c_t(x,x')}\right) q_t(x,x')\,, \\ q_0(x,x') &= \sigma_b^2 + \sigma_w^2\,\tfrac{x\cdot x'}{d}\,, \\ c_t(x,x') &= \tfrac{q_t(x,x')}{\sqrt{q_t(x,x)q_t(x',x')}}\,. \end{aligned} \quad (8)$$

As discussed in Section A2 of the Appendix, for any $x, x'$, the solution of the above Cauchy problem exists and is unique. Moreover, the solutions $q_t$ and $c_t$ are kernels on $K$, in the sense of Definition 2.

Clearly, for finite $L$, the continuous ODE (8) is an approximation. However, the following result holds.

**Lemma 5** (Convergence to the continuous limit)**.** *Let $Q_{l|L}$ be the covariance kernel of the layer $l$ in a net of $L+1$ layers $[0:L]$, and $q_t$ be the solution of (8), then*

$$\lim_{L\to\infty} \sup_{l\in[0:L]} \sup_{(x,x')\in K^2} |Q_{l|L}(x,x') - q_{t=l/L}(x,x')| = 0\,.$$

## 3.2 Universality of the covariance kernel

When $\sigma_b > 0$, the kernel $q_t$ is universal for $t > 0$.

**Theorem 1** (Universality of $q_t$)**.** *Let $K \subset \mathbb{R}^d$ be compact and assume $\sigma_b > 0$. For any $t \in (0,1]$, the solution $q_t$ of (8) is a universal kernel on $K$.*

The proof of the above statement is detailed in Appendix A2. The main idea is to show that the integral operator $T(q_t)$ is strictly positive definite and then use a characterization of universal kernels, due to [Sriperumbudur et al., 2011], which connects the universality of Definition 4 with the strict positivity of the induced integral operator.[9]

As mentioned previously, the presence of the bias is essential to achieve full expressivity on a generic compact $K \subset \mathbb{R}^d$. However, we can still have universality when no bias is present, limiting ourselves to the case of the unit sphere $K = \mathbb{S}^{d-1}$.

**Proposition 6** (Universality on $\mathbb{S}^{d-1}$)**.** *For any $t \in (0,1]$, the covariance kernel $q_t$, solution of (8) with $\sigma_b = 0$, is universal on $\mathbb{S}^{d-1}$, with $d \geq 2$.*

# 4 DECREASING SCALING

Consider a Stable ResNet with decreasing scaling, that is a sequence of scaling factors $(\lambda_k)_{k\geq 1}$ such that $\sum_{k\geq 1} \lambda_k^2 < \infty$. In this setting, each additional layer can be seen as a correction to the network output with decreasing magnitude. As for the uniform scaling, we

[9] The details are more involved as we need to show that the kernel induces a strictly positive definite operator on $L^2(K,\mu)$ for any finite Borel measure $\mu$ on $K$.

show in the next proposition that the kernel $Q_L$ converges to a limiting kernel $Q_\infty$, and the convergence is uniform over any compact set of $\mathbb{R}^d$. The notation $g(x) = \Theta(m(x))$ means there exist two constants $A, B > 0$ such that $Am(x) \leq g(x) \leq Bm(x)$.

**Proposition 7** (Uniform Convergence of the Kernel)**.** *Consider a Stable ResNet with a decreasing scaling, i.e. the sequence $\{\lambda_l\}_{l\geq 1}$ is such that $\sum_l \lambda_l^2 < \infty$. Then for all $(\sigma_b, \sigma_w) \in \mathbb{R}^+ \times (\mathbb{R}^+)^*$, there exists a kernel $Q_\infty$ on $\mathbb{R}^d$ such that for any compact set $K \subset \mathbb{R}^d$,*

$$\sup_{x,x'\in K} |Q_L(x,x') - Q_\infty(x,x')| = \Theta\big(\textstyle\sum_{k\geq L} \lambda_k^2\big)\,.$$

The convergence of the kernel $Q_L$ to the limiting kernel $Q_\infty$ is governed by the convergence rate of the series of scaling factors. Moreover, leveraging the RKHS hierarchy from Proposition 2, we find that $Q_\infty$ is universal.

**Corollary 1** (Universality of $Q_\infty$)**.** *The following statements hold*

- *Let $K$ be a compact set of $\mathbb{R}^d$ and assume $\sigma_b > 0$. Then, $Q_\infty$ is universal on $K$.*
- *Assume $\sigma_b = 0$. Then $Q_\infty$ is universal on $\mathbb{S}^{d-1}$.*

As in the uniform scaling case, the limiting kernel exists and is universal unlike the standard ResNet architecture that yields a divergent kernel $Q_L$ as $L \to \infty$.

To validate our universality and expressivity results, Figure 2 plots the leading eigenvalues of the NNGP (& NTK, introduced in Section 5) kernels on a set of 1000 points sampled uniformly at random from the circle, normalized so that the largest eigenvalue is 1. We use the recursion formulas for NNGP correlation (Lemma A4) and normalized NTK (Lemma A19) to avoid the exploding variance/gradient problem. We see that the unscaled ResNet NNGP becomes inexpressive with depth because all non-leading eigenvalues converge to 0, whereas our Stable ResNets (decreasing and uniform scaling) are expressive even in the large depth limit.

## 5 NEURAL TANGENT KERNEL

In the so-called lazy training regime [Chizat and Bach, 2019], the training dynamics of an infinitely wide network can be described via the Neural Tangent Kernel (NTK) [Lee et al., 2019], introduced in [Jacot et al., 2018] and defined as

$$\tilde{\Theta}_L^{ij}(x,x') = \nabla_{\text{par}}\, y_L^i(x) \cdot \nabla_{\text{par}}\, y_L^j(x')\,,$$

with $\nabla_{\text{par}}$ the gradient wrt the parameters of the NN.[10] To simplify our presentation we will assume that the output dimension of the network is 1.[11]

[10] All network considered in this section are assumed to have NTK parametrization, cf Appendix A4 for details.

[11] This does not affect our final conclusion of universality for the NTK, which is diagonal in the output space, that is $\tilde{\Theta}^{ij} = \Theta\delta^{ij}$, [Jacot et al., 2018, Hayou et al., 2019b].

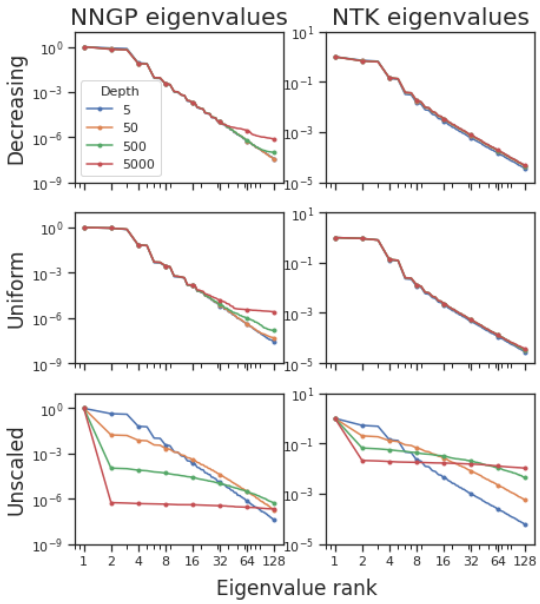



**Figure 2:** (Normalized) NNGP & NTK matrix eigenvalues of Stable (decreasing & uniform) & unscaled (i.e. standard) ResNets.

Let $F_\tau$ be the output function of the ResNet at training time $\tau$. In the NTK regime (infinite width), the gradient flow is equivalent to a simple linear model [Lee et al., 2019], that gives

$$F_\tau(x) - F_0(x) = \Theta_L(x,\mathcal{X})\hat{\Theta}_L^{-1}(I - e^{-\eta\hat{\Theta}_L\tau})(\mathcal{Y} - F_0(\mathcal{X}))\,,$$

where $\mathcal{X}$ and $\mathcal{Y}$ are respectively the input and output datasets, $\Theta_L(x,\mathcal{X}) = \{\Theta_L(x,x')\}_{x'\in\mathcal{X}}$ and $\hat{\Theta}_L$ is the matrix $\{\Theta_l(x,x')\}_{x,x'\in\mathcal{X}}$. The universality of the NTK is crucial for the ResNet to learn beyond initialization, since the residual $F_\tau - F_0$ lies in the RKHS generated by $\Theta_L$. For unscaled ResNet, [Hayou et al., 2019b] showed that the limiting NTK is trivial in the sense of Lemma 4. However, this is not the case for Stable ResNet.

Consider a ResNet of type (2). We have [12]

$$\Theta_0 = Q_0,\;\; \Theta_{l+1} = \Theta_l + \lambda_{l,L}^2\,(\Psi_l + \Psi_l'\,\Theta_l)\;, \qquad (9)$$

where $\Psi_l(x,x') = \sigma_b^2 + \sigma_w^2\mathbb{E}[\phi(y_1^l(x))\phi(y_1^l(x'))]$ and $\Psi_l'(x,x') = \sigma_w^2\mathbb{E}[\phi'(y_1^l(x))\phi'(y_1^l(x'))]$ (see Appendix A4).

**Proposition 8.** *Fix a compact $K \subset \mathbb{R}^d$ ($0 \notin K$ if $\sigma_b = 0$) and consider a Stable ResNet with decreasing scaling. Then $\Theta_L$ converges uniformly over $K^2$ to a kernel $\Theta_\infty$. Moreover $\Theta_\infty$ is universal on $K$ if $\sigma_b > 0$. If $K = \mathbb{S}^{d-1}$, then the universality holds for $\sigma_b = 0$.*

[12] This is true under the technical assumption that the parameters appearing in the back-propagation can be considered independent from the ones of the forward pass (Gradient Independent Assumption) [Yang, 2019a]

An analogous result can be stated for the uniform scaling, after noticing that a continuous formulation $(\Theta_l \mapsto \theta_{t(l)})$ can be obtained in analogy with what has been done for the covariance kernel (cf Appendix A4).

**Proposition 9.** *Let $K \subset \mathbb{R}^d$ and fix $t \in (0,1]$. If $\sigma_b > 0$, then $\theta_t$ is universal on $K$. The same holds true if $\sigma_b = 0$ and $K = \mathbb{S}^{d-1}$.*

Figure 2 shows that the non-leading NTK eigenvalues do not decay to 0 with depth for Stable ResNets, unlike for unscaled ResNets. This is in line with findings of Propositions 8 and 9.

## 6 A PAC-BAYES RESULT

Consider a dataset $S$ with $N$ iid training examples $(x_i, y_i)_{1\leq i\leq N} \in X \times Y$, and a hypothesis space $\mathcal{P}$ from which we want to learn an optimal hypothesis according to some bounded loss function $\ell : Y \times Y \mapsto [0,1]$. The empirical/generalization loss of a hypothesis $h \in \mathcal{U}$ are

$$r_S(h) = \frac{1}{N}\sum_{i=1}^N \ell(h(x_i), y_i), \quad r(h) = \mathbb{E}_\nu[\ell(h(x), y)]\,,$$

where $\nu$ is a probability distribution on $X \times Y$. For some randomized learning algorithm $\mathcal{A}$, the empirical and generalization loss are given by:

$$r_S(\mathcal{A}) = \mathbb{E}_{h\sim\mathcal{A}}[r_S(h)], \qquad r(\mathcal{A}) = \mathbb{E}_{h\sim\mathcal{A}}[r(h)]\,.$$

The PAC-Bayes theorem gives a probabilistic upper bound on the generalization loss $r(\mathcal{A})$ of a randomized learning algorithm $\mathcal{A}$ in terms of the empirical loss $r_S(\mathcal{A})$. Fix a prior distribution $\mathcal{P}$ on the hypothesis set $\mathcal{U}$. The Kullback-Leibler divergence between $\mathcal{A}$ and $\mathcal{P}$ is defined as $\mathrm{KL}(\mathcal{A}\|\mathcal{P}) = \int \mathcal{A}(h)\log\frac{\mathcal{A}(h)}{\mathcal{P}(h)}\mathrm{d}h \in [0,\infty]$. The Bernoulli KL-divergence is given by $\mathrm{kl}(a||p) = a\log\frac{a}{p} + (1-a)\log\frac{1-a}{1-p}$ for $a, p \in [0,1]$. We define the inverse Bernoulli KL-divergence $\mathrm{kl}^{-1}$ by

$$\mathrm{kl}^{-1}(a,\varepsilon) = \sup\{p \in [0,1] : \mathrm{kl}(a||p) \leq \varepsilon\}\,.$$

**Theorem 2** (PAC-Bayes bound Theorem [Seeger, 2002])**.** *For any loss function $\ell$ that is $[0,1]$ valued, any distribution $\nu$, any $N \in \mathbb{N}$, any prior $\mathcal{P}$, and any $\delta \in (0,1]$, with probability at least $1-\delta$ over the sample $S$, we have*

$$\forall \mathcal{A}, \quad r(\mathcal{A}) \leq \mathrm{kl}^{-1}\left(r_S(\mathcal{A}), \frac{\mathrm{KL}(\mathcal{A}\|\mathcal{P}) + \log(2\sqrt{N}/\delta)}{N}\right)\,.$$

The KL-divergence term $\mathrm{KL}(\mathcal{A}\|\mathcal{P})$ plays a major role as it controls the generalization gap, i.e. the difference (in terms of Bernoulli KL-divergence) between the empirical loss and the generalization loss. In our setting, we consider an ordinary GP regression with prior $\mathcal{P}(f) = \mathcal{GP}(f|0, Q(x,x'))$. Under the standard assumption that the outputs $y_N = (y_i)_{i\in[1:N]}$ are noisy versions of $f_N = (f(x_i))_{i\in[1:N]}$ with $y_N|f_N \sim \mathcal{N}(y_N|f_N, \sigma^2 I)$, the Bayesian posterior $\mathcal{A}$ is also a GP and is given by

$$\begin{aligned}\mathcal{A}(f) = \mathcal{GP}(f|&Q_N(x)(Q_{NN}+\sigma^2 I)^{-1}y_N, Q(x,x')\\ &- Q_N(x)(Q_{NN}+\sigma^2 I)^{-1})Q_N(x')^T)\,.\end{aligned} \tag{10}$$

$Q_N(x) = (Q(x,x_i))_{i\in[1:N]}$, $Q_{NN} = (Q(x_i,x_j))_{1\leq i,j\leq N}$. In this setting, we have the following result

**Proposition 10** (Curse of Depth)**.** *Let $Q_L$ be the kernel of a ResNet. Let $P_L$ be a GP with kernel $Q_L$ and $\mathcal{A}_L$ be the corresponding Bayesian posterior for some fixed noise level $\sigma^2 > 0$. Then, in a fixed setting (fixed sample size N), the following results hold:*

- *With a standard ResNet, $\mathrm{KL}(\mathcal{A}_L\|P_L) \gtrsim L$.*
- *With a Stable ResNet, $\mathrm{KL}(\mathcal{A}_L\|P_L) = \mathcal{O}_L(1)$.*

The KL-divergence bound diverges for a standard ResNet while it remains bounded for Stable ResNet. Although PAC-Bayes bounds only give an upper bound on the generalization error, Proposition 10 shows that Stable ResNet does not suffer from the "curse of depth", i.e. the KL-divergence does not explode as the depth becomes large.

## 7 EXPERIMENTS

In line with our theory, we now present results demonstrating empirical advantages of Stable ResNets (both uniform and decreasing scaling) compared to their unscaled counterparts on a toy regression task and standard image classification tasks, both for infinite-width NNGP kernels as well as trained finite-width NNs in the latter case. In the interests of space, all experimental details not described in this section can be found in Appendix A7. All error bars in this section correspond to 3 independent runs.

**Stable NNGP regression experiment** We first present a toy regression posterior regression experiment with NNGP kernel. We compare across different depths and scalings, with target test function $y = x\sin(x)$ and a small amount of observation noise $\sigma = 0.1$ ($\sigma$ as defined in Eq. 10). We use 5 training points (dark green dots).

We map our 1D inputs $x$ onto the circle $(\cos(x), \sin(x))$ before performing GP regression. This is so that all inputs have unit norm and we can use the NNGP correlation kernel (Eq. 3) for the vanilla ResNet (ResNet with fully connected blocks), in order to avoid the exploding variance problem.

As expected from our theory, in Figure 3, for depth 1000 the NNGP correlation kernel without stable scaling (top row, red) is unable to learn anything beyond a constant function due to inexpressivity, whereas our

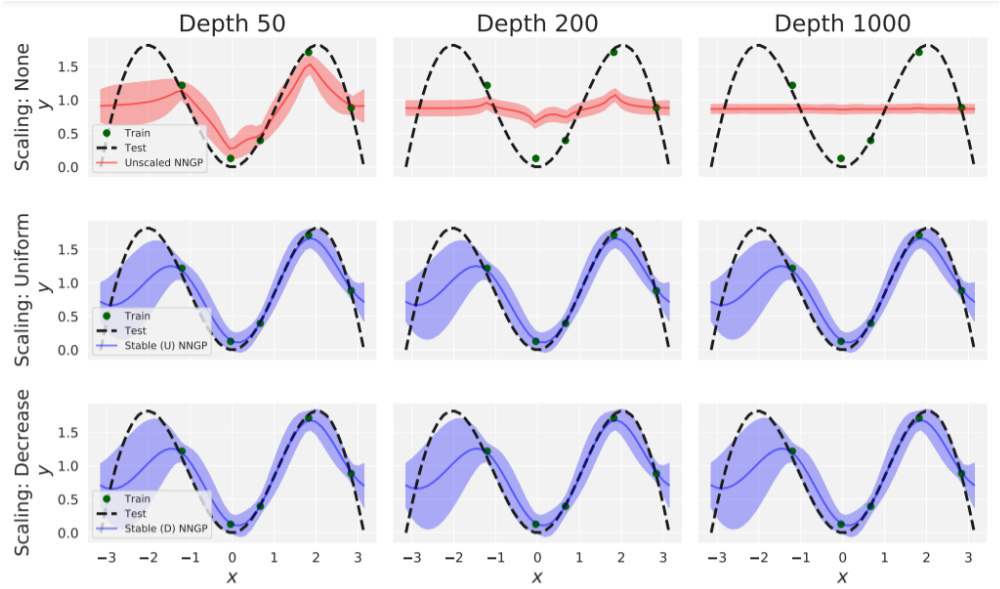



**Figure 3:** NNGP toy regression experiment.

Stable ResNets (bottom two rows, blue) are still expressive in the large depth limit. We plot mean and 95% posterior predictive credible interval for NNGP posteriors.

**Stable NNGP classification results** We first compare the performance of Stable and standard ResNets of varying depths through their infinite-width NNGP kernels, on MNIST & CIFAR-10. For each considered NNGP kernel $Q$ and training set $(x_i, y_i)_{i\in[1:N]}$, we report test accuracy using the mean of the posterior predictive (Eq. 10): $Q_N(\cdot)(Q_{NN} + \sigma^2 I)^{-1} y_N$, which is also the kernel ridge regression predictor [Kanagawa et al., 2018]. We treat classification labels $y$ as one-hot regression targets, similar to recent works [Arora et al., 2019, Lee et al., 2019, Shankar et al., 2020], and tune the noise $\sigma^2$ using prediction accuracy on a held-out validation set.

**Table 1:** CIFAR-10 test accuracies (%) using posterior predictive mean of NNGP kernels for deep Wide-ResNets [Zagoruyko and Komodakis, 2016] with different training set sizes $N$. Scaled (D) & Scaled (U) refer to decreasing and uniform scaling respectively.

| $N$ | Depth | Scaled (D) | Scaled (U) | Unscaled |
|---|---|---|---|---|
| 1K | 112 | $36.84_{\pm 0.53}$ | $36.43_{\pm 0.49}$ | $37.71_{\pm 0.50}$ |
| | 202 | $36.89_{\pm 0.55}$ | $36.47_{\pm 0.49}$ | — |
| 10K | 112 | $53.81_{\pm 0.11}$ | $53.55_{\pm 0.41}$ | $53.34_{\pm 0.07}$ |
| | 202 | $53.80_{\pm 0.10}$ | $53.57_{\pm 0.40}$ | — |

First, in Table 1, we demonstrate the exploding NNGP variance problem for unscaled Wide-ResNets (WRN) [Zagoruyko and Komodakis, 2016]. For an unscaled WRN of depth 202, the NNGP kernel values explode resulting in numerical errors, whereas Stable ResNets achieve 54% test accuracy with 10K training points (out of full size 50K). Note that any numerical errors from exploding NNGP also afflict the NTK, as the difference between the NTK and NNGP is positive semi-definite [Lee et al., 2019, He et al., 2020] (which is why the NTK lines always lie above their corresponding NNGP in Figure 1).

To isolate the disadvantages of inexpressivity in unscaled Resnets NNGPs compared to our Stable ResNets, we need to avoid the exploding variance problem and ensuing numerical errors. In order to do so, we use the NNGP correlation kernel $C$ instead of the NNGP covariance kernel $Q$, noting that these two kernels are equal up to multiplicative constant on the sphere, and that the posterior predictive mean is invariant to the scale of $Q$ (with $\sigma^2$ also tuned relative to the scale of $Q$). Moreover, the formula in Lemma A4 for NNGP correlation recursion for vanilla ResNets without bias can be recast as a ResNet with a modified scaling (see Appendix A6), allowing us to use existing optimised libraries [Novak et al., 2020]. In order to use the vanilla ResNet correlation recursion, we standardise all MNIST & CIFAR-10 images to lie on the 784 & 3072-dimension sphere respectively.

Our expressivity results, as well as Proposition 10, suggest that we expect Stable ResNets to outperform standard ResNets for large depths even when exploding variance numerical errors are alleviated for standard ResNets. In Table 2, we see that unscaled ResNets suffer from a degradation in test accuracy with depth, due to inexpressivity, whereas our Stable ResNets (both de-

**Table 2:** MNIST and CIFAR-10 test accuracies (%) using posterior predictive mean of NNGP kernels for deep vanilla ResNets (ResNet with fully connected blocks) with different size training sets $N$.

| | Dataset | MNIST | | | CIFAR-10 | | |
|---|---|---|---|---|---|---|---|
| $N$ | Depth | Scaled (D) | Scaled (U) | Unscaled | Scaled (D) | Scaled (U) | Unscaled |
| 1K | 50 | $92.88_{\pm 0.35}$ | $92.39_{\pm 0.33}$ | $92.44_{\pm 0.21}$ | $35.83_{\pm 0.14}$ | $34.73_{\pm 0.14}$ | $\mathbf{37.16}_{\pm 0.25}$ |
| | 200 | $92.91_{\pm 0.35}$ | $92.39_{\pm 0.32}$ | $89.56_{\pm 0.56}$ | $\mathbf{35.86}_{\pm 0.14}$ | $34.76_{\pm 0.11}$ | $34.85_{\pm 0.17}$ |
| | 1000 | $92.92_{\pm 0.34}$ | $92.39_{\pm 0.32}$ | $55.13_{\pm 5.31}$ | $\mathbf{35.89}_{\pm 0.14}$ | $34.76_{\pm 0.11}$ | $12.43_{\pm 3.97}$ |
| 10K | 50 | $97.57_{\pm 0.12}$ | $97.55_{\pm 0.12}$ | $97.06_{\pm 0.10}$ | $48.71_{\pm 0.31}$ | $48.12_{\pm 0.27}$ | $\mathbf{50.11}_{\pm 0.37}$ |
| | 200 | $97.57_{\pm 0.11}$ | $97.55_{\pm 0.12}$ | $95.55_{\pm 0.13}$ | $\mathbf{48.77}_{\pm 0.30}$ | $47.15_{\pm 0.18}$ | $47.00_{\pm 0.30}$ |
| | 1000 | $97.57_{\pm 0.10}$ | $97.54_{\pm 0.12}$ | $67.53_{\pm 2.96}$ | $\mathbf{48.76}_{\pm 0.30}$ | $47.16_{\pm 0.17}$ | $17.86_{\pm 2.32}$ |

creasing and uniform) do not suffer from a drop in performance. For example, the posterior predictive mean using the NNGP of an unscaled vanilla ResNet with depth 1000 attains only 17.86% accuracy on CIFAR-10 with 10K training points, compared to 48.76% for Stable ResNet (decreasing scale).

We focus on the NNGP rather than the NTK as recent works [Lee et al., 2020, Shankar et al., 2020] have empirically demonstrated that there is no advantage to the state-of-the-art NTK over the NNGP as infinite-width kernel predictors. Moreover, we do not aim for near state-of-the-art kernel results due to computational resources, and instead aim to empirically validate the theoretical advantages of Stable ResNets.

**Trained Stable ResNet results** Finally, we consider the benefits of trained Stable ResNets on the large-scale CIFAR-10, CIFAR-100 and TinyImageNet[13] datasets. We compare trained convolutional ResNets [He et al., 2016] of depths 32, 50 & 104 in terms of test accuracy. In the main text we present results for ResNets trained with Batch Normalization [Ioffe and Szegedy, 2015] (BatchNorm), while results for trained ResNets without BatchNorm can be found in Appendix A7. We particularly show that Stable ResNet can be trained without BatchNorm with minimal loss in performance. Avoiding BatchNorm is a desirable property because of its expensive memory cost during training. For the results with BatchNorm, we apply the Stable ResNet scalings to the residual connection after all convolution, ReLU and BatchNorm layers.

We use initial learning rate 0.1 which is decayed by 0.1 at 50% and 75% of the way through training. This learning rate schedule has been used previously [He et al., 2016] for unscaled ResNets and we found it to work well for all ResNets trained with BatchNorm. We train for 160 epochs on CIFAR-10/100 and 250 epochs on TinyImageNet. Test accuracy results are displayed in Table 3. As we can see, Stable ResNets consistently outperform standard ResNets across datasets and depths. Moreover, the performance gap is larger for larger depths: for example on CIFAR-100 our Stable ResNet (decreasing) outperforms its standard counterpart by 1.05% (75.06 vs 74.01) on average for depth 32 whereas for depth 104 the test accuracy gap is 2.36% (77.44 vs 75.08) on average. A similar trend can also be observed for the more challenging TinyImageNet dataset. Interestingly, we see that among the Stable ResNets, decreasing scaling also consistently outperforms uniform scaling.

**Table 3:** Test accurracies (%) of trained deep ResNets of various scalings and depths on CIFAR-10 (C-10), CIFAR-100 (C-100) & TinyImageNet (Tiny-I).

| Dataset | Depth | Scaled (D) | Scaled (U) | Unscaled |
|---|---|---|---|---|
| C-10 | 32 | $94.84_{\pm 0.08}$ | $94.78_{\pm 0.17}$ | $94.66_{\pm 0.07}$ |
| | 50 | $\mathbf{95.07}_{\pm 0.06}$ | $94.99_{\pm 0.03}$ | $94.85_{\pm 0.06}$ |
| | 104 | $95.14_{\pm 0.19}$ | $\mathbf{95.31}_{\pm 0.07}$ | $95.10_{\pm 0.21}$ |
| C-100 | 32 | $\mathbf{75.06}_{\pm 0.05}$ | $74.79_{\pm 0.28}$ | $74.01_{\pm 0.14}$ |
| | 50 | $\mathbf{76.20}_{\pm 0.22}$ | $75.81_{\pm 0.20}$ | $74.66_{\pm 0.33}$ |
| | 104 | $\mathbf{77.44}_{\pm 0.09}$ | $76.88_{\pm 0.39}$ | $75.08_{\pm 0.42}$ |
| Tiny-I | 32 | $63.01_{\pm 0.22}$ | $\mathbf{63.06}_{\pm 0.04}$ | $62.79_{\pm 0.08}$ |
| | 50 | $64.78_{\pm 0.24}$ | $64.74_{\pm 0.10}$ | $63.96_{\pm 0.39}$ |
| | 104 | $66.57_{\pm 0.39}$ | $66.67_{\pm 0.12}$ | $65.27_{\pm 0.52}$ |

## 8 CONCLUSION

Stable ResNets have the benefit of stabilizing the gradient and ensuring expressivity in the limit of infinite depth. We have demonstrated theoretically and empirically that this type of scaling makes NNGP inference robust and improves test accuracy with SGD on modern ResNet architectures. However, while Stable ResNets with both uniform and decreasing scalings outperform standard ResNet, the selection of an optimal scaling remains an open question; we leave this topic for future work.

[13] Available at http://cs231n.stanford.edu/tiny-imagenet-200.zip

## ACKNOWLEDGMENTS

This material is based upon work supported in part by the U.S. Army Research Laboratory and the U. S. Army Research Office, and by the U.K. Ministry of Defence (MoD) and the U.K. Engineering and Physical Research Council (EPSRC) under grant number EP/R013616/1. AD is also partially supported by EPSRC EP/R034710/1. BH is supported by the EPSRC and MRC through the OxWaSP CDT programme (EP/L016710/1). 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

G. Yang and S. Schoenholz. Mean field residual networks: On the edge of chaos. In *Advances in Neural Information Processing Systems*, pages 7103–7114, 2017.

S. Hayou, A. Doucet, and J. Rousseau. On the impact of the activation function on deep neural networks training. In *International Conference on Machine Learning*, 2019a.

R.M. Neal. *Bayesian Learning for Neural Networks*, volume 118. Springer Science & Business Media, 1995.

B. Poole, S. Lahiri, M. Raghu, J. Sohl-Dickstein, and S. Ganguli. Exponential expressivity in deep neural networks through transient chaos. In *Advances in Neural Information Processing Systems*, 2016.

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

J. Lee, L. Xiao, S. Schoenholz, Y. Bahri, R. Novak, J. Sohl-Dickstein, and J. Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in Neural Information Processing Systems*. 2019.

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

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

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

S. Hayou, A. Doucet, and J. Rousseau. Mean-field behaviour of neural tangent kernel for deep neural networks. *arXiv preprint arXiv:1905.13654*, 2019b.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International Conference on Machine Learning*, 2015.

S. Hayou, J.F. Ton, A. Doucet, and Y.W. Teh. Robust pruning at initialization. In *International Conference on Learning Representations*, 2021.

G. Yang. Tensor programs i: Wide feedforward or recurrent neural networks of any architecture are Gaussian processes. *arXiv preprint arXiv:1910.12478*, 2019b.

A. Daniely, R. Frostig, and Y. Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. *Advances in Neural Information Processing Systems 29*, 2016.

V.I. Paulsen and M. Raghupathi. *An Introduction to the Theory of Reproducing Kernel Hilbert Spaces*. Cambridge University Press, 2016.

R. M. Dudley. *Real Analysis and Probability*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2 edition, 2002.

I. Steinwart. Convergence types and rates in generic Karhunen-Loeve expansions with applications to sample path properties. *Potential Analysis*, 51(3): 361–395, 2019.

S. Lang. *Real and Functional Analysis*. Graduate Texts in Mathematics. Springer, New York, 3rd edition, 2012.

U. Grenander. Stochastic processes and statistical inference. *Arkiv Matematik*, 1(3):195–277, 10 1950.

G. Yang and H. Salman. A fine-grained spectral perspective on neural networks. *arXiv preprint 1907.10599*, 2019.

B. Sriperumbudur, K. Fukumizu, and G. Lanckriet. Universality, characteristic kernels and RKHS embedding of measures. *Journal of Machine Learning Research*, 12(70):2389–2410, 2011.

M. Kanagawa, P. Hennig, D. Sejdinovic, and B.K. Sriperumbudur. Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv preprint arXiv:1807.02582*, 2018.

L. Chizat and F. Bach. A note on lazy training in supervised differentiable programming. In *Advances in Neural Information Processing Systems*, 2019.

A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, 2018.

M. Seeger. PAC-Bayesian generalisation error bounds for Gaussian process classification. *Journal of Machine Learning Research*, page 233–269, 02 2002.

S. Arora, S.S. Du, W. Hu, Z. Li, R. Salakhutdinov, and R. Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, 2019.

V. Shankar, A. Fang, W. Guo, S. Fridovich-Keil, L. Schmidt, J. Ragan-Kelley, and B. Recht. Neural kernels without tangents. *International Conference on Machine Learning*, 2020.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*, 2016.

B. He, B. Lakshminarayanan, and Y. W. Teh. Bayesian deep ensembles via the neural tangent kernel. *Advances in Neural Information Processing Systems*, 2020.

R. Novak, L. Xiao, J. Hron, J. Lee, A. Alemi, J. Sohl-Dickstein, and S. Schoenholz. Neural tangents: Fast and easy infinite neural networks in Python. In *International Conference on Learning Representations*, 2020. URL https://github.com/google/neural-tangents.

J. Lee, S. Schoenholz, J. Pennington, B. Adlam, L. Xiao, R. Novak, and J. Sohl-Dickstein. Finite versus infinite neural networks: an empirical study. *Advances in Neural Information Processing Systems*, 2020.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

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

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, pages 337–404, 1950.

C. Micchelli, Y. Xu, and H. Zhang. Universal kernels. *Journal of Machine Learning Research*, pages 2651–2667, 2006.

O. Kounchev. *Multivariate Polysplines: Applications to Numerical and Wavelet Analysis*. Elsevier Science, 2001.

J. Bradbury, R. Frostig, P. Hawkins, M. Johnson, C. Leary, D. Maclaurin, and S. Wanderman-Milne. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.

R. Novak, L. Xiao, J. Lee, Y. Bahri, G. Yang, J. Hron, D. A Abolafia, J. Pennington, and J. Sohl-Dickstein. Bayesian deep convolutional networks with many channels are Gaussian processes. In *International Conference on Learning Representations*, 2019.

C. Wang, G. Zhang, and R. Grosse. Picking winning tickets before training by preserving gradient flow. In *International Conference on Learning Representations*, 2020.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8026–8037, 2019.

S De and SL Smith. Batch normalization biases residual blocks towards the identity function in deep networks. *Advances in Neural Information Processing Systems*, 2020.

H. Zhang, Y. N Dauphin, and T. Ma. Fixup initialization: Residual learning without normalization. In *International Conference on Learning Representations*, 2019.

T.M. MacRobert. *Spherical Harmonics: An Elementary Treatise on Harmonic Functions with Applications*. Pergamon Press, 1967.

# Appendix

## A0 Mathematical preliminaries

We will make use of functional analysis results on the theory of Hilbert space. We refer to [Lang, 2012] for a comprehensive introduction to the topic. We precise here that, even when not explicitly stated, all Hilbert spaces considered in the present work are real, and all linear operators are bounded.
We will make use of the spectral theory for compact self-adjoint operators. We refer again to [Lang, 2012] for a detailed discussion.

We will now introduce some concepts from the theory of kernels and RKHSs.
Consider a compact $K \subset \mathbb{R}^d$. A function $Q : K \to \mathbb{R}$ is said to be symmetric if for all $x, x' \in K$ we have $Q(x, x') = Q(x', x)$. Let us restate the definition of kernel.

**Definition 2** (Kernel)**.** *A kernel $Q$ on $K$ is a symmetric continuous function $K^2 \to \mathbb{R}$ such that, for all $n \in \mathbb{N}$, for any finite subset $\{x_1 \dots x_n\} \subset K$, the matrix $\{Q(x_i, x_j)\}_{i,j}$ is non-negative definite.*

We state here a characterisation of kernels, which is an extension of Lemma 2. Despite being a classical result (see the discussion about Mercer kernels in [Paulsen and Raghupathi, 2016]), we will give a proof, for the sake of completeness.

**Lemma A1.** *[Extension of Lemma 2] Let $Q : K^2 \to \mathbb{R}$ be a continuous symmetric function. Then, given any finite Borel measure $\mu$ on $K$, we can define the integral operator $T_\mu(Q)$ on $L^2(K, \mu)$, via*

$$T_\mu(Q)\,\varphi(x) = \int_K T(x, x')\,\varphi(x')\,\mathrm{d}\mu(x')\,,$$

*for any $\varphi \in L^2(K, \mu)$. The operator $T_\mu(Q)$ is a bounded compact self-adjoint definite operator.*
*Moreover, $Q$ is a kernel if and only if $T_\mu(Q)$ is non-negative definite for all finite Borel measures $\mu$ on $K$.*

*Proof.* Let $Q : K^2 \to \mathbb{R}$ be a continuous symmetric function. Then $T_\mu(Q)$ is a well defined bounded compact self-adjoint operator [Lang, 2012].
Let us assume that $Q$ is a kernel. By Mercer's theorem [Paulsen and Raghupathi, 2016], we can find continuous functions $\{Y_k\}_{k\in\mathbb{N}}$ such that for all $x, x' \in K$

$$Q(x, x') = \sum_{k=0}^{\infty} Y_k(x) Y_k(x')$$

and the convergence is uniform on $K^2$.
The continuity of the $Y_k$'s implies that they can be seen as elements of $L^2(K, \mu)$. Moreover, the uniform convergence, along with the fact that $\mu(K) < \infty$, implies the convergence of the sum wrt the $L^2(K, \mu)$ operator norm. In particular $T_\mu(K)$ is a limit of non-negative definite operators and hence non-negative definite.
Now, assume that, for all finite Borel $\mu$, $T_\mu(Q)$ is non-negative definite. Chosen a finite set $\{x_1 \dots x_n\} \subset K$, in particular we have that $\mu = \sum_{i=1}^n \delta_{x_i}$ is a finite Borel measure (where $\delta_x$ is the Dirac measure on $x \in K$). Hence $T_\mu(Q)$ is the matrix $\{Q(x_i, x_j)\}_{i,j}$. We conclude that $Q$ is a kernel. □

We will now give a definition of the Reproducing Kernel Hilbert Space associated to a kernel. We refer to [Paulsen and Raghupathi, 2016] for a general and comprehensive introduction to the topic.

**Definition A1** (RKHS)**.** *Given a kernel $Q$ on $K$, we can associate to it a real Hilbert space $\mathcal{H}_Q$, with the following properties:*

- *The elements of $\mathcal{H}_Q$ are functions $K \to \mathbb{R}$.*
- *Denoting as $\langle \cdot, \cdot \rangle_Q$ the inner product of $\mathcal{H}_Q$, for each $x \in K$, there exists an element $k_x \in \mathcal{H}_Q$ such that $h(x) = \langle h, k_x \rangle_Q$, for all $h \in \mathcal{H}_Q$.*
- *For all $x, x' \in K$, $\langle k_x, k_{x'} \rangle_Q = Q(x, x')$.*

*Such a Hilbert space exists for each kernel $Q$ and it is unique up to isomorphism, [Paulsen and Raghupathi, 2016]. $\mathcal{H}_Q$ is called the Reproducing Kernel Hilbert Space (RKHS) of $Q$.*

In general, it is not easy to give an explicit form for the RKHS associated to a kernel $Q$. However, we can say that it contains the linear span of $\{x \mapsto Q(x, x')\}_{x' \in K}$. Actually, this linear span is a dense subset of $\mathcal{H}_Q$, wrt the norm of $\mathcal{H}_Q$ [Paulsen and Raghupathi, 2016].

A kernel on $K$ is said to be universal if its RKHS is dense in the space of continuous functions $C(K)$, wrt the uniform norm.

**Definition 4** (Universal Kernel)**.** *Let $Q$ be a kernel on $K$, and $\mathcal{H}_Q(K)$ its RKHS. We say that $Q$ is universal on $K$ if for any $\varepsilon > 0$ and any continuous function $g$ on $K$, there exists $h \in \mathcal{H}_Q(K)$ such that $\|h - g\|_\infty < \varepsilon$.*

We can now state a characterization of universal kernels, from [Sriperumbudur et al., 2011].

**Lemma A2.** *Let $Q : K^2 \to \mathbb{R}$ be a kernel, where $K \subset \mathbb{R}^d$ is compact. $Q$ is a universal kernel if and only if $T_\mu(Q)$ is strictly positive definite for all finite Borel measures $\mu$ on $K$, i.e., $\langle T_\mu(Q)\,\varphi, \varphi\rangle > 0$ for all non-zero $\varphi \in L^2(K, \mu)$.*

As a final note, hereafter we often omit the explicit reference to the measure $\mu$, that is we will speak of the operator $T(Q)$ on $L^2(K)$. Unless otherwise stated, this notation implies the choice of an arbitrary finite Borel measure $\mu$ on the compact $K$.

## A1 Residual Neural Networks and Gaussian processes

Consider a standard ResNet architecture with $L+1$ layers, labelled with $l \in [0 : L]$, of dimensions $\{N_l\}_{l \in [0:L]}$.

$$\begin{aligned} &y_0(x) = W_0\, x + B_0\,; \\ &y_l(x) = y_{l-1}(x) + \mathcal{F}((W_l, B_l), y_{l-1}) \quad \text{for } l \in [1 : L]\,, \end{aligned} \tag{1}$$

where $x \in \mathbb{R}^d$ is an input, $y_l(x)$ is the vector of pre-activations, $W_l$ and $B_l$ are respectively the weights and bias of the $l^{th}$ layer, and $\mathcal{F}$ is a mapping that defines the nature of the layer. In general, the mapping $\mathcal{F}$ consists of successive applications of simple activation functions. In this work, for the sake of simplicity, we consider Fully Connected blocks with ReLU activation function $\phi : x \mapsto \max(0, x)$

$$\mathcal{F}((W, B), x) = W\phi(x) + B\,.$$

Hereafter, $N_l$ denotes the number of neurons in the $l^{th}$ layer, $\phi$ the activation function and $[m : n] := \{m, m+1, ..., n\}$ for $m \leq n$. The components of weights and bias are respectively initialized with $W_l^{ij} \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_w^2/N_{l-1})$, and $B_l^i \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_b^2)$ where $\mathcal{N}(\mu, \sigma^2)$ denotes the normal distribution of mean $\mu$ and variance $\sigma^2$.

In [Yang and Schoenholz, 2017], authors showed that wide deep ResNets might suffer from gradient exploding during backpropagation.

Recent results by [Hayou et al., 2021] suggest that scaling the residual blocks with $L^{-1/2}$ might have some beneficial properties on model pruning at initialization. This is a result of the stabilization effect of scaling on the gradient.

More generally, we introduce the residual architecture:

$$\begin{aligned} &y_0(x) = W_0\, x + B_0\,; \\ &y_l(x) = y_{l-1}(x) + \lambda_{l,L}\, \mathcal{F}((W_l, B_l), y_{l-1})\,, \quad l \in [1 : L]\,, \end{aligned} \tag{2}$$

where $(\lambda_{k,L})_{k \in [1:L]}$ is a sequence of scaling factors. We assume hereafter that there exists $\lambda_{\max} \in (0, \infty)$ such that for all $L \geq 1$ and $k \in [1 : L]$, we have that $\lambda_{k,L} \in (0, \lambda_{\max}]$.

### A1.1 Recurrence for the covariance kernel

Recall that in the limit of infinite width, each layer of a ResNet can be seen a centred Gaussian Process. For the layer $l$ we define the covariance kernel $Q_l$ as $Q_l(x, x') = \mathbb{E}[y_l^1(x) y_l^1(x')]$ for $x, x' \in \mathbb{R}^d$.

By a standard approach, introduced by [Schoenholz et al., 2017] for feedforward neural networks, and easily generalizable for ResNets [Yang, 2019b, Hayou et al., 2019b], it is possible to evaluate the covariance kernels layer by layer, recursively. More precisely, consider a ResNet of form (2). Assume that $y^i_{l-1}$ is a Gaussian process for all $i$. Let $x, x' \in \mathbb{R}^d$. We have that

$$\begin{aligned} Q_l(x,x') &= \mathbb{E}[y^1_l(x)y^1_l(x')] \\ &= \mathbb{E}[y^1_{l-1}(x)y^1_{l-1}(x')] + \sum_{j=1}^{N_{l-1}} \mathbb{E}[(W^{1j}_l)^2\phi(y^j_{l-1}(x))\phi(y^j_{l-1}(x'))] + \mathbb{E}[(B^1_l)^2] + \mathbb{E}[B^1_l(y^1_{l-1}(x)+y^1_{l-1}(x'))] \\ &+ \mathbb{E}\left[\sum_{j=1}^{N_{l-1}} W^{1j}_l(y^1_{l-1}(x)\phi(y^1_{l-1}(x')) + y^1_{l-1}(x')\phi(y^1_{l-1}(x)))\right]. \end{aligned}$$

Some terms vanish because $\mathbb{E}[W^{1j}_l] = \mathbb{E}[B^j_l] = 0$. Let $Z_j = \frac{\sqrt{N_{l-1}}}{\sigma_w} W^{1j}_l$. The second term can be written as

$$\mathbb{E}\left[\frac{\sigma_w^2}{N_{l-1}}\sum_j (Z_j)^2\phi(y^j_{l-1}(x))\phi(y^j_{l-1}(x'))\right] \to \sigma_w^2\,\mathbb{E}[\phi(y^1_{l-1}(x))\phi(y^1_{l-1}(x'))]\,,$$

where we have used the Central Limit Theorem. Therefore, we have

$$Q_l(x,x') = Q_{l-1}(x,x') + \lambda^2_{l,L}\Psi_{l-1}(x,x')\,, \tag{A1}$$

where $\Psi_{l-1}(x,x') = \sigma_b^2 + \sigma_w^2\mathbb{E}[\phi(y^1_{l-1}(x))\phi(y^1_{l-1}(x'))]$.

For the ReLU activation function $\phi(x) = \max(0,x)$, the recurrence relation can be written more explicitly, since we can give a simple expression for the expectation $\mathbb{E}[\phi(y^1_{l-1}(x))\phi(y^1_{l-1}(x'))]$, [Daniely et al., 2016]. Let $C_l$ be the correlation kernel, defined as

$$C_l(x,x') = \frac{Q_l(x,x')}{\sqrt{Q_l(x,x)Q_l(x,x')}}$$

and let $f : [-1,1] \to \mathbb{R}$ be given by

$$f : \gamma \mapsto \frac{1}{\pi}(\sqrt{1-\gamma^2} - \gamma\arccos\gamma)\,. \tag{4}$$

Then we have $\mathbb{E}[\phi(y^1_{l-1}(x))\phi(y^1_{l-1}(x'))] = \frac{1}{2}\left(1 + \frac{f(C_{l-1})}{C_{l-1}}\right)Q_{l-1}$ and so we find the recurrence relation (5)

$$\begin{aligned} &Q_l = Q_{l-1} + \lambda^2_{l,L}\left[{\sigma_b}^2 + \tfrac{{\sigma_w}^2}{2}\left(1 + \tfrac{f(C_{l-1})}{C_{l-1}}\right)Q_{l-1}\right]; \\ &Q_0(x,x') = \sigma_b^2 + \sigma_w^2\,\tfrac{x\cdot x'}{d}\,. \end{aligned} \tag{5}$$

For the remainder of this appendix, we define the function

$$\hat{f}(\gamma) = \gamma + f(\gamma) = \frac{1}{\pi}\left(\gamma\arcsin(\gamma) + \sqrt{1-\gamma^2}\right) + \frac{1}{2}\gamma\,. \tag{A2}$$

For all $l$, the diagonal terms of $Q_l$ have closed-form expressions. We show this in the next lemma.

**Lemma A3** (Diagonal elements of the covariance)**.** *Consider a ResNet of the form (2) and let $x \in \mathbb{R}^d$. We have that for all $l \in [1:L]$,*

$$Q_l(x,x) = -\frac{2\sigma_b^2}{\sigma_w^2} + \prod_{k=1}^{l}\left(1 + \frac{\sigma_w^2\lambda^2_{k,L}}{2}\right)\left(Q_0(x,x) + \frac{2\sigma_b^2}{\sigma_w^2}\right).$$

*Proof.* We know that

$$Q_l(x,x) = Q_{l-1}(x,x) + \lambda^2_{l,L}\left(\sigma_b^2 + \frac{\sigma_w^2}{2}\hat{f}(1)\right),$$

where $\hat{f}$ is given by (A2). It is straightforward that $\hat{f}(1) = 1$. This yields

$$Q_l(x,x) + \frac{2\sigma_b^2}{\sigma_w^2} = \left(1 + \lambda_{l,L}^2 \frac{\sigma_w^2}{2}\right)\left(Q_{l-1}(x,x) + \frac{2\sigma_b^2}{\sigma_w^2}\right).$$

we conclude by telescopic product. □

As a corollary of the previous result, it is easy to show that for a Standard ResNet the diagonal terms explode with depth, which is Lemma 1 in the main paper.

**Lemma 1** (Exploding kernel with standard ResNet)**.** *Consider a ResNet of type (1). Then, for all $x \in \mathbb{R}^d$,*

$$Q_L(x,x) \geq \left(1 + \tfrac{\sigma_w^2}{2}\right)^L \left(\sigma_b^2\left(1 + \tfrac{2}{\sigma_w^2}\right) + \tfrac{\sigma_w^2}{d}\|x\|^2\right).$$

*Proof.* The statement trivially follows from Lemma A3, using that $Q_0(x,x) = \sigma_b^2 + \frac{\sigma_w^2}{d}\|x\|^2$ and the fact that for a Standard ResNet (1), all the coefficients $\lambda_{l,L}$'s are equal to 1. □

In the case of a ResNet with no bias, the correlation kernel follows a simple recursive formula described in the next lemma.

**Lemma A4** (Correlation formula with zero bias)**.** *For a ResNet of the form (2) with $\sigma_b = 0$, we have that for all $x, x' \in \mathbb{R}^d$ and $l \leq L$:*

$$C_l(x,x') = \frac{1}{1+\alpha_{l,L}} C_{l-1}(x,x') + \frac{\alpha_{l,L}}{1+\alpha_{l,L}} \hat{f}(C_{l-1}(x,x')),$$

*where $\alpha_{l,L} = \frac{\lambda_{l,L}^2 \sigma_w^2}{2}$.*

*Proof.* This is direct result of the covariance recursion formula (5). □

### A1.2 Proof of Proposition 1

We use the following result from [Yang, 2020] in order to derive closed form expressions for the second moment of the gradients.

**Lemma A5** (Corollary of Theorem D.1. in [Yang, 2020])**.** *Consider a ResNet of the form (2) with weights $W$. In the limit of infinite width, we can assume that $W^T$ used in back-propagation is independent from $W$ used for forward propagation, for the calculation of Gradient Covariance and NTK.*

Next we re-state and prove Proposition 1.

**Proposition 1** (Stable Gradient)**.** *Consider a ResNet of type (2), and let $\mathcal{L}_y(x) := \ell(y_L^1(x), y)$ for some $(x,y) \in \mathbb{R}^d \times \mathbb{R}$, where $\ell : (z,y) \mapsto \ell(z,y)$ is a loss function satisfying $\sup_{K_1 \times K_2} \left|\frac{\partial \ell(z,y)}{\partial z}\right| < \infty$, for all compacts $K_1, K_2 \subset \mathbb{R}$. Then, in the limit of infinite width, for any compacts $K \subset \mathbb{R}^d$, $K' \subset \mathbb{R}$, there exists a constant $C > 0$ such that for all $(x,y) \in K \times K'$*

$$\sup_{l \in [0:L]} \mathbb{E}\left[\left|\frac{\partial \mathcal{L}_y(x)}{\partial W_l^{11}}\right|^2\right] \leq C \exp\left(\tfrac{\sigma_w^2}{2}\textstyle\sum_{l=1}^L \lambda_{l,L}^2\right).$$

*Moreover, if there exists $\lambda_{\min} > 0$ such that for all $L \geq 1$ and $l \in [1:L]$ we have $\lambda_{l,L} \geq \lambda_{\min}$, then, for all $(x,y) \in (\mathbb{R}^d \setminus \{0\}) \times \mathbb{R}$ such that $\left|\frac{\partial \ell(z,y)}{\partial z}\right| \neq 0$, there exists $\kappa > 0$ such that for all $l \in [1:L]$*

$$\mathbb{E}\left[\left|\frac{\partial \mathcal{L}_y(x)}{\partial W_l^{11}}\right|^2\right] \geq \kappa\left(1 + \tfrac{\lambda_{\min}^2 \sigma_w^2}{2}\right)^L.$$

*Proof.* Let $(x,y) \in \mathbb{R}^d \times \mathbb{R}$ and $\bar{q}^l(x,y) = \mathbb{E}\left[\left|\frac{\partial \mathcal{L}_y(x)}{\partial y_l^1}\right|^2\right]$. Using Lemma A6, we have that

$$\bar{q}^l(x,y) = \left(1 + \frac{\sigma_w^2 \lambda_{l+1,L}^2}{2}\right) \bar{q}^{l+1}(x,y)\,.$$

This yields

$$\bar{q}^l(x,y) = \prod_{k=l+1}^{L} \left(1 + \frac{\sigma_w^2 \lambda_{k,L}^2}{2}\right) \bar{q}^l(x,y)\,.$$

Moreover, using Lemma A5, we have that $\mathbb{E}\left[\left|\frac{\partial \mathcal{L}_y(x)}{\partial W_l^{11}}\right|^2\right] = \lambda_{l,L}^2 \bar{q}^l(x,y)\mathbb{E}[\phi(y_{l-1}^1(x))^2]$. We have $\mathbb{E}[\phi(y_{l-1}^1(x))^2] = \frac{1}{2}Q_{l-1}(x,x)$. From Lemma A3 we know that

$$Q_{l-1}(x,x) = -\frac{2\sigma_b^2}{\sigma_w^2} + \prod_{k=1}^{l-1}\left(1 + \frac{\sigma_w^2\lambda_{k,L}^2}{2}\right)\left(Q_0(x,x) + \frac{2\sigma_b^2}{\sigma_w^2}\right) \leq \prod_{k=1}^{l-1}\left(1 + \frac{\sigma_w^2\lambda_{k,L}^2}{2}\right)\left(Q_0(x,x) + \frac{2\sigma_b^2}{\sigma_w^2}\right)\,,$$

This yields

$$\mathbb{E}\left[\left|\frac{\partial \mathcal{L}_y(x)}{\partial W_{11}^l}\right|^2\right] \leq \frac{2}{\sigma_w^2}\prod_{k=1}^{L}\left(1 + \frac{\sigma_w^2\lambda_{k,L}^2}{2}\right)\left(\frac{1}{2}Q_0(x,x) + \frac{\sigma_b^2}{\sigma_w^2}\right)\bar{q}^l(x,y)\,.$$

It is straightforward that $\prod_{k=1}^{L}\left(1 + \frac{\sigma_w^2\lambda_{k,L}^2}{2}\right) \leq \exp\left(\frac{\sigma_w^2}{2}\sum_{k=1}^{L}\lambda_{k,L}^2\right)$. Let $K \subset \mathbb{R}^d$, $K' \subset \mathbb{R}$ be two compact subsets. Using the condition on the loss function $\ell$, we have that

$$\mathbb{E}\left[\left|\frac{\partial \mathcal{L}_y(x)}{\partial W_{11}^l}\right|^2\right] \leq C\exp\left(\frac{\sigma_w^2}{2}\sum_{k=1}^{L}\lambda_{k,L}^2\right),$$

where $C = \frac{2}{\sigma_w^2}\left(\sup_{(x,y)\in K\times K'}\bar{q}^l(x,y)\right)\left(\sup_{x\in K}Q_0(x,x) + \frac{2\sigma_b^2}{\sigma_w^2}\right)$. We conclude by taking the supremum over $l$ and $x, y$.

Let $(x,y) \in (\mathbb{R}^d \setminus \{0\}) \times \mathbb{R}$ such that $\left|\frac{\partial \ell(z,y)}{\partial z}\right| \neq 0$. We have that

$$\begin{aligned}\mathbb{E}\left[\left|\frac{\partial \mathcal{L}_y(x)}{\partial W_{11}^l}\right|^2\right] &\geq \frac{1}{2}\frac{\lambda_{l,L}^2}{1 + \frac{\sigma_w^2}{2}\lambda_{l,L}^2}\prod_{k=2}^{L}\left(1 + \frac{\sigma_w^2\lambda_{k,L}^2}{2}\right)Q_1(x,x)\bar{q}^l(x,y) \\ &\geq \kappa\left(1 + \frac{\sigma_w^2\lambda_{\min}^2}{2}\right)^L,\end{aligned}$$

where $\kappa = \frac{1}{2}\frac{\lambda_{\min}^2}{\left(1+\frac{\sigma_w^2}{2}\lambda_{\max}^2\right)\left(1+\frac{\sigma_w^2}{2}\lambda_{\min}^2\right)}Q_1(x,x)\,\bar{q}^l(x,y) > 0.$ □

Using Lemma A5, we can derive simple recursive formulas for the second moment of the gradient as well as for the Neural Tangent Kernel (NTK). This was previously done in [Schoenholz et al., 2017] for feedforward neural networks, we prove a similar result for ResNet in the next lemma.

**Lemma A6** (Gradient Second moment)**.** *In the limit of infinite width, using the same notation as in proposition 1, we have that*

$$\bar{q}^l(x,y) = \left(1 + \frac{\sigma_w^2\lambda_{l+1,L}^2}{2}\right)\bar{q}^{l+1}(x,y)\,.$$

*Proof.* It is straighforward that

$$\frac{\partial \mathcal{L}_y(x)}{\partial y_l^i} = \frac{\partial \mathcal{L}_y(x)}{\partial y_{l+1}^i} + \lambda_{l+1,L} \sum_j \frac{\partial \mathcal{L}_y(x)}{\partial y_{l+1}^j} W_{l+1}^{ji} \phi'(y_l^i)\,.$$

Using lemma A5 and the Central Limit Theorem, we have that

$$\bar{q}^l(x,y) = \bar{q}^{l+1}(x,y) + \lambda_{l+1,L}^2 \bar{q}^{l+1}(x,y)\sigma_w^2 \mathbb{E}[\phi'(y_l^i(x))^2]\,.$$

We conclude using $\mathbb{E}[\phi'(y_i^l(x))^2] = \mathbb{P}(\mathcal{N}(0,1) > 0) = \frac{1}{2}$. □

Before moving to the next proofs, recall the definition of Stable ResNet.

**Definition 1** (Stable ResNet). *A ResNet of type (2) is called a Stable ResNet if and only if* $\lim_{L\to\infty} \sum_{k=1}^{L} \lambda_{k,L}^2 < \infty$.

### A1.3 Some general results: $Q_l$ and $C_l$ are kernels

Fix a compact $K \subset \mathbb{R}^d$. If $\sigma_b = 0$, then assume that $0 \notin K$. We will now show that, for all layers $l$, the covariance function $Q_l$ is a kernel in the sense of Definition 2.
The symmetric property of $Q_l$ is clear by definition as the covariance of a Gaussian Process. Let us now discuss the regularity of $Q_l$ as a function on $K^2$.

The next result shows that any function $F(\phi) : \gamma \mapsto \mathbb{E}[\phi(X)\phi(Y), (X,Y) \sim \mathcal{N}(0, \left(\begin{smallmatrix} 1 & \gamma \\ \gamma & 1 \end{smallmatrix}\right))]$ is analytic on the segment $[-1,1]$.

**Lemma A7** (O'Donnell (2014)). *Let* $F(\phi)(\gamma) = \mathbb{E}[\phi(X)\phi(Y), (X,Y) \sim \mathcal{N}(0, \left(\begin{smallmatrix} 1 & \gamma \\ \gamma & 1 \end{smallmatrix}\right))]$. *Then for all* $\phi \in L^2(\mathcal{N}(0,1))$, *there exists a non negative sequence* $\{a_n\}_{n\in\mathbb{N}}$ *such that* $F(\phi)(\gamma) = \sum_{i\in\mathbb{N}} a_i \gamma^i$ *for all* $\gamma \in [-1,1]$.

Leveraging the previous result, the function $f$ defined in (4) is analytic. We clarify this in the next lemma.

**Lemma A8** (Analytic property of $f$). *The function* $f : [-1,1] \to \mathbb{R}$, *defined in (4), is an analytic function on* $(-1,1)$, *whose expansion* $f(\gamma) = \sum_{n\in\mathbb{N}} \alpha_n \gamma^n$ *converges absolutely on* $[-1,1]$. *Moreover,* $\alpha_n > 0$ *for all even* $n \in \mathbb{N}$, $\alpha_1 = -1/2$ *and* $\alpha_n = 0$ *for all odd* $n \geq 3$.

*Proof.* With the notations of Lemma A7, when $\phi$ is the ReLU activation function we have that $F(\phi) = 1/2\hat{f}$, defined in (A2). Hence, by Lemma A7, we know that $\hat{f}$ is analytic on $(-1,1)$ and its expansion around 0 converges on $[-1,1]$. In particular this will be true for $f$ as well.
For $\gamma \in [-1,1]$, let us write $\hat{f}(\gamma) = \sum_{n\in\mathbb{N}} a_n \gamma^n$. Recalling the explicit form of $\hat{f}$, that is

$$\hat{f}(\gamma) = \frac{1}{\pi}\gamma \arcsin(\gamma) + \frac{1}{\pi}\sqrt{1-\gamma^2} + \frac{1}{2}\gamma\,,$$

we get $a_0 = \frac{1}{\pi}$. Moreover, we have that for all $\gamma \in (-1,1)$

$$\hat{f}'(\gamma) = \frac{1}{\pi}\arcsin\gamma + \frac{1}{2}\,.$$

This yields $a_1 = \hat{f}'(0) = \frac{1}{2}$. Then, noticing that

$$\hat{f}^{(3)}(\gamma) = \frac{\gamma}{\pi(1-\gamma^2)^{3/2}}$$

is an odd function, we get that for all $i \geq 1, a_{2i+1} = 0$. Now let us prove that for all $k \geq 1$, there exist $b_{k,0}, b_{k,1}, ..., b_{k,k-1} > 0$ such that, for all $\gamma \in (-1,1)$,

$$\hat{f}^{(2k)}(\gamma) = \frac{1}{\pi}\sum_{m=0}^{k-1} b_{k,m}\gamma^{2m}(1-\gamma^2)^{-k-m+1/2}\,.$$

We prove this by induction. For $k = 1$, we have that

$$\hat{f}^{(2)}(\gamma) = \frac{1}{\pi}(1-\gamma^2)^{-1/2},$$

so that our claim holds. Assume now that it is true for some $k \geq 1$, let us prove it for $k+1$. It is easy to see that

$$b_{k+1,m} = \begin{cases} 2(2k-1)b_{k,0} + 2b_{k,1} & \text{if } m = 0; \\ 2(4k^2-1)b_{k,0} + 5(2k+1)b_{k,1} + 12b_{k,2} & \text{if } m = 1; \\ 2(m+1)(2m+1)b_{k,m+1} + (4m+1)(2k+2m-1)b_{k,m} & \\ \quad + (2k+2m-3)(2k+2m-1)b_{k,m-1} & \text{if } m \in \{2,3,...,k-1\}; \\ (4k-3)(4k-1)b_{k,k-1} & \text{if } m = k. \end{cases} \tag{A3}$$

The induction is straightforward. In particular, we have shown that $a_{2i} = \frac{\hat{f}^{(2i)}(0)}{(2i)!} = \frac{b_{i,0}}{(2i)!} > 0$.
The conclusion for the coefficients $\alpha$'s of the expansion of $f$ is then trivial. □

Using Lemma A8, it will not be hard to show that $Q_l$ is continuous. The non-negativity of $T(Q_l)$ can be seen as a consequence of the definition of $Q_l$ as the covariance of a Gaussian Process. However, we will give a direct proof of it, so that we can state here a general result which we will need later on.

**Lemma A9.** *Let $C$ be a kernel on $K$, such that $|C(z)| \leq 1$ for all $z \in K$. Consider a non-negative real sequence $\{\alpha_n\}_{n\in\mathbb{N}}$, and assume that*

$$g(\gamma) = \sum_{k=0}^{\infty} \alpha_k \gamma^k$$

*converges uniformly on $[-1,1]$. Then, for all finite Borel measure $\mu$ on $K$, $T_\mu(g(C))$ is a non-negative definite compact operator, and in particular $g(C)$ is a kernel.*

*Proof.* Fix a finite Borel measure $\mu$ on $K$ and notice that $g(C)$ is continuous and symmetric (as uniform limit of continuous and symmetric functions). Moreover, since the Taylor expansion of $g$ around 0 converges uniformly on $[-1,1]$, and since $|C(z)| \leq 1$ for all $z \in K$, we have that $T_\mu(g(C)) = \sum_{k\in\mathbb{N}} \alpha_k T_\mu(C^k)$, the sum converging wrt the operator norm on $L^2(K,\mu)$.
As a consequence of the Schur product theorem[14], the product of two kernels is still a kernel.
As a consequence, it is easy to prove by induction that $T_\mu(C^k)$ is non-negative definite for all $k$. Hence $T_\mu(g(C))$ is the converging limit of a sum of compact non-negative definite operator. We conclude by Lemma A1. □

**Lemma A10.** *For both Standard and Stable ResNet architectures, for any layer $l$, the covariance function $Q_l$ and the correlation function $C_l$ are kernels on $K$, in the sense of Definition 2.*

*Proof.* It is straightforward to prove that $Q_0$ is a kernel. Now let us show that if $Q_l$ is a kernel for some $l$, then $C_l$ is a kernel. Since $Q_l$ is symmetric and so $C_l$ is. Moreover, the diagonal elements of $Q_l$ are continuous by Lemma A3 and do not vanish (since if $\sigma_b = 0$ we are assuming that $0 \notin K$). Hence $C_l$ is continuous. It is then trivial to show that the non-negative definiteness of $T(Q_l)$ implies that $T(C_l)$ is non-negative definite, and so $C_l$ is a kernel if $Q_l$ is.
Now we proceed by induction. Suppose that $Q_{l-1}$ and $C_{l-1}$ are kernels and recall the recursion (5), taking the coefficient $\lambda$ to be 1 in the case of a Standard ResNet. Notice that it can be rewritten as

$$Q_l = Q_{l-1} + \lambda_l^2 \left( {\sigma_b}^2 + \tfrac{{\sigma_w}^2}{2} \hat{f}(C_{l-1})\, R_{l-1} \right),$$

where we have omitted the dependence on $L$ for $\lambda$, we have defined $R_{l-1}(x,x') = \sqrt{Q_{l-1}(x,x)Q_{l-1}(x',x')}$ and $\hat{f}$ is defined in (A2). Clearly $R_{l-1}$ is a kernel. By Lemma A8 and Lemma A9 we have that $\hat{f}(C_l)$ is a kernel. Using the property that sums and products of kernels are kernels (the sum is trivial, cf Footnote 14 for the product), we conclude that $Q_l$, and so $C_l$, is a kernel on $K$. □

[14] Given two matrices $M_1$ and $M_2$, define they're Schur product as the matrix $M = M_1 \circ M_2$, whose elements are $M^{ij} = {M_1}^{ij} {M_2}^{ij}$. If $M_1$ and $M_2$ are non-negative definite, then $M$ is non-negative definite.

### A1.4 Proof of Proposition 2

As always, consider an arbitrary compact set $K \in \mathbb{R}^d$. Assume that $0 \notin K$ if $\sigma_b = 0$. Recall from Appendix A0 that with the notation $\mathcal{H}_Q(K)$ we refer to the RKHS generated by a kernel $Q$ on $K$. We will now prove Proposition 2.

**Proposition 2.** *$\mathcal{H}_{Q_l}(K) \subseteq \mathcal{H}_{Q_{l+1}}(K)$ for all $l \in [0 : L-1]$.*

*Proof.* We have already shown that $T(Q_l) - T(Q_{l-1})$ is non-negative definite in the proof of Lemma A10. We conclude by using the RKHS hierarchy result (see for instance [Paulsen and Raghupathi, 2016] or page 354 in [Aronszajn, 1950]). □

### A1.5 Proof of Lemma 3

We present here the proof of Lemma 3. We have already recalled the Definition 4 of universal kernel in Appendix A0. For convenience of the reader, we restate here the definition of expressive GP.
Let $K$ be a compact in $\mathbb{R}^d$.

**Definition 5** (Expressive GP)**.** *A Gaussian Process on $K$ is said to be expressive on $L^2(K)$ if, denoted by $\psi$ a random realisation, for all $\varphi \in L^2(K)$, for all $\varepsilon > 0$,*

$$\mathbb{P}(\|\psi - \varphi\|_2 \leq \varepsilon) > 0\,.$$

**Lemma 3.** *A universal kernel $Q$ on $K$ induces an expressive GP on $L^2(K)$.*

*Proof.* First, notice that if $Q$ is universal then $T(Q)$ is strictly positive definite [Sriperumbudur et al., 2011] and so all its eigenvalues are strictly positive.
Recall the spectral theorem for compact self-adjoint operators: there is a orthonormal basis of $L^2(K)$ made of the eigenfunctions $\{\psi_n\}_{n\in\mathbb{N}}$ of $T(Q)$. Denoting by $\mu_n > 0$ the eigenvalue of $T(Q)$ relatively to $\psi_n$, since $T(Q)$ is compact we have the equality (Karhunen - Loève decomposition [Grenander, 1950])

$$\psi = \sum_{k=0}^{\infty} Z_k \sqrt{\mu_k}\, \psi_k \sim \mathcal{GP}(0, Q)\,,$$

where $\{Z_k\}_{k\in\mathbb{N}}$ is a family of iid normal random variables, and the series is convergent uniformly on $K$ and in $L^2$ for the stochastic part [Paulsen and Raghupathi, 2016], that is $\lim_{N\to\infty} \sup_{x\in K} \mathbb{E}[(\psi(x) - \sum_{k=0}^{N} Z_k \sqrt{\mu_k}\, \psi_k(x))^2] = 0$ uniformly for $x \in K$. In particular, we get that $\lim_{N\to\infty} \mathbb{E}[\|\psi - \sum_{k=0}^{N} Z_k \sqrt{\mu_k}\psi_k\|_2^2] = 0$. As consequence, for all $\varphi \in L^2(K)$, we have that $\|\sum_{k=0}^{N} Z_k \sqrt{\mu_k}\, \psi_k - \varphi\|_2^2$ converges in squared mean to $\|\psi - \varphi\|_2^2$, for $N \to \infty$.
Now, let $\varphi = \sum_{k=0}^{N} a_k\, \psi_k$ for some finite $N$ and some real coefficients $\{a_0 \dots a_N\}$. We have (with convergence in squared mean)

$$\|\psi - \varphi\|_2^2 = \sum_{k=0}^{N} (Z_k \sqrt{\mu_k} - a_k)^2 + \sum_{k=N+1}^{\infty} \mu_k\, Z_k^2\,.$$

For $k \in [0 : N]$, we can define the interval $I_k = \left[\frac{a_k}{\sqrt{\mu_k}} - \frac{\varepsilon}{\sqrt{2(N+1)\mu_k}}, \frac{a_k}{\sqrt{\mu_k}} + \frac{\varepsilon}{\sqrt{2(N+1)\mu_k}}\right]$, so that, for all $z \in I_k$ we have $(z\sqrt{\mu_k} - a_k)^2 \leq \frac{\varepsilon^2}{2(N+1)}$. Since all these intervals are non empty, we get

$$\mathbb{P}\left(\sum_{k=0}^{N} (Z_k \sqrt{\mu_k} - a_k)^2 \leq \frac{\varepsilon^2}{2}\right) \geq \prod_{k=0}^{N} \mathbb{P}(Z_k \in I_k) > 0\,.$$

On the other hand, we have that

$$\delta_N = \mathbb{E}\left[\sum_{k=N+1}^{\infty} \mu_k\, Z_k^2\right] = \sum_{k=N+1}^{\infty} \mu_k\,.$$

By Mercer's theorem [Paulsen and Raghupathi, 2016], $T(Q)$ is trace class and hence $\delta_N \to 0$ for diverging $N$. By Markov's inequality

$$\mathbb{P}\left(\sum_{k=N+1}^{\infty} \mu_k Z_k^2 \geq \frac{\varepsilon^2}{2}\right) \leq \frac{2\delta_N}{\varepsilon^2}$$

and we can conclude that $\mathbb{P}(\|\psi - \varphi\|_2 \leq \varepsilon) > 0$ for $N$ large enough.

For a general $\varphi = \sum_{k=0}^{\infty} a_k \psi_k$, let $\varphi_N = \sum_{k=0}^{N} a_k \psi_k$. Since $\{\psi_k\}_{k\in\mathbb{N}}$ is a basis of $L^2(K)$, for a fixed $\varepsilon > 0$, it is always possible to find a $N$ such that $\|\varphi - \varphi_N\|_2 \leq \varepsilon/2$ and $\mathbb{P}(\|\varphi_N - \psi\|_2 \leq \varepsilon/2) > 0$, and so we conclude. □

### A1.6 Proof of Proposition 3

In order to prove Proposition 3 we first need a preliminary result, which will be at the core of the proof of Theorem 1 as well.

**Proposition A1.** *Let $K \subset \mathbb{R}^d$ be compact. Assume $\sigma_b > 0$ and let $\tilde{f} : \gamma \mapsto \frac{\gamma}{2} + f(\gamma)$ be defined on $[-1, 1]$. Then the kernel $\tilde{f}(c_0)$, defined point-wise as $\tilde{f}(c_0)(x, x') = \tilde{f}(c_0(x, x'))$, is universal on $K$.*

*Proof.* First notice that $c_0(x, x') = \frac{1+\zeta\, x\cdot x'}{\sqrt{(1+\zeta\,\|x\|^2)(1+\zeta\,\|x'\|^2)}}$, where $\zeta = \sigma_w^2/\sigma_b^2$. For $n \in \mathbb{N}$, define $p_n : (x, x') \mapsto c_0(x, x')^{2n}$, with the convention that $p_0 \equiv 1$. It is easy to verify that $c_0$ is kernel. As a consequence, $p_n$ is a kernel for all $n$, since it is a product of kernels.[15] From Lemma A8, we can write

$$\tilde{f}(c_0) = \sum_{n\in\mathbb{N}} \alpha_n\, p_n\,,$$

the sum converging uniformly on $K^2$, with $\alpha_n > 0$ for all $n \in \mathbb{N}$. By Lemma A9, $\tilde{f}(c_0)$ is a kernel.
Now, for each $n$, we have

$$p_n(x, x') = \frac{1}{(1+\zeta\,\|x\|^2)^n (1+\zeta\,\|x'\|^2)^n} \sum_{k=0}^{2n} \omega_{k,n}\,(x\cdot x')^k\,,$$

where the coefficients $\omega_{k,n}$'s are all strictly positive, explicitly $\omega_{k,n} = \zeta^k \binom{n}{k}$.
Expanding the inner product $x \cdot x'$, we can express $p_n$ in the form

$$p_n(x, x') = \sum_{J\in\mathcal{J}_n} \beta_{J,n}\, A_{J,n}(x) A_{J,n}(x')\,,$$

where $\mathcal{J}_n = \{(j_1 \dots j_d) \in \mathbb{N}^d : \sum_{i=1}^d j_i \in [0 : 2n]\}$, all the coefficients $\beta_{J,n}$'s are strictly positive and the $A_{J,n}$'s are defined as

$$A_{J,n}(x) = \frac{{x_1}^{j_1} \dots {x_d}^{j_d}}{(1+\zeta\,\|x\|^2)^n}\,.$$

Hence we can write $\tilde{f}(c_0)$ as

$$\tilde{f}(c_0)(x, x') = \sum_{n\in\mathbb{N}} \sum_{J\in\mathcal{J}_n} \alpha_n \beta_{J,n} A_{J,n}(x) A_{J,n}(x')\,. \tag{A4}$$

For any $n, n' \in \mathbb{N}$, $J \in \mathcal{J}_n$, $J' \in \mathcal{J}_{n'}$, it is clear that $A_{J,n} A_{J',n'} = A_{J'',n+n'}$, where $J''$ is some element in $\mathcal{J}_{n+n'}$. As a consequence, the linear span of the family $\{A_{J,n}\}_{n\in\mathbb{N}, J\in\mathcal{J}_n}$ is an algebra $\mathcal{A}$ (which is actually a subalgebra of $C(K)$ since all the $A_{J,n}$'s are continuous). Moreover $A_{(0\dots0),0} \equiv 1$, so that $\mathcal{A}$ contains a constant, and it is straightforward to check that $\mathcal{A}$ separates points, that is for all distinct $x, x' \in K$ there exists $a \in \mathcal{A}$ such that $a(x) \neq a(x')$. Then, from Stone-Weierstrass theorem [Lang, 2012], $\mathcal{A}$ is dense in $C(K)$ wrt the uniform norm.
For all $n \in \mathbb{N}$, $J \in \mathcal{J}_n$, let $\theta_{n,J} = \sqrt{\alpha_n \beta_{n,J}}$. Define a bijection $\iota : \mathbb{N} \to \{(n, J) : n \in \mathbb{N}, J \in \mathcal{J}_n\}$ and let $\Phi_n = \theta_{\iota(n)} A_{\iota(n)}$. For all $x \in K$, we have that $\Phi(x) = \{\Phi_n(x)\}_{n\in\mathbb{N}} \in \ell^2$, since $p_n(x, x) < \infty$. We conclude that $\Phi$ is a feature map for $\tilde{f}(c_0)$, and the density of the linear span of $\{\Phi_n\}_{n\in\mathbb{N}}$ allows to claim that the kernel is universal on $K$, in the sense of Definition 4 (cf Theorem 7 in [Micchelli et al., 2006]). □

[15] See footnote 14.

Let $K \subset \mathbb{R}^d$ be an arbitrary compact set. We are now ready to prove Proposition 3.

**Proposition 3.** *If $\sigma_b > 0$, then $Q_2$ is universal on $K$. From Proposition 2, $Q_L$ is universal for all $L \geq 2$.*

*Proof.* Assume $\sigma_b > 0$ and let $K \subset \mathbb{R}^d$ be a compact set. With the notation of Proposition A1, we have that

$$Q_1 = Q_0 + \lambda_{1,L}^2 \left( \sigma_b^2 + \frac{\sigma_w^2}{2} \left( \frac{1}{2} + \frac{\tilde{f}(C_0)}{C_0} \right) Q_0 \right).$$

By proposition A1, we know that the kernel $\tilde{f}(C_0)$ given by $\tilde{f}(C_0)(x,x') = \tilde{f}(C_0(x,x'))$ is universal on $K$. Let us prove that $\frac{\tilde{f}(C_0)}{C_0} Q_0$ is universal. Let $\varepsilon > 0$ and $\varphi \in C(K)$, the space of continuous functions on $K$. Define $\frac{\varphi}{Q_0}(x) = \frac{\varphi(x)}{Q_0(x,x)}$. By the universality of $\tilde{f}(C_0)$, there exists $g \in \mathcal{H}_{\tilde{f}(C_0)}(K)$ such that

$$\left\| g - \frac{\varphi}{\sqrt{Q_0}} \right\|_\infty \leq \epsilon \,.$$

with $g$ can be written as a finite linear combination of the functions $\{\hat{f}(C_0)(x,.)\}_{x \in K}$. This yields

$$\left\| g\sqrt{Q_0} - \varphi \right\|_\infty \leq \epsilon\kappa \,,$$

where $g\sqrt{Q_0}(x) = g(x)\sqrt{Q_0(x,x)}$ and $\kappa = \sup_{x \in K} \sqrt{Q_0(x,x)}$. It is straightforward that $g\sqrt{Q_0} \in \mathcal{H}_{\frac{\tilde{f}(C_0)}{C_0} Q_0}(K)$,[16] Therefore, $\frac{\tilde{f}(C_0)}{C_0} Q_0$ is universal. Since $Q_0$ is non-negative, we have that $Q_1$ is universal by an RKHS hierarchy argument similar to Proposition 2. Using Proposition 2, we conclude that $Q_L$ is universal on $K$. □

### A1.7 Proof of Proposition 4

**Proposition 4.** *Assume $\sigma_b = 0$. Then for all $L \geq 2$, $Q_L$ is universal on $\mathbb{S}^{d-1}$ for $d \geq 2$.*

*Proof.* See the proof of Proposition A7 in Appendix A8. □

### A1.8 Proof of Proposition 5

Proposition 5 is a well known classical result (see for instance Appendix H in [Yang and Salman, 2019] and the references therein. For completeness we give a proof in Appendix A8.

**Proposition 5** (Spectral decomposition on $\mathbb{S}^{d-1}$)**.** *Let $Q$ be a zonal kernel on $\mathbb{S}^{d-1}$, that is $Q(x,x') = p(x \cdot x')$ for a continuous function $p : [-1,1] \to \mathbb{R}$. Then, there is a sequence $\{\mu_k \geq 0\}_{k \in \mathbb{N}}$ such that for all $x, x' \in \mathbb{S}^{d-1}$*

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

*where $\{Y_{k,j}\}_{k \geq 0, j \in [1:N(d,k)]}$ are spherical harmonics of $\mathbb{S}^{d-1}$ and $N(d,k)$ is the number of harmonics of order $k$. With respect to the standard spherical measure, the spherical harmonics form an orthonormal basis of $L^2(\mathbb{S}^{d-1})$ and $T(Q)$ is diagonal on this basis.*

*Proof.* See the proof of Lemma A22 in Appendix A8. □

[16] This is trivial for a function $g$ that can be written as a finite sum of functions of the form $\alpha_i \tilde{f}(C_0)(x_i,.)$, and this would be enough since these functions are dense in $C(K)$ as shown in the proof of Proposition A1. More generally, given two kernels $Q$ and $Q'$, if $h \in \mathcal{H}_Q$ and $h' \in \mathcal{H}_{Q'}$, then $hh' \in \mathcal{H}_{QQ'}$, cf Theorem 5.16 in [Paulsen and Raghupathi, 2016].

### A1.9 Proof of Lemma 4

**Lemma 4.** *Consider a standard ResNet of type (1) and let $K \subset \mathbb{R}^d \setminus \{0\}$ be a compact set. We have that*

$$\lim_{L\to\infty} \sup_{x,x'\in K} |1 - C_L(x,x')| = 0\,.$$

*Moreover, if $\sigma_b = 0$, then,*

$$\sup_{x,x'\in K} |1 - C_L(x,x')| = \mathcal{O}(L^{-2})\,.$$

*Therefore, $\mathcal{H}_{C_\infty}(K)$ is the space of constant functions.*

*Proof.* This result was proven in [Hayou et al., 2019a] in the case of no bias. It was also proven for a slightly different ResNet architecture in [Yang and Schoenholz, 2017].
Consider a ResNet of type (1) and let $K \subset \mathbb{R}^d \setminus \{0\}$ be a compact set. We have that for all $x, x' \in K$

$$Q_L(x,x') = Q_{L-1}(x,x') + \sigma_b^2 + \frac{\sigma_w^2}{2}\hat{f}(C_{L-1}(x,x'))\sqrt{Q_{L-1}(x,x)Q_{L-1}(x',x')}\,.$$

Since $\hat{f}(x) \geq x$, $C_L$ is non-decreasing wrt $L$ and converges to the unique fixed point of $\hat{f}$ which is 1. This convergence is uniform in $x, x'$, i.e. $\lim_{L\to\infty} \sup_{x,x'\in K} 1 - C_L(x,x') = 0$.
Re-writing the recursion yields

$$C_L(x,x') = \delta_L \frac{1}{1+\alpha} C_{L-1}(x,x') + \zeta_L + \delta_L \frac{\alpha}{1+\alpha}\hat{f}(C_{L-1}(x,x'))\,,$$

where $\alpha = \frac{\sigma_w^2}{2}$, $\delta_l = \left(1 + \frac{\sigma_b^2}{(1+\alpha)Q_{L-1}(x,x)}\right)^{-1/2}\left(1 + \frac{\sigma_b^2}{(1+\alpha)Q_{L-1}(x,x)}\right)^{-1/2}$ and $\zeta_L = \sigma_b^2(Q_L(x,x)Q_L(x',x'))^{-1/2}$.
Using Lemma A3, and the boundedness of $C_L$, a simple Taylor expansion yields

$$\begin{aligned} C_L(x,x') &= \frac{1}{1+\alpha} C_{L-1}(x,x') + \frac{\alpha}{1+\alpha}\hat{f}(C_{L-1}(x,x')) + g_L(x,x') \\ &= C_{L-1}(x,x') + \frac{\alpha}{1+\alpha} f(C_{L-1}(x,x')) + g_L(x,x')\,, \end{aligned}$$

where the expansion is uniform on $x, x' \in K$, and $f(x) = \hat{f}(x) - x$, and $g_L = \mathcal{O}(e^{-\beta L})$ for some $\beta > 0$.
The previous dynamical system can be decomposed in two parts, a first part without the term $\mathcal{O}(e^{-\beta L})$ which is the homogeneous system, i.e. the system without bias, and the term $\mathcal{O}(e^{-\beta L})$ which is the contribution of the bias in the dynamical system.
Assume $\sigma_b = 0$, then the term $g_L$ vanishes. Moreover, a Taylor expansion of $\hat{f}$ near 1 yields

$$f(x) = s(1-x)^{3/2} + \mathcal{O}((1-x)^{5/2})\,.$$

Therefore, uniformly in $x, x' \in K$, we have that

$$C_L(x,x') = C_{L-1}(x,x') + \frac{s\alpha}{1+\alpha}(1 - C_{L-1}(x,x'))^{3/2} + \mathcal{O}((1 - C_{L-1}(x,x'))^{5/2})\,.$$

Letting $\gamma_L = 1 - C_L$, a simple Taylor expansion leads to

$$\gamma_L^{-1/2} = \gamma_{L-1}^{-1/2} + \frac{s\alpha}{2(1+\alpha)} + \mathcal{O}(\gamma_{L-1})\,.$$

Therefore, $\gamma_L \sim \kappa L^{-2}$ where $\kappa = \frac{4(1+\alpha)^2}{s^2\alpha^2}$. This equivalence is uniform in $x, x' \in K$.

It is likely that the rate $\mathcal{O}(L^{-2})$ holds without assuming $\sigma_b = 0$. However, the analysis in this requires unnecessarily complicated details. □

## A2 Stable ResNet with uniform scaling

In this section we detail the proofs for the uniform scaling of a Scaled ResNet, that is $\lambda_{l,L} = 1/\sqrt{L}$.
When not otherwise specified, $K$ is a generic compact of $\mathbb{R}^d$. We assume that $0 \notin K$ if $\sigma_b = 0$.

### A2.1 Continuous formulation

We provide the results of existence, uniqueness and regularity of the solution of (8) in Lemma A11. Corollary A1 shows that the differential problem can be restated in the operator space. Eventually we give a proof of Lemma 5, assuring uniform convergence to the continuous limit.

We recall that by continuous formulation we mean a rescaling of the layer index $l$, which becomes a continuous index $t$, spanning the interval $[0,1]$, as the depth diverges, that is $L \to \infty$.
More precisely, for all $L \geq 1$ and all $l \in [0:L]$, we can define $t(l,L) = l/L$.
Consider a sequence $\{l_n, L_n\}_{n\in\mathbb{N}}$ (where, for all $n$, $L_n \geq 1$ and $l \in [0:L_n]$), such that $L_n$ diverges but $l_n/L_n$ converges to a finite $t = \lim_{n\to\infty} t(l_n, L_n)$. We will show in this section (Lemma 5) that the kernels $Q_{l_n|L_n}$ (covariance kernel of the layer $l_n$ in a net with $L_n + 1$ layers) converge uniformly to a kernel, $q_t$, on $K$.
Moreover we can define a differential problem for the mapping $t \mapsto q_t$, with $q \in [0,1]$, that is

$$\begin{aligned} \dot{q}_t(x,x') &= {\sigma_b}^2 + \tfrac{{\sigma_w}^2}{2}\left(1 + \tfrac{f(c_t(x,x'))}{c_t(x,x')}\right) q_t(x,x')\,, \\ q_0(x,x') &= \sigma_b^2 + \sigma_w^2\,\tfrac{x\cdot x'}{d}\,, \\ c_t(x,x') &= \tfrac{q_t(x,x')}{\sqrt{q_t(x,x)q_t(x',x')}}\,. \end{aligned} \tag{8}$$

**Lemma A11** (Existence and uniqueness)**.** *For any $x, x'$ in $K$, the solution of (8) is unique and well defined for all $t \in [0,1]$. The maps $(x,x') \mapsto q_t(x,x')$ and $(x,x') \mapsto c_t(x,x')$ are Lipschitz continuous on $K^2$ and $c_t$ takes values in $[-1,1]$. Moreover, both $q_t$ and $c_t$ are kernels in the sense of Definition 2.*

*Proof.* First notice that from (8) we can find, with few algebraic manipulations, an explicit recurrence relation for the correlation $C_l$, defined in (3). For any $x, x' \in K$ we have

$$\begin{aligned} C_{l+1}(x,x') &= A_{l+1}(x,x')\,C_l(x,x') + \tfrac{\sigma_w^2}{2L}\left(1 + \tfrac{\sigma_w^2}{2L}\right)^{-1} A_{l+1}(x,x')\,f(c_l(x,x')) + \tfrac{1}{L}\,\tfrac{\sigma_b^2}{\sqrt{Q_l(x,x)Q_l(x',x')}}\,; \\ A_l(x,x') &= \sqrt{\left(1 - \tfrac{1}{L}\tfrac{\sigma_b^2}{Q_l(x,x)}\right)\left(1 - \tfrac{1}{L}\tfrac{\sigma_b^2}{Q_l(x',x')}\right)}\,. \end{aligned} \tag{A5}$$

We can find a Cauchy problem for the correlation directly from (8) or by noting that $A_l(x,x') = 1 - \frac{\sigma_b^2}{2L}\left(\frac{1}{Q_l(x,x)} + \frac{1}{Q_l(x',x')}\right) + o(1/L)$, for $L \to \infty$. With both approaches, we have

$$\begin{aligned} \dot{c}_t(x,x') &= {\sigma_b}^2\left(\mathcal{G}_t(x,x') - \mathcal{A}_t(x,x')\,c_t(x,x')\right) + \frac{\sigma_w^2}{2}\,f(c_t(x,x'))\,, \\ c_0(x,x') &= \frac{\sigma_b^2 + \sigma_w^2\,x\cdot x'}{\sqrt{(\sigma_b^2 + \sigma_w^2\,\|x\|^2)(\sigma_b^2 + \sigma_w^2\,\|x'\|^2)}}\,, \end{aligned} \tag{A6}$$

where $f$ is defined in (4) and

$$\mathcal{A}_t(x,x') = \frac{1}{2}\left(\frac{1}{q_t(x,x)} + \frac{1}{q_t(x',x')}\right)\,; \qquad \mathcal{G}_t(x,x') = \sqrt{\frac{1}{q_t(x,x)\,q_t(x',x')}}\,.$$

Note that for the diagonal terms $q_t(x,x)$, (8) reduces to $\dot{q}_t = {\sigma_b}^2 + \frac{{\sigma_w}^2}{2}\,q_t$, whose solution is

$$q_t(x,x) = e^{\frac{{\sigma_w}^2}{2}t}\,q_0(x,x) + \frac{2{\sigma_b}^2}{{\sigma_w}^2}\left(e^{\frac{{\sigma_w}^2}{2}t} - 1\right) = e^{\frac{{\sigma_w}^2}{2}t}\,(\sigma_b^2 + \sigma_w^2\,\|x\|^2) + \frac{2{\sigma_b}^2}{{\sigma_w}^2}\left(e^{\frac{{\sigma_w}^2}{2}t} - 1\right)\,.$$

Now, fix $z = (x,x') \in K^2$ and let $\gamma_0 = c_0(z) \in [-1,1]$. Consider $\bar{f} : \mathbb{R} \to \mathbb{R}$, an arbitrary Lipschitz extension of $f$ to the whole $\mathbb{R}$ and define $H : [0,\infty) \times \mathbb{R} \to \mathbb{R}$ as

$$H(t,\gamma) = \sigma_b^2(\mathcal{G}_t(z) - \mathcal{A}_t(z)\,\gamma) + \frac{\sigma_w^2}{2}\,\bar{f}(\gamma)\,.$$

$H$ is Lipschitz continuous in $\gamma$ and $C^\infty$ in $t$, so there exists $\tau > 0$ such that the Cauchy problem

$$\dot{\gamma}(t) = H(t, \gamma(t))\,;$$
$$\gamma(0) = \gamma_0$$

has a unique $C^1$ solution defined for $t \in [0, \tau)$.
Noticing that

$$\mathcal{G}_t(x, x') - \mathcal{A}_t(x, x') = -\frac{1}{2}\left(\frac{1}{q_t(x,x)} - \frac{1}{q_t(x',x')}\right)^2 \leq 0\,,$$

we get that for all $t_1$ such that $\gamma(t_1) = 1$ we have $\dot{\gamma}(t_1) \leq 0$, since $f(1) = 0$, and for all $t_{-1}$ such that $\gamma(t_{-1}) = -1$ we have $\dot{\gamma}(t_{-1}) = \sigma_b^2(\mathcal{G}_t(x, x') + \mathcal{A}_t(x, x')) + \frac{\sigma_w^2}{2} > 0$. As a consequence $\gamma(t) \in [-1, 1]$ for all $t \in [0, \tau)$ and we can take $\tau = \infty$.
In particular we get that (A6) has a unique solution $t \mapsto c_t(z)$, defined for $t \in [0, 1]$ and bounded in $[-1, 1]$.
As a consequence, (8) has a unique and well defined solution for all $t \geq 0$.

Now notice that $z \mapsto c_0(z)$ is Lipschitz on $K^2$. let us denote as $L_0$ a Lipschitz constant for $c_0$.
Since both $\mathcal{G}_t$ and $\mathcal{A}_t$ are $C^1$, we can find real constants $L_G$, $L_A$ and $M_A$ such that for all $z, z'$ elements of $K^2$

$$|\mathcal{G}_t(z) - \mathcal{G}_t(z')| \leq L_G \|z - z'\|\,;$$
$$|\mathcal{A}_t(z) - \mathcal{A}_t(z')| \leq L_A \|z - z'\|\,;$$
$$|\mathcal{A}_t(z)| \leq M_A\,.$$

Let $L_f$ be a Lipschitz constant for $f$. Using the fact that $|c_t| \leq 1$, we can write

$$|\dot{c}_t(z) - \dot{c}_t(z')| \leq L_1 \|z - z'\| + L_2 |c_t(z) - c_t(z')|\,,$$

where $L_1 = \sigma_b^2(L_G + L_A)$ and $L_2 = \sigma_b^2 M_A + \frac{\sigma_w^2}{2} L_f$.
Now fix $z$ and $z'$ and consider $\Delta(t) = c_t(z) - c_t(z')$. We have

$$|\dot{\Delta}(t)| \leq L_1 \|z - z'\| + L_2 |\Delta(t)|\,;$$
$$|\Delta(0)| \leq L_0 \|z - z'\|\,.$$

So $|\Delta(t)| \leq \left(\frac{L_1}{L_2}\left(e^{L_2 t} - 1\right) + L_0\, e^{L_2 t}\right)\|z - z'\|$, meaning that $c_t$ (and so $q_t$) is Lipschitz on $L^2$.

Since the mapping $(x, x') \mapsto q_t(x, x')$ is continuous, it defines a compact integral operator $T(q_t)$ on $L^2(K)$ [Lang, 2012]. Since $q_t$ is real and symmetric under the swap of $x$ and $x'$, the operator is self-adjoint. The same holds true for $c_t$.

The fact that $T(q_t)$ is a non-negative operator can be seen as a corollary of Lemma 5. Indeed all $T(Q_{l|L})$ is a non-negative definite operator, since it is induced by a kernel. Hence, for each $t \in [0, 1]$ it is enough to find a sequence $\{l_n, L_n\}_{n\in\mathbb{N}}$ (where $L_n \geq 1$ is an integer and $l_n \in [0 : L_n]$) such that $L_n \to \infty$ and $l_n/L_n \to t$. By Lemma 5, $T(Q_{l_n|L_n}) \to T(q_t)$ in the $L^\infty$ norm, and hence in $L^2$, as we are on a compact set. By Lemma A10, for all $n \in \mathbb{N}$ we have that $T(Q_{l_n|L_n})$ is non-negative definite. Since the subspace of non-negative definite operators in $L^2$ is closed wrt the $L^2$ operator norm, we conclude.
Once we have established that $T(q_t)$ is non-negative definite, it follows immediately that $T(c_t)$ is non-negative as well. Since these results hold for any arbitrary finite Borel measure $\mu$ on $K$, we can thus conclude by Lemma A1 that both $q_t$ and $c_t$ are kernels, in the sense of Definition 2. □

**Corollary A1.** *The maps $t \mapsto T(q_t)$ and $t \mapsto T(c_t)$, defined on $[0, 1]$, are continuous and twice differentiable with respect to the operator norm in $L^2(K)$. Moreover, $\frac{\mathrm{d}}{\mathrm{d}t}T(q_t) = T(\dot{q}_t)$, $\frac{\mathrm{d}}{\mathrm{d}t}T(c_t) = T(\dot{c}_t)$, $\frac{\mathrm{d}^2}{\mathrm{d}t^2}T(q_t) = T(\ddot{q}_t)$ and $\frac{\mathrm{d}^2}{\mathrm{d}t^2}T(c_t) = T(\ddot{c}_t)$.*

*Proof.* Consider the map $(t,z) \mapsto q_t(z)$, defined on $[0,1] \times K^2$, which is continuous wrt $z$ and $C^2$ wrt $t$, as it can be easily checked. Since $K^2$ and $[0,1]$ are compact sets, it follows that for any $t$

$$\lim_{s\to t} \sup_{z\in K^2} \left| \frac{q_s(z) - q_t(z)}{s-t} - \dot{q}_t(z) \right| = \sup_{z\in K^2} \lim_{s\to t} \left| \frac{q_s(z) - q_t(z)}{s-t} - \dot{q}_t(z) \right| = 0\,.$$

Hence $\lim_{s\to t} \frac{q_s - q_t}{t-s} = \dot{q}_t$ uniformly on $K^2$, and hence $\lim_{s\to t} \frac{T(q_s)-T(q_t)}{t-s} = T(\dot{q}_t)$ in the $L^2(K,\mu)$ norm for operators, since $K$ is compact.
The proof for the second derivative works in the same way, using the fact that $(t,z) \mapsto q_t(z)$ is continuous in $z$ and $C^1$ in $t$.
As a consequence of the above results, $t \mapsto T(q_t)$ is continuous and twice differentiable, with $\frac{\mathrm{d}}{\mathrm{d}t} T(q_t) = T(\dot{q}_t)$ and $\frac{\mathrm{d}^2}{\mathrm{d}t^2} T(q_t) = T(\ddot{q}_t)$.
The proof for $T(c_t)$ is analogous. □

**Lemma 5** (Convergence to the continuous limit)**.** *Let $Q_{l|L}$ be the covariance kernel of the layer $l$ in a net of $L+1$ layers $[0:L]$, and $q_t$ be the solution of (8), then*

$$\lim_{L\to\infty} \sup_{l\in[0:L]} \sup_{(x,x')\in K^2} |Q_{l|L}(x,x') - q_{t=l/L}(x,x')| = 0\,.$$

*Proof.* We will show that the relation holds for $c_t$, and hence for $q_t$.
Let $H$, defined on $[0,1] \times K^2$, be such that $\dot{c}_t(z) = H(z,t,c_t(z))$. Explicitly, with the same notations as in (A6), we have

$$H(z,t,\gamma) = \sigma_b^2(\mathcal{G}_t(z) - \mathcal{A}_t(z)\,\gamma) + \frac{\sigma_w^2}{2} f(\gamma)\,.$$

Define

$$\tau(h) = \sup_{t,z} \left| \frac{c_{t+h}(z) - c_t(z)}{h} - H(z,t,c_t(z)) \right|.$$

Since $t$ and $z$ takes values on compact sets, by uniform continuity, fixed $h$ we can write, for $h \to 0$

$$\sup_t \sup_{s\in[t,t+h]} |H(z,s,c_s(z)) - H(z,t,c_t(z))| = o(h)\,.$$

Hence, since $\tau$ can be rewritten as $\tau(h) = \frac{1}{h} \sup_{t,z} \left| \int_t^{t+h} (H(z,s,c_s(z)) - H(z,t,c_t(z))\,\mathrm{d}s \right|$, it is clear that $\tau(h) \to 0$ for $h \to 0$.
Now, for any integer $L \geq 1$, let $\tilde{H}_L : K^2 \times [0:L-1] \times [-1,1]$ be given by

$$\tilde{H}_L(z,l,\gamma) = (A_{l+1|L}(x,x') - 1)\,L\,\gamma + \tfrac{\sigma_w^2}{2} \left(1 + \tfrac{\sigma_w^2}{2L}\right)^{-1} A_{l+1|L}(x,x')\, f(c_l(x,x')) + \frac{\sigma_b^2}{\sqrt{Q_{l|L}(x,x) Q_{l|L}(x',x')}}\,,$$

where,

$$A_{l|L}(x,x') = \sqrt{\left(1 - \tfrac{1}{L}\tfrac{\sigma_b^2}{Q_{l|L}(x,x)}\right)\left(1 - \tfrac{1}{L}\tfrac{\sigma_b^2}{Q_{l|L}(x',x')}\right)}\,.$$

It is clear from (A5) that $\tilde{H}_L$ has been defined so that $C_{l+1|L}(z) - C_{l|L}(z) = \frac{1}{L}\tilde{H}_L(z,l,\gamma)$, for all $L \in [0:L-1]$ and all $z \in K^2$. Using the explicit form of the diagonal terms of $Q$ and $q$, it can be easily shown that, for $L \to \infty$,

$$\sup_{(x,x')\in K^2} \sup_{l\in[0:L-1]} A_{l+1|L}(x,x') = 1 + \tfrac{\sigma_b^2}{L}\mathcal{A}_{t=l/L}(x,x') + O(1/L^2)\,;$$

$$\sup_{(x,x')\in K^2} \sup_{l\in[0:L]} \frac{\sigma_b^2}{\sqrt{Q_{l|L}(x,x) Q_{l|L}(x',x')}} = \mathcal{G}_{t=l/L}(x,x') + O(1/L^2)\,,$$

where $\mathcal{A}_t$ and $\mathcal{G}_t$ are defined as in (A6). As a consequence, we can find a constant $M_1 > 0$ and an integer $L_\star > 0$ such that, for all $\gamma \in [-1,1]$, for all $z \in K^2$, for all $L \geq L^\star$

$$|\tilde{H}_L(z,l,\gamma) - H(z,l/L,\gamma)| \leq \frac{M_1}{L}\,. \tag{A7}$$

Moreover, there exists a constant $M_2 > 0$ such that for all $z \in K^2$, all $t \in [0,1]$ and all pairs $(\gamma, \gamma') \in [-1,1]^2$

$$|H(z,t,\gamma) - H(z,t,\gamma')| \leq M_2 \|\gamma - \gamma'\| \,. \tag{A8}$$

Thanks to the two above uniform inequalities, we will now show that, for $L \geq L_\star$,

$$\sup_{l \in [0:L]} \sup_{z \in K^2} |C_{l|L}(x,x') - c_{t(l,L)}(x,x')| \leq \tilde{\tau}(1/L) \frac{e^{M_2} - 1}{M_2} \,, \tag{A9}$$

where $\tilde{\tau} : h \mapsto \tau(h) + M_1 h$.
To do so, fix $L \geq L_\star$ and define $\Delta_l = \sup_{z \in K^2} |C_{l|L}(x,x') - c_{t(l,L)}(x,x')|$. Using the definition of $\tau$, (A7) and (A8) we get

$$|\Delta_{l+1}| \leq \left(1 + \tfrac{M_2}{L}\right) |\Delta_l| + \tfrac{1}{L}\tau(1/L) + \tfrac{M_1}{L} = \left(1 + \tfrac{M_2}{L}\right) |\Delta_l| + \tfrac{1}{L}\tilde{\tau}(1/L) \,.$$

At this point, using the fact that $\Delta_0 = 0$, it is easy to show by induction that

$$\Delta_l \leq \tilde{\tau}(1/L) \, \frac{\left(1 + \frac{M_2}{L}\right)^l - 1}{M_2} \,,$$

and so (A9) follows.
Finally, the uniform convergence of $C$ to $c$ implies the one of $Q$ to $q$ and so we conclude. □

### A2.2 Universality of the covariance kernel

We will now prove the results of universality of Theorem 1 and Proposition 6.

**Proof of Theorem 1**

The idea is to prove that for any finite Borel measure $\mu$ on $K$, the operator $T_\mu(q_t)$ is strictly positive definite if $t > 0$, and then use the characterization of universal kernels given in Lemma A2.
To prove the strict positive definiteness, we will proceed in two steps. First we show in Proposition A2 that for all non-zero $\varphi \in L^2(K,\mu)$, $\langle T_\mu(q_t)\,\varphi, \varphi \rangle > 0$ for $t$ small enough. Then we use Proposition A3, which shows that $\frac{\mathrm{d}}{\mathrm{d}t} T_\mu(q_t)$ is non-negative definite.

**Proposition A2.** *Fix any finite Borel measure $\mu$ on $K$, and assume that $\sigma_b > 0$. Given any non-zero $\varphi \in L^2(K,\mu)$, there exists a $t_\varphi \in (0,1]$ such that $\langle T_\mu(q_t)\,\varphi, \varphi \rangle > 0$, for all $t \in (0, t_\varphi)$.*

*Proof.* From Corollary A1, we can expand $T_\mu(q_t)$ around $t = 0$ as

$$T_\mu(q_t) = T_\mu(q_0) + t\, T_\mu(\dot{q}_0) + o(t) = t\, T_\mu\left(\sigma_b^2 + \frac{\sigma_w^2}{2} q_0\right) + T_\mu((c_0 + t f(c_0)) R_0) + o(t) \,,$$

the $o(t)$ being wrt the operator norm, where we have defined the kernel $R_0$ via $R_0(x,x') = \frac{\sigma_w^2}{2}\sqrt{(1+\zeta\|x\|^2)(1+\zeta\|x'\|^2)}$.
Since $T_\mu(q_0)$ is non-negative, for any $\varphi \in L^2(I)$, we have

$$\langle T_\mu(q_t)\,\varphi, \varphi \rangle \geq \langle T_\mu((c_0 + t f(c_0)) R_0)\,\varphi, \varphi \rangle + o(t) = \left(1 - \frac{t}{2}\right) \langle T_\mu(c_0)\,\psi, \psi \rangle + t\, \langle T_\mu(f(c_0)))\,\psi, \psi \rangle + o(t) \,,$$

where $\psi(x) = \sigma_w \sqrt{(1+\zeta\|x\|^2)/2}\,\varphi(x)$. We conclude by the strict positivity of $\tilde{f}(c_0)$ on $L^2(K,\mu)$, thanks to Proposition A1 and Lemma A2. □

**Proposition A3.** *For any finite Borel measure $\mu$ on $K$, for any $t \in [0,1]$, the operator $T_\mu(\dot{q}_t)$ on $L^2(K,\mu)$ is non-negative definite. In particular, for all $\varphi \in L^2(K,\mu)$ we have*

$$\tfrac{\mathrm{d}}{\mathrm{d}t} \langle T_\mu(q_t)\,\varphi, \varphi \rangle \geq 0 \,.$$

*Proof.* Fix $\mu$ and $\varphi \in L^2(K,\mu)$. From (8) we can write

$$T_\mu(\dot{q}_t) = T_\mu\left(\sigma_b^2 + \frac{\sigma_w^2}{2}\, q_t + \frac{\sigma_w^2}{2}\,\frac{f(c_t)}{c_t} q_t\right).$$

By Lemma A11, $T_\mu(q_t)$ is non-negative definite, so we can write

$$\begin{aligned}\langle T_\mu(\dot{q}_t)\,\varphi,\varphi\rangle &= \sigma_b^2|\langle 1,\varphi\rangle|^2 + \frac{\sigma_w^2}{2}\left\langle T_\mu\left(\frac{c_t+f(c_t)}{c_t}\, q_t\right)\varphi,\varphi\right\rangle \\ &\geq \frac{\sigma_w^2}{2}\left\langle T_\mu\left(\tilde{f}(c_t)\,\frac{q_t}{c_t}\right)\varphi,\varphi\right\rangle \\ &= \frac{\sigma_b^2}{2}\langle T_\mu(\tilde{f}(c_t))\,\psi,\psi\rangle\,,\end{aligned}$$

where $\tilde{f}:\gamma\mapsto\frac{\gamma}{2}+f(\gamma)$, for $\gamma\in[-1,1]$, and $\psi(x)=\sqrt{q_t(x,x)}\,\varphi(x)$. By Lemma A8, the Taylor expansion of $\tilde{f}$ around 0 converges uniformly on $[-1,1]$, and all its coefficients are non-negative. We conclude by Lemma A9 that $T_\mu(\dot{q}_t)$ is non-negative definite.
Finally, to prove the inequality, it is enough to recall that $\frac{\mathrm{d}}{\mathrm{d}t}T_\mu(q_t)=T_\mu(\dot{q}_t)$ by Corollary A1, the derivative $\frac{\mathrm{d}}{\mathrm{d}t}$ being wrt the operator norm on $L^2(K,\mu)$. □

**Theorem 1** (Universality of $q_t$)**.** *Let $K\subset\mathbb{R}^d$ be compact and assume $\sigma_b>0$. For any $t\in(0,1]$, the solution $q_t$ of (8) is a universal kernel on $K$.*

*Proof.* By Lemma A2, it suffices to show that for any finite Borel measure $\mu$ on $K$, $T_\mu(q_t)$ is strictly positive definite for all $t\in(0,1]$. Fix any nonzero $\varphi\in L^2(K,\mu)$, define the map $F$ on $[0,1]$ by $F(t)=\langle T_\mu(q_t)\,\varphi,\varphi\rangle$. For any fixed $t\in(0,1]$, by Proposition A2 we can find $s\in(0,t)$ such that $F(s)>0$. Since $F$ is non decreasing by Proposition A3, we get that $F_t>0$. Hence $T_\mu(q_t)$ is strictly positive definite. □

### Proof of Proposition 6

The proof of Proposition 6 is quite similar to the one of Theorem 1.
Using Lemma A15 instead of Lemma A2, we will not need to consider a generic finite Borel measure $\mu$ on $\mathbb{S}^{d-1}$, but it will be enough to show that $T_\nu(q_t)$ is a stricly positive operator on $L^2(\mathbb{S}^{d-1},\nu)$, where $\nu$ is the standard unifrom spherical measure on $\mathbb{S}^{d-1}$.
Since $\sigma_b=0$, we will not be able to use Proposition A1. We will hence state some preliminary results.

**Lemma A12.** *Let $\{A_n\}_{n\in\mathbb{N}}$ be a family of compact non-negative operators on a separable Hilbert space $\mathcal{H}$. Let $R_n$ be the range of $A_n$ and assume that $V=\mathrm{Span}(\bigcup_{n\in\mathbb{N}}R_n)$ is dense in $\mathcal{H}$. Let $\{\alpha_n\}_{n\in\mathbb{N}}$ be a strictly positive sequence such that the sum*

$$A=\sum_{n\in\mathbb{N}}\alpha_n\,A_n$$

*converges in the operator norm. Then $A$ is a compact strictly positive definite operator.*

*Proof.* $A$ is the convergent limit of a sum of compact self-adjoint operators and hence it is compact and self-adjoint.
Now, fix an arbitrary nonzero $h\in\mathcal{H}$. To show that $A$ is strictly positive it is enough to prove that $\langle A\,h,h\rangle>0$.
Denote by $V_N$ the linear span of $\bigcup_{n\in[0:N]}R_n$. Since $V_N\subseteq V_{N+1}$ for all $N$, and $\bigcup_{N\in\mathbb{N}}V_N=V$ is dense in $H$, there exists a sequence $\{h_N\}_{N\in\mathbb{N}}$ converging to $h$ and such that $h_N\in V_N$ for all $N$.
Now let us show that there must exist $n^\star\in\mathbb{N}$ such that $A_{n^\star}\,h\neq 0$. Since $\lim_{N\to\infty}\langle h,h_N\rangle=\langle h,h\rangle>0$, there must be a $N^\star$ such that $\langle h,h_{N^\star}\rangle>0$ and so there exists $n^\star\in[0:N^\star]$ and $h_{n^\star}\in V_{n^\star}$ such that $\langle h,h_{n^\star}\rangle\neq 0$. In particular, $h$ is not orthogonal to $R_{n^\star}$ and cannot lie in the nullspace of $A_{n^\star}$, using the fact that $A_{n^\star}$ is compact and self-adjoint and so its range and its nullspace are orthogonal [Lang, 2012].
Using the spectral decomposition of non-negative compact operators, it is straightforward that $A_{n^\star}\,h\neq 0$ implies that $\langle A_{n^\star}\,h,h\rangle>0$. Now, since $A_n$ is non-negative and $\alpha_n>0$ for all $n$, we have

$$\langle A\,h,h\rangle=\sum_{n\in\mathbb{N}}\alpha_n\langle A_n\,h,h\rangle\geq\alpha_{n^\star}\langle A_{n^\star}\,h,h\rangle>0\,,$$

and so we conclude. □

**Lemma A13.** *For all $n \in \mathbb{N}$, consider the kernel $p_n$ on $\mathbb{S}^{d-1}$, defined by $p_n(x, x') = (x \cdot x')^n$, and let $T_\nu(p_n)$ be the induced integral operator on $L^2(\mathbb{S}^{d-1}, \nu)$. Denoting as $R_n$ the range of $T_\nu(p_n)$, the subspace $V = \mathrm{Span}\left(\bigcup_{n\in\mathbb{N}} R_n\right)$ is dense in $L^2(\mathbb{S}^{d-1}, \nu)$.*
*Moreover, letting $V' = \mathrm{Span}\left(\bigcup_{n\in\mathbb{N}} R_{2n}\right)$ and $V'' = \mathrm{Span}\left(\bigcup_{n\in\mathbb{N}} R_{2n+1}\right)$, we have $L^2(\mathbb{S}^{d-1}, \nu) = \overline{V'} \oplus \overline{V''}$, the overline denoting the closure in $L^2(\mathbb{S}^{d-1}, \nu)$.*

*Proof.* To prove that $V$ is dense, first notice that for each spherical harmonic $Y$, we can find an operator in the form $T_\nu(P(x \cdot x'))$, for a polynomial $P$, which has $Y$ in its range. Since the range of such an operator is trivially contained in $V$, it follows that $V$ contains all the spherical harmonics, and so it is dense in $L^2(\mathbb{S}^{d-1}, \nu)$.
Now, note that for any even $n$ and odd $n'$ we have

$$\int_{\mathbb{S}^{d-1}} (x \cdot z)^n (z \cdot x')^{n'} \mathrm{d}\nu(z) = 0\,,$$

by an elementary symmetry argument, since it is the integral on the sphere of a homogeneous polynomial of odd degree $n + n'$ in the components $z_i$'s of $z$.
It follows that $V'$ and $V''$ are orthogonal. Since their union $V$ is dense, we conclude that $L^2(\mathbb{S}^{d-1}, \nu) = \overline{V'} \oplus \overline{V''}$. □

**Corollary A2.** *With the notations of Lemma A13, assume that a sequence $\{\alpha_{n\in\mathbb{N}}\}$ is such that $A = \sum_{n\in\mathbb{N}} \alpha_n T_\nu(p_n)$ converges wrt the operator norm on $L^2(\mathbb{S}^{d-1}, \nu)$. Then $A = A' + A''$, where $A' : \overline{V'} \to \overline{V'}$ and $A'' : \overline{V''} \to \overline{V''}$. Such a decomposition is unique and*

$$A' = \sum_{n\in\mathbb{N}} \alpha_{2n}\, T_\nu(p_{2n})\,; \qquad\qquad A'' = \sum_{n\in\mathbb{N}} \alpha_{2n+1}\, T_\nu(p_{2n+1})\,,$$

*both sums converging wrt the operator norm.*

*Proof.* It is clear that $A = A' + A''$, when both $A'$ and $A''$ are defind on the whole $L^2(\mathbb{S}^{d-1}, \nu)$.
Consider any $\varphi \in L^2(\mathbb{S}^{d-1}, \nu)$. We have $A'\varphi \in \overline{V'}$, since $T_\nu(p_{2n})\varphi \in \overline{V'}$ for all $n$. Analogously, we can show that $A''\varphi \in \overline{V''}$. To conclude that we can consider the restrictions of $A'$ and $A''$ to $\overline{V'}$ and $\overline{V''}$ respectively, it is enough to recall that for compact self adjoint operators the nullspace is the orthogonal of the closure of the range [Lang, 2012], so that the nullspace of $A'$ contains $\overline{V''}$ and the nullspace of $A''$ contains $\overline{V'}$. □

**Lemma A14.** *The function $f : [-1, 1] \to \mathbb{R}$, defined in (4), is an analytic function on $(-1, 1)$, whose expansion $f(\gamma) = \sum_{n\in\mathbb{N}} \alpha_n \gamma^n$ converges absolutely on $[-1, 1]$. Moreover, $\alpha_n > 0$ for all even $n \in \mathbb{N}$, $\alpha_1 = -1/2$ and $\alpha_n = 0$ for all odd $n \geq 3$.*
*Let $g : [-1, 1] \to \mathbb{R}$ be defined as $g(\gamma) = f(\gamma)f'(\gamma)$. $g$ is analytic on $(-1, 1)$ and its expansion $g(\gamma) = \sum_{n\in\mathbb{N}} \beta_n \gamma^n$ converges absolutely on $[-1, 1]$. Moreover, for all odd $n \in \mathbb{N}$ the coefficient $\beta_n$ is strictly positive.*

*Proof.* The claims for $f$ have been already proven in Lemma A8. As for $g$, the analyticity of $f$ implies the one of $f'$, and it is easy to check the convergence on $[-1, 1]$. Moreover, all the odd Taylor coefficients of $f'$ are stricly positive, as the even coefficients of $f$ are. It follows that $\beta_n > 0$ for all odd $n$. □

**Proposition A4.** *Given any non-zero $\varphi \in L^2(\mathbb{S}^{d-1}, \nu)$, there exists a $t_\varphi \in (0, 1]$ such that $\langle T_\nu(q_t)\,\varphi, \varphi\rangle > 0$, for all $t \in (0, t_\varphi)$.*

*Proof.* The case $\sigma_b > 0$ has been already established in Proposition A2, hence suppose that $\sigma_b = 0$.
First recall (A6)

$$\dot{c}_t = \frac{\sigma_w^2}{2} f(c_t)\,. \tag{A10}$$

Deriving once more we have

$$\ddot{c}_t = g(c_t)\,, \tag{A11}$$

where $g = ff'$ as in Lemma A14.
Define the kernels $p_n$'s, and the subspaces $V'$ and $V''$ of $L^2(\mathbb{S}^{d-1}, \nu)$, as in Lemma A13. By (A10) and (A11) we can write

$$c_t = c_0 + t\,\dot{c}_0 + \tfrac{t^2}{2}\,\ddot{c}_0 + o(t^2) = c_0 + t\,f(c_0) + \tfrac{t^2}{2}\,g(c_0) + o(t^2)\,.$$

Since $\sigma_b = 0$, we have that $c_0(x, x') = x \cdot x'$, so that $c_0 = p_1$.
From Lemma A14, $T_\nu(\dot{c}_0) = \sum_{n\in\mathbb{N}} \alpha_n\, T_\nu(p_n)$ and $T_\nu(\ddot{c}_0) = \sum_{n\in\mathbb{N}} \beta_n\, T_\nu(p_n)$, both sums converging in the operator norm. Moreover, $\alpha_n > 0$ for all even $n$ and $\alpha_n = 0$ for all odd $n \geq 3$, whilst $\beta_n > 0$ for all odd $n$.
In particular, by Corollary A2 and Lemma A12, we deduce that the restriction of $T_\nu(\dot{c}_0)|_{\overline{V'}} : \overline{V'} \to \overline{V'}$ is well defined and strictly positive, and the same holds true for the restriction $T_\nu(\ddot{c}_0)|_{\overline{V''}} : \overline{V''} \to \overline{V''}$.
Now fix a non-zero $\varphi \in L^2(\mathbb{S}^{d-1}, \nu)$. By Lemma A13, we can write $\varphi = \varphi' + \varphi''$, with $\varphi' \in \overline{V'}$, $\varphi'' \in \overline{V''}$ uniquely determined.
First, suppose that $\varphi' \neq 0$. Using Corollary A1 and recalling that $c_0 = p_1$, we get

$$\langle T_\nu(c_t)\,\varphi, \varphi\rangle = t\langle T_\nu(\dot{c}_0)|_{\overline{V'}}\,\varphi', \varphi'\rangle + \langle (1 + t\,\alpha_1) T_\nu(p_1)\,\varphi'', \varphi''\rangle + o(t) > 0\,,$$

for $t$ small enough.
On the other hand, for $\varphi' = 0$, we have $\varphi = \varphi''$ and so

$$\langle T_\nu(c_t)\,\varphi, \varphi\rangle = \langle (1 + t\,\alpha_1) T_\nu(p_1)\,\varphi'', \varphi''\rangle + \tfrac{t^2}{2}\,\langle T_\nu(\ddot{c}_0)|_{\overline{V''}}\,\varphi'', \varphi''\rangle + o(t^2) > 0$$

for $t$ small enough.
So there is a $t_\varphi$ such that, for $t \in (0, t_\varphi)$, $\langle T_\nu(c_t)\,\varphi, \varphi\rangle > 0$. It follows immediately that the same property is true for $T_\nu(q_t)$. □

**Lemma A15.** *Let $Q$ be a kernel on $\mathbb{S}^{d-1}$. Then $Q$ is universal on $\mathbb{S}^{d-1}$ if and only if $T_\nu(Q)$ is strictly positive definite on $L^2(\mathbb{S}^{d-1}, \nu)$.*

*Proof.* If $Q$ is universal, $T_\nu(Q)$ is strictly positive definite by Lemma A2. On the other hand, if $T_\nu(Q)$ is strictly positive definite, by Proposition 5 its range contains all the spherical harmonics. Since the RKHS generated by $Q$ contains the range of $T_\nu(Q)$ (Proposition 11.17 in [Paulsen and Raghupathi, 2016]), it contains the linear span of the spherical harmonics, which is dense in $C(\mathbb{S}^{d-1})$ [Kounchev, 2001]. Hence $Q$ is universal. □

**Proposition 6** (Universality on $\mathbb{S}^{d-1}$)**.** *For any $t \in (0, 1]$, the covariance kernel $q_t$, solution of (8) with $\sigma_b = 0$, is universal on $\mathbb{S}^{d-1}$, with $d \geq 2$.*

*Proof.* Proceeding as in the proof of Theorem 1, using Proposition A3 and Proposition A4 we can show that $T_\nu(q_t)$ is strictly posititive definite on $L^2(\mathbb{S}^{d-1}, \nu)$ for all $t \in (0, 1]$. We conclude by Lemma A15 that $q_t$ is universal on $\mathbb{S}^{d-1}$. □

## A3 Stable ResNet with decreasing scaling

### A3.1 Proof of Proposition 7

**Proposition 7** (Uniform Convergence of the Kernel)**.** *Consider a Stable ResNet with a decreasing scaling, i.e. the sequence $\{\lambda_l\}_{l\geq 1}$ is such that $\sum_l \lambda_l^2 < \infty$. Then for all $(\sigma_b, \sigma_w) \in \mathbb{R}^+ \times (\mathbb{R}^+)^*$, there exists a kernel $Q_\infty$ on $\mathbb{R}^d$ such that for any compact set $K \subset \mathbb{R}^d$,*

$$\sup_{x,x'\in K} |Q_L(x, x') - Q_\infty(x, x')| = \Theta\big(\textstyle\sum_{k\geq L} \lambda_k^2\big)\,.$$

*Proof.* Let $x, x' \in \mathbb{R}^d$. The kernel $Q_l$ is given recursively by the formula

$$Q_l(x, x') = Q_{l-1}(x, x') + \lambda_l^2\sigma_b^2 + \frac{\sigma_w^2\lambda_l^2}{2}\,\hat{f}(C_{l-1}(x, x'))\sqrt{Q_{l-1}(x, x)}\sqrt{Q_{l-1}(x', x')}\,,$$

where $\hat{f}(t) = 2\mathbb{E}[\phi'(Z_1)\phi'(tZ_1 + \sqrt{1-t^2}Z_2)] = t + f(t)$ and $Z_1, Z_2$ are iid standard Gaussian variables. In particular, we have

$$Q_l(x,x) = \lambda_l^2\sigma_b^2 + (1 + \frac{\sigma_w^2\lambda_l^2}{2})Q_{l-1}(x,x)\,.$$

which brings

$$Q_l(x,x) + \frac{2\sigma_b^2}{\sigma_w^2} = (1 + \frac{\sigma_w^2\lambda_l^2}{2})(Q_{l-1}(x,x) + \frac{2\sigma_b^2}{\sigma_w^2})\,,$$

Therefore, we can assume without loss of generality that $\sigma_b = 0$. This yields

$$C_l(x,x') = \frac{1}{1+\frac{\lambda_l\sigma_w^2}{2}}C_{l-1}(x,x') + \frac{\frac{\sigma_w^2\lambda_l^2}{2}}{1+\frac{\lambda_l\sigma_w^2}{2}}\hat{f}(C^{l-1}(x,x'))\,.$$

Letting $\alpha_l = \frac{\sigma_w^2\lambda_l^2}{2}$ and $C_l := C_l(x,x')$, we have that

$$C_l = \frac{1}{1+\alpha_l}C_{l-1} + \frac{\alpha_l}{1+\alpha_l}\hat{f}(C_{l-1})\,.$$

Since $\hat{f}$ is non decreasing, $C^l$ is non-decreasing and has a limit $C_\infty(x,x') \leq 1$.

Now let us prove that the convergence of $C_l$ to $C_\infty$ happens uniformly with a rate $\sum_{k\geq l}\lambda_l^2$. Using the recursive formula of $C_l$, and knowing that we have that

$$C_\infty - C_l = \frac{1}{1+\alpha_l}(C_\infty - C_{l-1}) + \frac{\alpha_l}{1+\alpha_l}(C_\infty - f(C_{l-1}))\,.$$

Letting $\delta_l = C_\infty - C_l$, it is easy to see that, uniformly in $x, x' \in \mathbb{R}^d$, we have that

$$\delta_l = \delta_{l-1} + \alpha_l + o(\alpha_l)\,.$$

Therefore, using the fact that $C_l \leq C_\infty$, we have

$$\sup_{(x,x')\in\mathbb{R}^d}|C_l(x,x') - C_\infty(x,x')| = \mathcal{O}\left(\sum_{k\geq l}\alpha_k\right)\,.$$

Moreover, we know that

$$Q_l(x,x) = Q_0(x,x)\prod_{k=1}^{l}(1+\alpha_k)\,,$$

so that for any compact set $K \subset \mathbb{R}^d$

$$\sup_{x\in K}|Q_l(x,x) - Q_\infty(x,x)| \sim \sum_{k\geq l}\alpha_k\,.$$

Moreover, since $C_\infty(x,x') \geq C_l(x,x')$ and $Q_\infty(x,x) \geq Q_l(x,x)$ for all $x \in \mathbb{R}^d$, we can use the fact that

$$\begin{aligned}Q_\infty(x,x') - Q_l(x,x') &= \sqrt{Q_\infty(x,x)Q_\infty(x',x')}(C_\infty(x,x') - C_l(x,x')) \\ &+ C_l(x,x')(\sqrt{Q_\infty(x,x)Q_\infty(x',x')} - \sqrt{Q_l(x,x)Q_l(x',x')})\end{aligned}$$

and hence conclude. □

### A3.2 Proof of Corollary 1

**Corollary 1.** *The following statements hold*
- *Let $K$ be a compact set of $\mathbb{R}^d$ and assume $\sigma_b > 0$. Then, $Q_\infty$ is universal on $K$.*
- *Assume $\sigma_b = 0$. Then $Q_\infty$ is universal on $\mathbb{S}^{d-1}$.*

*Proof.* Corollary 1 is a direct result of Propositions 3, 4 and 2. Indeed, for any compact $K \subset \mathbb{R}^d$, $\mathcal{H}_{Q_L}(K) \subset \mathcal{H}_{Q_\infty}(K)$ for all $L \geq 0$. Therefore, the universality of $Q_L$ for some finite $L$ is sufficient to conclude that $Q_\infty$ is universal. □

## A4 Neural Tangent Kernel

Throughout this section, we will consider ResNets with NTK parameterization [Jacot et al., 2018]. This simply means that all the components of the biases and the weights will be initialized as iid standard normal random variables. In order to compensate this change of parameterization, the propagation through the network needs to be slightly modified. Hence (2) will be replaced by

$$\begin{aligned} y_0(x) &= \tfrac{\sigma_w}{\sqrt{d}} W_0\, x + \sigma_b\, B_0\,; \\ y_l(x) &= y_{l-1}(x) + \lambda_{l,L}\, \tfrac{\sigma_w}{\sqrt{N_{l-1}}} W_l + \sigma_b\, B_l\,. \end{aligned} \tag{A12}$$

However, it is strightforward to verify that the recurrence (5) for the covariance kernels keeps unchanged. Clearly, the dynamics of a standard ResNet with NTK parameterization can be recovered from (A12) by setting $\lambda_{l+1,L} = 1$ for all $l, L$.

The Neural Tangent Kernel, introduced by [Jacot et al., 2018], is defined as

$$\tilde{\Theta}_L^{ij}(x, x') = \nabla_{\text{par}}\, y_L^i(x) \cdot \nabla_{\text{par}}\, y_L^j(x')\,,$$

where $\nabla_{\text{par}}$ denotes the gradient wrt the parameters of the network.
The NTK of a Stable ResNet can be evaluated recursively. We will now prove the recurrence formula (9). The following result was proven in Lemma 3 in [Hayou et al., 2019b] for the case of a standard ResNet without bias. We extend it to ResNet with bias.

**Lemma A16** (Recurrence relatino for the NTK)**.** *For a Stable ResNet, the NTK can be evaluated recursively, layer by layer, as*

$$\Theta_0 = Q_0\,; \qquad \Theta_{l+1} = \Theta_l + \lambda_{l+1,L}^2\,(\Psi_l + \Psi_l'\,\Theta_l)\,, \tag{9}$$

*where* $\Psi_l(x, x') = \sigma_b^2 + \sigma_w^2 \mathbb{E}[\phi(y_1^l(x))\phi(y_1^l(x'))]$ *and* $\Psi_l'(x, x') = \sigma_w^2 \mathbb{E}[\phi'(y_1^l(x))\phi'(y_1^l(x'))]$.

*Proof.* The first result is the same as in the FFNN case [Jacot et al., 2018], since we assume there is no residual connections between the first layer and the input. Let $x, x' \in \mathbb{R}^d$. We have

$$\Theta_0(x, x') = \sum_{j=0}^{d} \frac{\partial y_0^1(x)}{\partial w_0^{1j}} \frac{\partial y_0^1(x)}{\partial w_0^{1j}} + \frac{\partial y_0^1(x)}{\partial b_0^1} \frac{\partial y_0^1(x)}{\partial b_0^1} = \frac{\sigma_w^2}{d} x \cdot x' + \sigma_b^2\,.$$

We prove the second result by induction. The proof is similar to the one of ResNet in [Hayou et al., 2019b]. Let $\theta_k = (W_k, B_k)$. For $l \geq 1$ and $i \in [1 : N_{l+1}]$

$$\partial_{\theta_{0:l}} y_{l+1}^i(x) = \partial_{\theta_{0:l}} y_l^i(x) + \lambda_{l+1,L} \frac{\sigma_w}{\sqrt{N_l}} \sum_{j=1}^{N_l} W_{l+1}^{ij} \phi'(y_l^j(x)) \partial_{\theta_{1:l}} y_l^j(x)\,.$$

Therefore, we obtain

$$\begin{aligned} (\partial_{\theta_{0:l}} y_{l+1}^i(x))(\partial_{\theta_{0:l}} y_{l+1}^i(x'))^t &= (\partial_{\theta_{0:l}} y_l^i(x))(\partial_{\theta_{0:l}} y_l^i(x'))^t \\ &\quad + \lambda_{l+1,L}^2 \frac{\sigma_w^2}{N_l} \sum_{j,j'}^{N_l} W_{l+1}^{ij} W_{l+1}^{ij'} \phi'(y_l^j(x))\phi'(y_l^{j'}(x')) \partial_{\theta_{0:l}} y_l^j(x) (\partial_{\theta_{0:l}} y_l^{j'}(x'))^t + I\,, \end{aligned}$$

where

$$I = \lambda_{l+1,L} \frac{\sigma_w}{\sqrt{N_l}} \sum_{j=1}^{N_l} W_{l+1}^{ij} (\phi'(y_l^j(x)) \partial_{\theta_{0:l}} y_l^i(x) (\partial_{\theta_{0:l}} y_l^j(x'))^t + \phi'(y_l^j(x')) \partial_{\theta_{0:l}} y_l^j(x) (\partial_{\theta_{0:l}} y_l^i(x'))^t)\,.$$

We prove the result by induction. Assume the result is true for layers $1, 2, ..., l$ and let us prove it for $l+1$. Using the induction hypothesis, as $N_1, N_2, ..., N_{l-1} \to \infty$ recursively, we have that

$$(\partial_{\theta_{0:l}} y_{l+1}^i(x))(\partial_{\theta_{0:l}} y_{l+1}^i(x'))^t + \lambda_{l+1,L}^2 \frac{\sigma_w^2}{N_l} \sum_{j,j'}^{N_l} W_{l+1}^{ij} W_{l+1}^{ij'} \phi'(y_l^j(x))\phi'(y_l^{j'}(x')) \partial_{\theta_{0:l}} y_l^j(x) (\partial_{\theta_{0:l}} y_l^{j'}(x'))^t + I$$

$$\to \Theta_l(x, x') + \lambda_{l+1,L}^2 \frac{\sigma_w^2}{N_l} \sum_{j}^{N_l} (W_{l+1}^{ij})^2 \phi'(y_l^j(x))\phi'(y_l^j(x')) \Theta_l(x, x') + I'\,,$$

where $I' = \frac{\sigma_w^2}{N_l} W_{l+1}^{ii}(\phi'(y_l^i(x)) + \phi'(y_l^i(x')))\,\Theta_l(x,x')$.

As $N_l \to \infty$, we have that $I' \to 0$. Using the law of large numbers, as $N_l \to \infty$

$$\frac{\sigma_w^2}{N_l}\sum_j^{N_l}(W_{l+1}^{ij})^2\phi'(y_l^j(x))\phi'(y_l^j(x'))\Theta_l(x,x') \to \Psi_l'\Theta_l(x,x').$$

Moreover, we have that

$$\begin{aligned}&(\partial_{W_{l+1}} y_{l+1}^i(x))(\partial_{W_{l+1}} y_{l+1}^i(x'))^t + (\partial_{B_{l+1}} y_{l+1}^i(x))(\partial_{B_{l+1}} y_{l+1}^i(x'))^t \\ &= \frac{\sigma_w^2}{N_l}\sum_j \phi(y_l^j(x))\phi(y_l^j(x')) + \sigma_b^2 \underset{N_l\to\infty}{\to} \sigma_w^2\mathbb{E}[\phi(y_l^1(x))\phi(y_l^1(x'))] + \sigma_b^2 = \Psi_l\,,\end{aligned}$$

and so we conclude. □

As a corollary of the above result, using the results in [Daniely et al., 2016] for the ReLU activation function, we can express the recursion more explicitly. We have

$$\Psi_l = \sigma_b^2 + \tfrac{\sigma_w^2}{2}\left(1 + \tfrac{f(C_l)}{C_l}\right)Q_l\,; \qquad \Psi_l' = \tfrac{\sigma_w^2}{2}(1 + f'(C_l))\,,$$

where $f$ is defined in (4) and $f' : \gamma \mapsto -\frac{1}{\pi}\arccos\gamma$ is the first derivative of $f$. So we can write

$$\Theta_{l+1} = \Theta_l + \lambda_{l+1,L}^2\left(\sigma_b^2 + \tfrac{\sigma_w^2}{2}\left(1 + \tfrac{f(C_l)}{C_l}\right)Q_l + \tfrac{\sigma_w^2}{2}(1 + f'(C_l))\,\Theta_l\right). \tag{A13}$$

We can now easily check that the NTK is a kernel in the sense of Definition 2.

**Lemma A17** ($\Theta_l$ is a kernel)**.** *For all layer $l$, $\Theta_L$ is a kernel in the sense of definition (2).*

*Proof.* It's clear that $\Theta_0 = Q_0$ is a kernel. Now fix any layer $l$. We have already proved in Lemma A10 that $\left(1 + \frac{f(C_l)}{C_l}\right)Q_l$ is a kernel. With a similar argument, noting that $1 + f'$ can be expressed as a power series with only non negative coefficients on $[-1,1]$, we conclude by Lemma A9 that $1 + f'(C_l)$ is a kernel. Using the usual argument that sums and product of kernels are kernels, we conclude by induction that $\Theta_l$ is a kernel. □

As a final remark, note that from (A1), we have that $\lambda_{l,L}^2\Psi_l = Q_{l+1} - Q_l$. Hence we can rewrite (A13) as

$$\Theta_{l+1} - \Theta_l = Q_{l+1} - Q_l + \lambda_{l+1,L}^2\tfrac{\sigma_w^2}{2}(1 + f'(C_l))\,\Theta_l\,. \tag{A14}$$

Since $1 + f'$ is non negative on $[-1,1]$, it is easy to show by induction that $\Theta_l \geq Q_l$, point-wise, for all $l$. This is done explicitly in the next Lemma, which is a Corollary of Lemma 1 and show the divergence of the NTK for a Standard ResNet.

**Lemma A18** (Exploding NTK)**.** *Consider a ResNet of form (1). For all $x \in \mathbb{R}^d$,*

$$\Theta_L(x,x) \geq \left(1 + \tfrac{\sigma_w^2}{2}\right)^L\left(Q_0(x,x) + \tfrac{2\sigma_b^2}{\sigma_w^2}\right). \tag{A15}$$

*Proof.* By Lemma 1, it suffices to show that $\Theta_L(x,x) \geq Q_L(x,x)$.
Recall (A13), noticing that $1 + f' \geq 0$ on $[-1,1]$, $\left(1 + \frac{f(C_l)}{C_l}\right)Q_l \geq 0$ and that $\Theta_0 = Q_0 \geq 0$, by an easy induction we have that $\Theta_l \geq 0$ for all $l$. As a consequence, from (A14), we get that $\Theta_{l+1} - \Theta_l \geq Q_{l+1} - Q_l$. Hence, again with a straightforward induction we have that $\Theta_l \geq Q_l$ for all $l$ and the the whole $K^2$. In particular $\Theta_L(x,x) \geq Q_L(x,x)$ for all $x \in K$. □

**Lemma A19** (Normalized NTK recursion)**.** *Consider a ResNet of type (1) without bias, and let $\alpha = \frac{\sigma_w^2}{2}$. The NTK recursion formula can be written in terms of normalized NTK $\kappa^l(x,x') = \Theta_l(x,x')/(1+\alpha)^{l-1}$*

$$\kappa_l(x,x') = \left(\frac{1 + \alpha\hat{f}'(C_{l-1}(x,x'))}{1+\alpha}\right)\kappa_{l-1}(x,x') + \alpha\hat{f}(C_{l-1}(x,x'))\sqrt{Q_0(x,x)Q_0(x',x')}\,,$$

*where $\hat{f}$ is given by (A2), $\hat{f}(t) = \frac{1}{\pi}(t\arcsin t + \sqrt{1-t^2}) + \frac{1}{2}t$.*

*Proof.* Let $x, x' \in \mathbb{R}^d$. For a ResNet of type (1), we have that

$$\Theta_l = \Theta_{l-1} + \left(\Psi_{l-1} + \Psi'_{l-1}\,\Theta_{l-1}\right)\,,$$

where $\Psi_{l-1} = \alpha Q_{l-1}(x,x')$ and $\Psi'_{l-1} = \alpha \hat{f}'(C_{l-1})$. Using the recursive formula for the diagonal elements, we have that $\Psi_{l-1} = \alpha(1+\alpha)^{l-1}\hat{f}(C_{l-1}(x,x'))\sqrt{Q_0(x,x)Q_0(x',x')}$. We conclude by dividing both sides by $(1+\alpha)^{l-1}$. □

### A4.1 Proof of Proposition 8

**Proposition 8.** *Fix a compact $K \subset \mathbb{R}^d$ ($0 \notin K$ if $\sigma_b = 0$) and consider a Stable ResNet with decreasing scaling. Then $\Theta_L$ converges uniformly over $K^2$ to a kernel $\Theta_\infty$. Moreover $\Theta_\infty$ is universal on $K$ if $\sigma_b > 0$. If $K = \mathbb{S}^{d-1}$, then the universality holds for $\sigma_b = 0$.*

*Proof.* Let $K \subset \mathbb{R}^d$ ($0 \notin K$ if $\sigma_b = 0$) be a compact. From (A13), with a decreasing scaling, we have that

$$\begin{aligned}\Theta_l &= \Theta_{l-1} + \lambda_l^2\left(\Psi_{l-1} + \Psi'_l\,\Theta_{l-1}\right)\\ &= \left(1 + \lambda_l^2\frac{\sigma_w^2}{2}f'(C_{l-1})\right)\Theta_{l-1} + \lambda_l^2\Psi_{l-1}\,.\end{aligned}$$

Therefore, the NTK can be expressed exclusively in terms of the covariance kernels $(Q_k)_{k\in[0:l-1]}$, more precisely we have that

$$\Theta_l = \prod_{k=1}^{l}\left(1 + \lambda_k^2\frac{\sigma_w^2}{2}f'(C_{k-1})\right)Q_0 + \sum_{k=1}^{l}\lambda_l^2\prod_{j=k}^{l}\left(1 + \lambda_j^2\frac{\sigma_w^2}{2}f'(C_{j-1})\right)\Psi_{k-1}\,.$$

It is straightforward that $\Theta_l$ converges pointwise to a limiting kernel $\Theta_\infty$. Let us prove that the convergence is uniform over $K$. By observing that $|f'| \le 1$, we have that for all $x, x' \in K$

$$\begin{aligned}|\Theta_\infty(x,x') - \Theta_l(x,x')| &\le \prod_{k=1}^{l}\left(1 + \lambda_k^2\frac{\sigma_w^2}{2}\right)\left|\prod_{k=l+1}^{\infty}\left(1 + \lambda_k^2\frac{\sigma_w^2}{2}\right) - 1\right|Q_0(x,x')\\ &\quad + \sum_{k=l+1}^{\infty}\lambda_k^2\prod_{j=k}^{l}\left(1 + \lambda_j^2\frac{\sigma_w^2}{2}\right)\Psi_{k-1}(x,x')\\ &\le \kappa\sum_{k=l+1}^{\infty}\lambda_k^2\,.\end{aligned}$$

where $\kappa$ is a constant that depends on the compact $K$. This proves the uniform convergence with a rate of $\mathcal{O}\left(\sum_{k=l+1}^{\infty}\lambda_k^2\right)$. As a consequence, being a uniform limit of kernels, $\Theta_\infty$ is a kernel.

Proceeding as in the proof of Lemma A17, it's easy to prove by induction that for all $l$, $\Theta_l - Q_l$ is a kernel. In particular,

$$T(\Theta_l) \succeq T(Q_l)\,,$$

where $\succeq$ is in the operator sense, that is $T(\Theta_l) - T(Q_l)$ is non-negative definite. This yields

$$T(\Theta_\infty) \succeq T(Q_\infty)\,.$$

Therefore $\Theta_\infty$ inherits the universality of $Q_\infty$ naturally by the RKHS hierarchy [Paulsen and Raghupathi, 2016]. We conclude that $\Theta_\infty$ is universal (for both cases). □

For the rest of this section, let $K \subset \mathbb{R}^d$ by a compact set. If $\sigma_b = 0$, assume that $0 \notin K$.

With the uniform scaling, for arbitrary $x, x' \in K$, the continuous version of (9) reads

$$\begin{aligned}&\dot{\theta}_t(x,x') = \dot{q}_t(x,x') + \tfrac{\sigma_w^2}{2}\left(1 + f'(c_t(x,x'))\right)\theta_t(x,x')\,;\\ &\theta_0 = q_0\,,\end{aligned} \tag{A16}$$

where $f' : \gamma \mapsto -\frac{1}{\pi}\arccos\gamma$ is the first derivative of $f$, defined in (4).

**Lemma A20.** *For any $x, x'$ in $K$, the solution $t \mapsto \Theta_t$ of (A16) is unique and well defined for all $t \in [0,1]$. Moreover, the map $(x, x') \mapsto \Theta_t(x, x')$ is a kernel in the sense of Definition 2 for all $t \in [0,1]$.*
*We have the $L^2(K)$ convergence of the discrete model to the continuous one:*

$$\lim_{L\to\infty} \sup_{l\in[0:L]} \left\| T(\Theta_{l|L}) - T(\theta_{t=l/L}) \right\|_2 = 0\,.$$

*Proof.* The existence and the uniqueness are clear, since it is a homogeneous first order Cauchy problem, with continuous coefficients. We can write explicitly the solution as

$$\theta_t = e^{G_t} \left( q_0 + \int_0^t \dot{q}_s\, e^{-G_s}\, \mathrm{d}s \right)\,, \tag{A17}$$

where $G_t(z) = \frac{\sigma_w^2}{2} \int_0^t (1 + f'(c_s(z)))\, \mathrm{d}s$ for $z \in K^2$. It becomes then clear that $z \mapsto \Theta_t(z)$ is a continuous and symmetric function on $K^2$.
it is easy to check that the uniform convergence of $C$ and $Q$ to $c$ and $q$ implies that for all $z \in K$, $\lim_{L\to\infty} \sup_{l\in[0:L]} |\Theta_{l|L}(z) - \theta_{l/L}(z)| = 0$. As consequence, by dominated convergence,

$$\lim_{L\to\infty} \sup_{l\in[0:L]} \left\| T(\Theta_{l|L}) - T(\theta_{t=l/L}) \right\|_2 = 0\,.$$

Hence, $T(\theta_t)$ is the limit of a sequence of non-negative definite operators and hence it is non-negative definite, so that $\theta_t$ is a kernel on $K$ for all $t \in [0,1]$. □

**Proposition 9.** *Let $K \subset \mathbb{R}^d$ and fix $t \in (0,1]$. If $\sigma_b > 0$, then $\theta_t$ is universal on $K$. The same holds true if $\sigma_b = 0$ and $K = \mathbb{S}^{d-1}$.*

*Proof.* Fix $t \in (0,1]$. The solution of (A16) can be written as $\theta_t = q_t + r_t$, where

$$r_t = \tfrac{\sigma_w^2}{2} \int_0^t (1 + f'(c_s))\, \theta_s\, \mathrm{d}s\,.$$

Now, let us show that $r_t$. First, by Lemma A14 it is easy to check that $1 + f'$ is analytic on $(-1,1)$ and its Taylor expansion around 0 converges on $[-1,1]$. Moreover all the Taylor coefficients are non negative. Hence, Lemma A9 shows that $(1 + f'(c_s))$ is a kernel for all $s \in [0,s)$. It follows that $(1 + f'(c_s))\, \theta_s$ is a kernel.[17] Now, $(1 + f'(c_s)\, \theta_s$ is continuous and symmetric on $Z^2$, and it is easy to check from (A17) that it is uniformly bounded for $s \in [0,t)$. It follows that $r_t$ is continuous and symmetric. Now, fix an arbitrary finite Borel measure $\mu$ on $K$. We have to show that $T_\mu(r_t)$ is non-negative definite, so that we can conclude by Lemma A1. Fixed $\varphi \in L^2(K,\mu)$, by simple standard arguments we have

$$\langle T_\mu(r_t)\, \varphi, \varphi \rangle = \int_0^t \langle T_\mu((1 + f'(c_s))\, \theta_s)\, \varphi, \varphi \rangle\, \mathrm{d}s \geq 0$$

and so $r_t$ is a kernel.
Now, given two kernels $Q$ and $R$, it is a classical result that $Q + R$ is a kernel and its RKHS contains the RKHS of $Q$ and $R$, [Paulsen and Raghupathi, 2016]. We conclude that the RKHS of $\theta_t$ contains the RKHS of $q_t$. Since $q_t$ is universal, $\theta_t$ is universal. □

## A5 A PAC-Bayes Generalization result

In this section, we study the PAC-Bayes upper bound of a GP with kernel $Q_L$. We consider a dataset $S$ with $N$ iid training examples $\{(x_i, y_i) \in X \times Y, i \in [1:N]\}$, and a hypothesis space $\mathcal{H}$ from which we want to learn an optimal hypothesis according to some bounded loss function $\ell : Y \times Y \to [0,1]$. The empirical loss of a hypothesis $h \in \mathcal{H}$ is given by

$$r_S(h) = \frac{1}{N} \sum_{i=1}^{N} \ell(h(x_i), y_i)\,.$$

[17] See footnote 14.

Assuming that the samples are distributed as $(x, y) \sim \nu$ where $\nu$ is a probability distribution on $X \times Y$, we define the generalization (true) loss by

$$r(h) = \mathbb{E}_\nu[\ell(f(x), y)]\,.$$

For some randomized learning algorithm $\mathcal{A}$, the empirical and generalization loss are given by

$$r_S(\mathcal{A}) = \mathcal{E}_{h\sim\mathcal{A}}[r_s(h)]\,; \qquad r(\mathcal{A}) = \mathcal{E}_{h\sim\mathcal{A}}[r(h)]\,.$$

The PAC-Bayes theorem gives a probabilistic upper bound on the generalization loss $r(\mathcal{A})$ of a randomized learning algorithm $\mathcal{A}$ in terms of the empirical loss $r_S(\mathcal{A})$. Fix a prior distribution $\mathcal{P}$ on the hypothesis set $\mathcal{H}$. The Kullback-Leibler divergence between $\mathcal{A}$ and $\mathcal{P}$ is defined as $KL(\mathcal{A}\|\mathcal{P}) = \int \log \frac{\mathcal{A}(h)}{P(h)} \mathcal{A}(h)dh \in [0, \infty]$. The Bernoulli KL-divergence is given by $kl(a||p) = a \log \frac{a}{p} + (1-a) \log \frac{1-a}{1-p}$ for $a, p \in [0, 1]$. We define the inverse Bernoulli KL-divergence $kl^{-1}$ by

$$kl^{-1}(a, \varepsilon) = \sup\{p \in [0, 1] : kl(a, p) \leq \varepsilon\}\,.$$

**Theorem 2** (PAC-Bayesian theorem)**.** *For any loss function $\ell$ that is $[0, 1]$ valued, for any distribution $\nu$, for any $N \in \mathbb{N}$, for any prior $P$, and any $\delta \in (0, 1]$, with probability at least $1 - \delta$ over the sample $S$, we have*

$$\forall \mathcal{A}, \quad r(\mathcal{A}) \leq kl^{-1}\left(r_S(\mathcal{A}), \frac{KL(\mathcal{A}\|P) + \log \frac{2\sqrt{N}}{\delta}}{N}\right).$$

The PAC-Bayesian theorem gives can also be stated as

$$kl(r_S(\mathcal{A}), r(\mathcal{A})) \leq \frac{KL(\mathcal{A}\|P) + \log \frac{2\sqrt{N}}{\delta}}{N}\,.$$

The KL-divergence term $KL(\mathcal{A}\|P)$ plays a major role as it controls the generalization gap, i.e. the difference (in terms of Bernoulli KL-divergence) between the empirical loss and the generalization loss. In our setting, we consider an ordinary GP regression with prior $P(f) = \mathcal{GP}(f|0, Q(x, x'))$. Under the standard assumption that the outputs $y_N = (y_i)_{i\in[1:N]}$ are noisy versions of $f_N = (f(x_i))_{i\in[1:N]}$ with $y_N|f_N \sim \mathcal{N}(y_N|f_N, \sigma^2 I)$, the Bayesian posterior $\mathcal{A}$ is also a GP and is given by

$$\mathcal{A}(f) = \mathcal{GP}(f|Q_N(x)(Q_{NN} + \sigma^2 I)^{-1} y_N, Q(x, x') - Q_N(x)(Q_{NN} + \sigma^2 I)^{-1})Q_N(x')^T)\,, \tag{A18}$$

where $Q_N(x) = (Q(x, x_i))_{i\in[1:N]}$ and $Q_{NN} = (Q(x_i, x_j))_{1\leq i,j\leq N}$. In this setting, we have the following result

**Proposition 10** (Stability of PAC-Bayes bound)**.** *Let $Q_L$ be the kernel of a ResNet. Let $P_L$ be a GP with kernel $Q_L$ and $\mathcal{A}_L$ be the corresponding Bayesian posterior for some fixed noise level $\sigma > 0$. Then, in a fixed setting (fixed sample size N), the following results hold:*

1. *With a standard ResNet, we have*
$$KL(\mathcal{A}_L\|P_L) \gtrsim L\,.$$
2. *With a Stable ResNet, we have*
$$KL(\mathcal{A}_L\|P_L) = \mathcal{O}_L(1)\,.$$

*Proof.* The proof relies on the simple observation that $P_L(f|f_N) = \mathcal{A}_L(f|f_N)$. This yields

$$\begin{aligned} KL(\mathcal{A}_L\|P_L) &= KL(\mathcal{A}_L(f_N)\mathcal{A}_L(f|f_N)\|P_L(f_N)P_L(f|f_N)) \\ &= KL(\mathcal{A}_L(f_N)\|P_L(f_N)) \\ &= \frac{1}{2}\log(\det(Q_{L,NN} + \sigma^2 I)) - \frac{N}{2}\log(\sigma^2) - \frac{1}{2}\operatorname{Tr}(Q_{L,NN}(Q_{L,NN} + \sigma^2 I)^{-1}) \\ &+ \frac{1}{2} y_N^T (Q_{L,NN} + \sigma^2 I)^{-1} Q_{L,NN} (Q_{L,NN} + \sigma^2 I)^{-1} y_N\,, \end{aligned} \tag{A19}$$

where $Q_{L,NN} = (Q_L(x_i, x_j))_{1\leq i,j\leq N}$.

Since $Q_{L,NN}$ is symmetric and strictly positive definite, it is straightforward that the largest eigenvalue of $Q_{L,NN}(Q_{L,NN}+\sigma^2 I)^{-1})$ is smaller than 1. This yields

$$\mathrm{Tr}(Q_{L,NN}(Q_{L,NN}+\sigma^2 I)^{-1}) \leq N$$

and

$$y_N^T(Q_{L,NN}+\sigma^2 I)^{-1}Q_{L,NN}(Q_{L,NN}+\sigma^2 I)^{-1}y_N \leq \sigma^{-2}\|y_N\|_2\,.$$

Both quantities are bounded independently from $L$ and the scaling factors $(\lambda_{k,L})_{k\in[2:L]}$.
Now let us analyse the first term $\frac{1}{2}\log(\det(Q_{L,NN}+\sigma^2 I))$. Let $\mu_{L,0}\geq\mu_{L,1}\geq\cdots\geq\mu_{L,N}$ be the eigenvalues of $Q_{L,NN}$. For a simplification purpose, we assume the inputs belong to the unit sphere $\mathbb{S}^{d-1}$. The proof extends to any compact set.
Let us study the behaviour of the first term for both cases.

*Case 1.* Assume we have a standard ResNet architecture. On the unit sphere $\mathbb{S}^{d-1}$, we have that $Q_L(x,x')\geq q_L C_L(x,x')$, where $q_L=(1+\frac{\sigma_2}{2})^L\delta$ with $\delta=(\sigma_b^2+\frac{\sigma_w^2}{d})/(1+\frac{\sigma_w^2}{2})$. Using Lemma 4, we know that $\lim_{L\to\infty}\hat{\mu}_{L,0}=\hat{\mu}_{\infty,0}\in(0,\infty)$ and for all $k\geq 1$, $\lim_{L\to\infty}\hat{\mu}_{L,k}=0$. This yields

$$\begin{aligned}\log(\det(Q_{L,NN}+\sigma^2 I)) &\geq \sum_{k=1}^{N}\log(q_L\hat{\mu}_{L,k}+\sigma^2)\\ &\geq \log(q_L\hat{\mu}_{L,0}+\sigma^2)+(N-1)\log(\sigma^2)\\ &\gtrsim L\log(1+\frac{\sigma_w^2}{2})\,,\end{aligned}$$

where the last inequality holds for sufficiently large $L$.

*Case 2.* In the case of Stable ResNet, we know that as $L\to\infty$, the kernel $Q_L$ converges to a strictly positive definite kernel $Q_\infty$, therefore the first term $\log(\det(Q_{L,NN}+\sigma^2 I))$ remains bounded as $L\to\infty$, which concludes the proof. □

## A6 NNGP correlation kernel without bias as a modified NNGP kernel

Unscaled ResNets suffer from the exploding variance problem, which needs to be avoided in order to isolate the disadvantages of inexpressivity in their NNGP kernel. In order to do so, we use the NNGP correlation kernel $C$ instead of NNGP covariance kernel $Q$, noting that Lemma A4 provides a simple recursion formula for $C$ if $\sigma_b=0$, at depth $l\leq L$:

$$C_l(x,x')=\frac{1}{1+\alpha_{l.L}}C_{l-1}(x,x')+\frac{\alpha_{l,L}}{1+\alpha_{l,L}}\hat{f}(C_{l-1}(x,x'))\,, \tag{A20}$$

where $\alpha_{l,L}=\frac{\lambda_{l,L}^2\sigma_w^2}{2}$ and $\hat{f}$ defined in (A2). In order to combine this with open-source packages [Novak et al., 2020, Bradbury et al., 2018] designed for NNGP calculation, we note that (A20) can be viewed as the NNGP kernel of the following modified ResNet layer, using the same notation as in (2):

$$y_l(x)=\sqrt{1-\hat{\alpha}_{l,L}}y_{l-1}(x)+\sqrt{\hat{\alpha}_{l,L}}\,\mathcal{F}((W_l,B_l),y_{l-1})\,,\quad l\in[1:L]\,, \tag{A21}$$

with $\hat{\alpha}_{l,L}=\frac{\alpha_{l,L}}{1+\alpha_{l,L}}$

## A7 Experimental details and additional results

### A7.1 NNGP results

For our Vanilla ResNet NNGP results, we preprocess all training, validation and test data by first centering the training set and then normalizing all images to lie on the pixel dimension sphere. For our Wide ResNet NNGP results we normalise all data so that the training set is centered and has channel-wise unit variance. We use Kaiming [He et al., 2015] initialisation throughout, with $\sigma_w^2=2$ and $\sigma_b^2=0$. Vanilla ResNets have the same

structure as type (2) in Table 2 and we use the same WRN kernel architecture as [Lee et al., 2019] in Table 1 but omit the final average pooling step, which is known to improve kernel performance but dramatically increase computational costs [Novak et al., 2019, Lee et al., 2020]. Throughout this work, where there are residual blocks with multiple layers, we calculate our scaling factors for uniform and decreasing scaled Stable ResNets by the number of residual connections. For example, a WRN-202 has only 99 residual connections, so we set $\lambda_{l,L}^{-1} = \sqrt{99}$ for the uniform scaling factors. We tune the noise variance $\sigma^2$, which is akin to the regularisation parameter in kernel ridge regression. To do so, we compute validation accuracy on a validation set of size 5000, selecting the best $\sigma^2 = \lambda \times \text{Trace}(Q_{NN})/N$ from a logarithmic scale of $\lambda = [0.001, 0.01, 0.1]$, where $N$ is the training set size and $Q_{NN}$ is the $N \times N$ training set Gram matrix for NNGP $Q$.

### A7.2 Trained ResNet results

For all our trained ResNet experiments we use a similar setup to the open-source code for [Wang et al., 2020] in PyTorch [Paszke et al., 2019]. We repeat each experiment 3 times and report the best test accuracy and error intervals. All ResNets are initialised with Kaiming initialisation [He et al., 2015] and like [Wang et al., 2020] we adopt ResNets architectures where we double the number of filters in each convolutional layer. For experiments with BatchNorm, on CIFAR-10/100 we use batch size 64 across all depths and on TinyImageNet we used batch size 128 for depths 32 & 50, and batch size 100 for depth 104 in order to allow the model to fit onto a single 11GB VRAM GPU. We use SGD with momentum parameter 0.9 and weight decay parameter $10^{-4}$ throughout.

We also present results for ResNets trained without BatchNorm [Ioffe and Szegedy, 2015]. BatchNorm is a normalization layer commonly used with modern ResNets that is known to improve performance and allows deeper ResNets to be trained, though the precise reasons for this are not well understood. Several recent works [De and Smith, 2020, Zhang et al., 2019] have studied the possibility of removing the need for BatchNorm layers, by introducing trainable uniform scalings to the residual connection to stabilise variance at initialisation & gradients, demonstrating promising results. Note, our work additionally introduces decreasing scaling and also uses the infinite-width NNGP/NTK connection to assess the theoretical advantages of scaled Stable ResNets in the limit of infinite depth.

Moreover, our focus is not towards the possibility of removing BatchNorm and we show in Table 3 that our scalings can improve BatchNorm ResNets. However, we also present results without BatchNorm in Table 4, where again we see that our scaled stable ResNets improve performance compared to their unscaled counterparts: for example both Decreasing and Uniform scaling outperform the unscaled ResNet by over 3% test accuracy on CIFAR-100 with ResNet-104.

For ResNets trained without BatchNorm, for a fair comparison we tuned the initial learning rate on a small logarithmic scale, using batch size 128.

**Table 4:** Test accurracies (%) of trained deep ResNets **without BatchNorm** of various scalings and depths on CIFAR-10 (C-10), CIFAR-100 (C-100).

| Dataset | Depth | Scaled (D) | Scaled (U) | Unscaled |
|---|---|---|---|---|
| C-10 | 32 | $92.64_{\pm 0.19}$ | $92.78_{\pm 0.18}$ | $92.11_{\pm 0.17}$ |
| | 50 | $92.33_{\pm 0.05}$ | $\mathbf{92.72}_{\pm 0.12}$ | $92.10_{\pm 0.17}$ |
| | 104 | $92.81_{\pm 0.09}$ | $\mathbf{93.28}_{\pm 0.17}$ | $92.70_{\pm 0.08}$ |
| C-100 | ResNet32 | $\mathbf{67.73}_{\pm 0.42}$ | $67.06_{\pm 0.38}$ | $65.37_{\pm 0.32}$ |
| | ResNet50 | $\mathbf{69.38}_{\pm 0.20}$ | $68.76_{\pm 0.18}$ | $66.02_{\pm 0.41}$ |
| | ResNet104 | $70.60_{\pm 0.52}$ | $\mathbf{70.95}_{\pm 0.13}$ | $67.41_{\pm 0.41}$ |

## A8 Some results on the Sphere $\mathbb{S}^{d-1}$

On the sphere $\mathbb{S}^{d-1}$, the kernel $Q_2$ is analytic as a result of lemma A8. Moreover, the coefficient of the analytic decomposition are all positive.

**Lemma A21** (Analytic decomposition of 2 layer ReLU ResNet)**.** *For all* $(x, x') \in \mathbb{S}^{d-1}$, $Q_2(x, x') = g(x \cdot x')$ *where* $g(z) = \sum_{i \geq 0} a_i z^i$ *and* $a_i > 0$ *for all* $i \geq 0$.

*Proof.* Let $x, x' \in \mathbb{S}^{d-1}$. We have

$$Q_0(x, x') = \sigma_b^2 + \frac{\sigma_w^2}{d} x \cdot x' \,.$$

As a result, for all $x, x'$, $Q_0(x, x) = Q_0(x', x') = \sigma_b^2 + \frac{\sigma_w^2}{d}$. The diagonal term of the kernel is the same for all $x \in \mathbb{S}^{d-1}$. We note $\beta_l = Q_l(x, x)$ and $z = x \cdot x'$. Using this observation, we have that

$$Q_1(x, x') = Q_0(x, x') + \lambda_1^2(\sigma_b^2 + \frac{\sigma_2}{2} \hat{f}(C_0(x, x'))\beta_0) \,.$$

It can be easily deduced from lemma A8 that there exist $\{b_i\}_{i \geq 0}$ such that

$$C_1(x, x') = b_0 + b_1 z + \sum_{i \geq 0} b_{2i} z^{2i} \,,$$

where $b_0, b_1, b_{2i} > 0$.
Following the same approach, we have that

$$Q_2(x, x') = Q_1(x, x') + \lambda_2(\sigma_b^2 + \frac{\sigma_w^2}{2} f(C_1(x, x'))\beta_2)$$

and

$$\hat{f}(C_1(x, x')) = a_0 + a_1 C_1(x, x') + \sum_{i \geq 1} a_{2i} (C_1(x, x'))^{2i} \,.$$

Having the terms of orders 0 and 1 in $C_1(x, x')$ ensures having a positive coefficient for all terms $z^i$ for $i \geq 1$, which concludes the proof. □

The previous result can be easily extended to general $L \geq 2$. We have that

$$Q_L(x, x') = g_L(x \cdot x') \,,$$

where $g_L : [-1, 1] \to \mathbb{R}$ is a continuous function. Kernels that can be written in this form are known as the dot-product kernels (or zonal kernels on the unit sphere). In our setting, we have a stronger property; we prove in the next result we show a that the kernel $Q_L$ is analytic on the sphere $\mathbb{S}^{d-1}$ in the sense that the function $g_L$ is analytic on $[-1, 1]$.

**Proposition A5** ($Q_L$ is analytic)**.** *Let $L \geq 2$, there exists $(\alpha_{L,i})_{i \geq 0}$ such that for all $x, x' \in \mathbb{S}^{d-1}$*

$$Q_L(x, x') = \sum_{i \geq 0} \alpha_{L,i} (x \cdot x')^i \,.$$

*Moreover, $(\alpha_{L+1,i})_{i \geq 0}$ can be expressed in terms of $(\alpha_{L,i})_{i \geq 0}$*

$$\alpha_{L+1,i} = \alpha_{L,i} + \lambda_{L+1,L+1} \times \gamma_{L,i} \,, \tag{A22}$$

*with*

$$\gamma_{L,i} = \begin{cases} \sigma_b^2 + \beta_L \frac{\sigma_w^2}{2} \sum_{m \geq 0} \frac{a_m}{\beta_L^m} \alpha_{L,0}^m & \text{if } i = 0 \,; \\ \beta_L \frac{\sigma_w^2}{2} \sum_{m \geq 0} \frac{a_m}{\beta_L^m} \sum_{k_1 + \ldots + k_m = i} \prod_{j=1}^{m} \alpha_{L,k_j} & \text{if } i \geq 1 \,. \end{cases}$$

*where $\beta_L = Q_L(x, x) = Q_L(x', x') = \sum_{i \geq 0} \alpha_{L,i}$ and $(a_m)_{m \geq 0}$ is such that $a_0, a_1 > 0$ and $a_{2i} > 0$ and $a_{2i+1} = 0$ for all $i \geq 1$.*
*As a result, for all $L \geq 2, i \geq 0, \alpha_{L,i} > 0$.*

*Proof.* The result is true for $L = 2$ by lemma A21. Let us prove the result for all $L \geq 3$ by induction.
Let $L \geq 3$, $x, x' \in \mathbb{S}^{d-1}$, $z = x \cdot x'$ and $\beta_l = Q_l(x, x) = Q_l(x', x')$. Assume the result is true for $L$ and let us prove it for $L + 1$. We have that

$$Q_{L+1}(x, y) = Q_L(x, y) + \lambda_{L+1,L+1}^2 (\sigma_b^2 + \frac{\sigma_w^2}{2} f(C_l(x, y))\beta_l) \,.$$

Knowing that $C_l(x,y) = \frac{1}{\beta_l} Q_l(x,y)$, we have that

$$\begin{aligned} f(C_l(x,y)) &= \sum_{m\geq 0} \frac{a_m}{\beta_l^m} C_l(x,y)^m \\ &= \sum_{m\geq 0} \frac{a_m}{\beta_l^m} (\sum_{i\geq 0} \alpha_{l,i} z^i)^m \\ &= \sum_{m\geq 0} \frac{a_m}{\beta_l^m} \sum_{i\geq 0} \sum_{k_1+\ldots+k_m=i} \prod_{j=1}^m \alpha_{l,k_j} z^i \\ &= \sum_{i\geq 0} \Big[ \sum_{m\geq 0} \frac{a_m}{\beta_l^m} \sum_{k_1+\ldots+k_m=i} \prod_{j=1}^m \alpha_{l,k_j} \Big] z^i \,, \end{aligned}$$

which gives the recursive formulas for the coefficients of the analytic decomposition. Observe that the coefficients are non-decreasing wrt $L$. Using lemma A21 we conclude that $\alpha_{L,i} > 0$. □

For depth $L \geq 2$, proposition A5 shows that all coefficient $(\alpha_{L,i})_{i\geq 0}$ are (strictly) positive. It turns out that this is a sufficient condition for the kernel $Q_L$ to be strictly positive definite. We state this in the next proposition. The result can be seen as a consequence of Lemma A12 and Lemma A13. However we will give here a more direct proof.

**Proposition A6** ($Q_L$ is strictly p.d. for $L \geq 2$)**.** *Let $Q$ be an analytic kernel on the unit sphere $\mathbb{S}^{d-1}$, i.e. there exist a sequence of real numbers $(\alpha_i)_{i\geq 0}$ such that for all $x, x' \in \mathbb{S}^{d-1}$*

$$Q(x,x') = \sum_{i\geq 0} \alpha_i (x\cdot x')^i$$

*Assume $\alpha_i > 0$ for all $i \in \mathbb{N}$. Then, $Q$ is strictly positive definite.*
*As a result, for all $L \geq 2$, $T_\nu(Q_L)$ is strictly positive definite, i.e. for any non-zero function $\varphi \in L^2(\mathbb{S}^{d-1}, \nu)$*

$$\langle T_\nu(Q_L)\varphi, \varphi\rangle > 0$$

*$\nu$ is the standard uniform measure on the sphere $\mathbb{S}^{d-1}$.*

*Proof.* Let $Q$ be an analytic kernel on the unit sphere $\mathbb{S}^{d-1}$, that is there exists a sequence of real numbers $(\alpha_i)_{i\geq 0}$ such that for all $x, x' \in \mathbb{S}^{d-1}$

$$Q(x,x') = \sum_{i\geq 0} \alpha_i (x\cdot x')^i \,,$$

and assume $\alpha_i > 0$ for all $i \in \mathbb{N}$. The map $(x,x') \mapsto x\cdot x'$ is trivially a kernel in the sense of Definition 2. For all $i \geq 0$, $(x,x') \mapsto (x\cdot x')^i$ is a kernel as well.[18] It follows that $T_\nu(Q)$ is non-negative definite, as a converging sum of non-negative operators. Let us prove that it is strictly positive definite.
Let $\varphi \in L_2(\mathbb{S}^{d-1}, \nu)$ such that $\langle T_\nu(Q)\,\varphi, \varphi\rangle = 0$. Since $\alpha_i > 0$ for all $i$, we have that for all $i \geq 0$

$$\int\int (x\cdot x')^i \varphi(x)\varphi(x')\,\mathrm{d}\nu(x)\mathrm{d}\nu(x') = 0\,,$$

recalling that $\nu$ is the uniform measure on the sphere $\mathbb{S}^{d-1}$. This yields

$$\int\int P(x\cdot x')\varphi(x)\varphi(x')\,\mathrm{d}\nu(x)\mathrm{d}\nu(x') = 0 \tag{A23}$$

for any polynomial function $P$.
Since $\varphi$ is a function on the sphere $\mathbb{S}^{d-1}$, it can be decomposed in the Spherical Harmonics orthonormal basis $(Y_{k,j})_{k,j}$ (see e.g. [MacRobert, 1967]) as

$$\forall x \in \mathbb{S}^{d-1}, \quad \varphi(x) = \sum_{k\geq 0} \sum_{j=1}^{N(d,k)} b_{k,j} Y_{k,j}(x)$$

[18] See footnote 14.

where $b_{k,j} = \int_{\mathbb{S}^{d-1}} \varphi(w) Y_{k,j}(w)\, \mathrm{d}\nu(w)$.
In particular, equation (A23) is true for the Associated Legendre Polynomials $P_k$. Knowing that $N(d,k)P_k(x\cdot x') = \sum_{j=1}^{N(d,k)} Y_{k,j}(x)Y_{k,j}(x')$, (A23) yields

$$\int\int \sum_{j=1}^{N(d,k)} Y_{k,j}(x)Y_{k,j}(x')\varphi(x)\varphi(x')\,\mathrm{d}\nu(x)\mathrm{d}\nu(x') = 0$$

for all $k \geq 0$. Therefore,

$$\sum_{j=1}^{N(d,k)} b_{k,j}^2 = 0$$

for all $k \geq 0$. We conclude that $\varphi = 0$. □

By Mercer's theorem [Paulsen and Raghupathi, 2016], the kernel $Q_L$ can be decomposed in an orthonormal basis of $L^2(\mathbb{S}^{d-1})$. It turns out that this orthonormal basis is the so-called Spherical Harmonics of $\mathbb{S}^{d-1}$. This is a corollary of the next lemma, which is a classical result [Yang and Salman, 2019].

**Lemma A22** (Spectral decomposition on $\mathbb{S}^{d-1}$)**.** *Let $Q$ be a zonal kernel on $\mathbb{S}^{d-1}$, that is $Q(x,x') = p(x\cdot x')$ for a continuous function $p : [-1,1] \to \mathbb{R}$. Then, there is a sequence $\{\mu_k \geq 0\}_{k\in\mathbb{N}}$ such that for all $x, x' \in \mathbb{S}^{d-1}$*

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

*where $\{Y_{k,j}\}_{k\geq 0, j\in[1:N(d,k)]}$ are spherical harmonics of $\mathbb{S}^{d-1}$ and $N(d,k)$ is the number of harmonics of order $k$. With respect to the standard spherical measure $\nu$ on $\mathbb{S}^{d-1}$, the spherical harmonics form an orthonormal basis of $L^2(\mathbb{S}^{d-1},\nu)$ and $T_\nu(Q)$ is diagonal on this basis.*

*Proof.* We start by giving a brief review of the theory of Spherical Harmonics ([MacRobert, 1967]). For some $k \geq 1$, let $(Y_{k,j})_{1\leq j\leq N(d,k)}$ be the set of Spherical Harmonics of degree $k$. We have $N(d,k) = \frac{2k+d-2}{k}\binom{k+d-3}{d-2}$.

The set of functions $(Y_{k,j})_{k\geq 1, j\in[1:N(d,k)]}$ form an orthonormal basis of $L^2(\mathbb{S}^{d-1},\nu)$, where $\nu$ is the uniform measure on $\mathbb{S}^{d-1}$.

For some function $p$, the Hecke-Funk formula reads

$$\int_{\mathbb{S}^{d-1}} p(\langle x,w\rangle) Y_{k,j}(w)\,\mathrm{d}\nu(w) = \frac{\Omega_{d-1}}{\Omega_d} Y_{k,j}(x) \int_{-1}^{1} p(t) P_k^d(t)(1-t^2)^{(d-3)/2} dt$$

where $\Omega_d$ 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

$$P_k^d(t) = \left(-\frac{1}{2}\right)^k \frac{\Gamma(\frac{d-1}{2})}{\Gamma(k+\frac{d-1}{2})}(1-t^2)^{\frac{3-d}{2}} \frac{\mathrm{d}^k}{\mathrm{d}t^k}(1-t^2)^{k+\frac{d-3}{2}}\,.$$

$(P_k^d)_{k\geq 0}$ 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.

Using the Heck-Funk formula, we prove that $Q$ 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

$$Q(x,x') = \sum_{k\geq 0}\sum_{j=1}^{N(d,k)} \left[\int_{\mathbb{S}^{d-1}} p(\langle w,x'\rangle) Y_{k,j}(w)\mathrm{d}\nu(w)\right] Y_{k,j}(x)\,.$$

Using the Hecke-Funk formula yields

$$Q(x,x') = \sum_{k\geq 0} \sum_{j=1}^{N(d,k)} \left[ \frac{\Omega_{d-1}}{\Omega_d} \int_{-1}^{1} p(t) P_k^d(t) (1-t^2)^{(d-3)/2} dt \right] Y_{k,j}(x) Y_{k,j}(x') \,.$$

We conclude that

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

where $\mu_k = \frac{\Omega_{d-1}}{\Omega_d} \int_{-1}^{1} p(t) P_k^d(t) (1-t^2)^{(d-3)/2} dt$. We also have that $\mu_k \geq 0$ since $Q$ is non-negative by definition. The last statement, follows from the spectral theory of compact self-adjoint operators and the orthonormality of the spherical harmonics (see the appendix of [Yang and Salman, 2019] for details). □

**Corollary A3** (Spectral decomposition of $Q_L$)**.** *For $L \geq 1$, there exist $(\mu_{L,k})_{k\geq 0}$ such that $\mu_{L,k} > 0$ for all $k \geq 0$, and for all $x, x' \in \mathbb{S}^{d-1}$ we have*

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

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

Corollary A3 shows that for any depth $L$, the Spherical Harmonics are the eigenfunctions of the kernel $Q_L$. The fact that $\mu_{L,k} > 0$ is a direct result of Proposition A6. Leveraging this result, we can prove a stronger result, which is the universality of the kernel $Q_L$.

**Proposition A7** (Universality on $\mathbb{S}^{d-1}$)**.** *For all $L \geq 2$, $Q_L$ is universal on $\mathbb{S}^{d-1}$ for $d \geq 2$.*

*Proof.* The result is a consequence of Lemma A15 and Proposition A6. An alternative proof is the following. It is a classical result that the set Spherical Harmonics form an orthonormal basis on $L^2(\mathbb{S}^{d-1}, \nu)$. Leveraging the result from corollary A3, it is straightforward that any continuous function in $L^2(\mathbb{S}^{d-1}, \nu)$ can be approximated by a function of the form $\sum_i Q_L(x_i, .)$ which belongs to the RKHS of $Q_L$. Therefore, $Q_L$ is universal on $\mathbb{S}^{d-1}$. Note that we have not made the assumption that $\sigma_b > 0$. □

## References


G. Yang and S. Schoenholz. Mean field residual networks: On the edge of chaos. In *Advances in Neural Information Processing Systems*, pages 7103–7114, 2017.

S. Hayou, A. Doucet, and J. Rousseau. On the impact of the activation function on deep neural networks training. In *International Conference on Machine Learning*, 2019a.

R.M. Neal. *Bayesian Learning for Neural Networks*, volume 118. Springer Science & Business Media, 1995.

B. Poole, S. Lahiri, M. Raghu, J. Sohl-Dickstein, and S. Ganguli. Exponential expressivity in deep neural networks through transient chaos. In *Advances in Neural Information Processing Systems*, 2016.

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

J. Lee, L. Xiao, S. Schoenholz, Y. Bahri, R. Novak, J. Sohl-Dickstein, and J. Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in Neural Information Processing Systems*. 2019.

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

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

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

S. Hayou, A. Doucet, and J. Rousseau. Mean-field behaviour of neural tangent kernel for deep neural networks. *arXiv preprint arXiv:1905.13654*, 2019b.

S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *International Conference on Machine Learning*, 2015.

S. Hayou, J.F. Ton, A. Doucet, and Y.W. Teh. Robust pruning at initialization. In *International Conference on Learning Representations*, 2021.

G. Yang. Tensor programs i: Wide feedforward or recurrent neural networks of any architecture are Gaussian processes. *arXiv preprint arXiv:1910.12478*, 2019b.

A. Daniely, R. Frostig, and Y. Singer. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. *Advances in Neural Information Processing Systems 29*, 2016.

V.I. Paulsen and M. Raghupathi. *An Introduction to the Theory of Reproducing Kernel Hilbert Spaces*. Cambridge University Press, 2016.

R. M. Dudley. *Real Analysis and Probability*. Cambridge Studies in Advanced Mathematics. Cambridge University Press, 2 edition, 2002.

I. Steinwart. Convergence types and rates in generic Karhunen-Loeve expansions with applications to sample path properties. *Potential Analysis*, 51(3):361–395, 2019.

S. Lang. *Real and Functional Analysis*. Graduate Texts in Mathematics. Springer, New York, 3rd edition, 2012.

U. Grenander. Stochastic processes and statistical inference. *Arkiv Matematik*, 1(3):195–277, 10 1950.

G. Yang and H. Salman. A fine-grained spectral perspective on neural networks. *arXiv preprint 1907.10599*, 2019.

B. Sriperumbudur, K. Fukumizu, and G. Lanckriet. Universality, characteristic kernels and RKHS embedding of measures. *Journal of Machine Learning Research*, 12(70):2389–2410, 2011.

M. Kanagawa, P. Hennig, D. Sejdinovic, and B.K. Sriperumbudur. Gaussian processes and kernel methods: A review on connections and equivalences. *arXiv preprint arXiv:1807.02582*, 2018.

L. Chizat and F. Bach. A note on lazy training in supervised differentiable programming. In *Advances in Neural Information Processing Systems*, 2019.

A. Jacot, F. Gabriel, and C. Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, 2018.

M. Seeger. PAC-Bayesian generalisation error bounds for Gaussian process classification. *Journal of Machine Learning Research*, page 233–269, 02 2002.

S. Arora, S.S. Du, W. Hu, Z. Li, R. Salakhutdinov, and R. Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, 2019.

V. Shankar, A. Fang, W. Guo, S. Fridovich-Keil, L. Schmidt, J. Ragan-Kelley, and B. Recht. Neural kernels without tangents. *International Conference on Machine Learning*, 2020.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference 2016*, 2016.

B. He, B. Lakshminarayanan, and Y. W. Teh. Bayesian deep ensembles via the neural tangent kernel. *Advances in Neural Information Processing Systems*, 2020.

R. Novak, L. Xiao, J. Hron, J. Lee, A. Alemi, J. Sohl-Dickstein, and S. Schoenholz. Neural tangents: Fast and easy infinite neural networks in Python. In *International Conference on Learning Representations*, 2020. URL https://github.com/google/neural-tangents.

J. Lee, S. Schoenholz, J. Pennington, B. Adlam, L. Xiao, R. Novak, and J. Sohl-Dickstein. Finite versus infinite neural networks: an empirical study. *Advances in Neural Information Processing Systems*, 2020.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

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

N. Aronszajn. Theory of reproducing kernels. *Transactions of the American Mathematical Society*, pages 337–404, 1950.

C. Micchelli, Y. Xu, and H. Zhang. Universal kernels. *Journal of Machine Learning Research*, pages 2651–2667, 2006.

O. Kounchev. *Multivariate Polysplines: Applications to Numerical and Wavelet Analysis*. Elsevier Science, 2001.

J. Bradbury, R. Frostig, P. Hawkins, M. Johnson, C. Leary, D. Maclaurin, and S. Wanderman-Milne. JAX: composable transformations of Python+NumPy programs, 2018. URL http://github.com/google/jax.

K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, 2015.

R. Novak, L. Xiao, J. Lee, Y. Bahri, G. Yang, J. Hron, D. A Abolafia, J. Pennington, and J. Sohl-Dickstein. Bayesian deep convolutional networks with many channels are Gaussian processes. In *International Conference on Learning Representations*, 2019.

C. Wang, G. Zhang, and R. Grosse. Picking winning tickets before training by preserving gradient flow. In *International Conference on Learning Representations*, 2020.

A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems*, pages 8026–8037, 2019.

S De and SL Smith. Batch normalization biases residual blocks towards the identity function in deep networks. *Advances in Neural Information Processing Systems*, 2020.

H. Zhang, Y. N Dauphin, and T. Ma. Fixup initialization: Residual learning without normalization. In *International Conference on Learning Representations*, 2019.

T.M. MacRobert. *Spherical Harmonics: An Elementary Treatise on Harmonic Functions with Applications*. Pergamon Press, 1967.

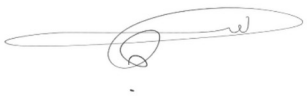
## Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

| | |
|---|---|
| Title of Paper | Stable ResNet |
| Publication Status | ■ Published □ Accepted for Publication<br>□ Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style |
| Publication Details | Hayou, S., Clerico, E., He, B., Deligiannidis, G., Doucet, A., and Rousseau, J. (2020). Stable ResNet. Proceedings of the 24th International Conference on Artificial Intelligence and Statistics (AISTATS 2021). |

### Student Confirmation

| | | | |
|---|---|---|---|
| Student Name: | Soufiane Hayou | | |
| Contribution to the Paper | I worked on the theory and proofs of the results in Sections 2, 4, half of section 5, and 6. I also helped on the experiments section. Eugenio Clerico worked on the rest of the theory in the paper, while Bobby He worked on the experiments section. Arnaud Doucet, George Deligiannidis and Judith Rousseau provided helpful insights and contributed to the writing of the draft, checking the proofs, and proof-reading. | | |
| Signature |  | Date | 21/01/2021 |

### Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

| | | |
|---|---|---|
| Supervisor name and title: Arnaud Doucet | | |
| Supervisor comments | | |
| Signature | Date | |

This completed form should be included in the thesis, at the end of the relevant chapter.

# 5

# Neural Networks Pruning at Initialization

# ROBUST PRUNING AT INITIALIZATION

**Soufiane Hayou, Jean-Francois Ton, Arnaud Doucet & Yee Whye Teh**
Department of Statistics
University of Oxford
United Kingdom
{soufiane.hayou, ton, doucet, teh}@stats.ox.ac.uk

## ABSTRACT

Overparameterized Neural Networks (NN) display state-of-the-art performance. However, there is a growing need for smaller, energy-efficient, neural networks to be able to use machine learning applications on devices with limited computational resources. A popular approach consists of using pruning techniques. While these techniques have traditionally focused on pruning pre-trained NN (LeCun et al., 1990; Hassibi et al., 1993), recent work by Lee et al. (2018) has shown promising results when pruning at initialization. However, for Deep NNs, such procedures remain unsatisfactory as the resulting pruned networks can be difficult to train and, for instance, they do not prevent one layer from being fully pruned. In this paper, we provide a comprehensive theoretical analysis of Magnitude and Gradient based pruning at initialization and training of sparse architectures. This allows us to propose novel principled approaches which we validate experimentally on a variety of NN architectures.

## 1 INTRODUCTION

Overparameterized deep NNs have achieved state of the art (SOTA) performance in many tasks (Nguyen and Hein, 2018; Du et al., 2019; Zhang et al., 2016; Neyshabur et al., 2019). However, it is impractical to implement such models on small devices such as mobile phones. To address this problem, network pruning is widely used to reduce the time and space requirements both at training and test time. The main idea is to identify weights that do not contribute significantly to the model performance based on some criterion, and remove them from the NN. However, most pruning procedures currently available can only be applied after having trained the full NN (LeCun et al., 1990; Hassibi et al., 1993; Mozer and Smolensky, 1989; Dong et al., 2017) although methods that consider pruning the NN during training have become available. For example, Louizos et al. (2018) propose an algorithm which adds a $L_0$ regularization on the weights to enforce sparsity while Carreira-Perpiñán and Idelbayev (2018); Alvarez and Salzmann (2017); Li et al. (2020) propose the inclusion of compression inside training steps. Other pruning variants consider training a secondary network that learns a pruning mask for a given architecture (Li et al. (2020); Liu et al. (2019)).

Recently, Frankle and Carbin (2019) have introduced and validated experimentally the Lottery Ticket Hypothesis which conjectures the existence of a sparse subnetwork that achieves similar performance to the original NN. These empirical findings have motivated the development of pruning at initialization such as SNIP (Lee et al. (2018)) which demonstrated similar performance to classical pruning methods of pruning-after-training. Importantly, pruning at initialization never requires training the complete NN and is thus more memory efficient, allowing to train deep NN using limited computational resources. However, such techniques may suffer from different problems. In particular, nothing prevents such methods from pruning one whole layer of the NN, making it untrainable. More generally, it is typically difficult to train the resulting pruned NN (Li et al., 2018). To solve this situation, Lee et al. (2020) try to tackle this issue by enforcing dynamical isometry using orthogonal weights, while Wang et al. (2020) (GraSP) uses Hessian based pruning to preserve gradient flow. Other work by Tanaka et al. (2020) considers a data-agnostic iterative approach using the concept of synaptic flow in order to avoid the layer-collapse phenomenon (pruning a whole layer). In our work, we use principled scaling and re-parameterization to solve this issue, and show numerically that our algorithm achieves SOTA performance on CIFAR10, CIFAR100, TinyImageNet and ImageNet in some scenarios and remains competitive in others.

Table 1: Classification accuracies on CIFAR10 for Resnet with varying depths and sparsities using SNIP (Lee et al. (2018)) and our algorithm SBP-SR

| | ALGORITHM | 90% | 95% | 98% | 99.5% | 99.9% |
|---|---|---|---|---|---|---|
| RESNET32 | SNIP | $92.26 \pm 0.32$ | $91.18 \pm 0.17$ | $87.78 \pm 0.16$ | 77.56±0.36 | 9.98±0.08 |
| | SBP-SR | **92.56 ± 0.06** | $91.21 \pm 0.30$ | **88.25 ± 0.35** | **79.54±1.12** | **51.56±1.12** |
| RESNET50 | SNIP | $91.95 \pm 0.13$ | $92.12 \pm 0.34$ | $89.26 \pm 0.23$ | 80.49±2.41 | 19.98±14.12 |
| | SBP-SR | **92.05 ± 0.06** | **92.74± 0.32** | **89.57 ± 0.21** | **82.68±0.52** | **58.76±1.82** |
| RESNET104 | SNIP | $93.25 \pm 0.53$ | $92.98 \pm 0.12$ | $91.58 \pm 0.19$ | 33.63±33.27 | 10.11±0.09 |
| | SBP-SR | **94.69 ± 0.13** | **93.88 ± 0.17** | **92.08 ± 0.14** | **87.47±0.23** | **72.70±0.48** |

In this paper, we provide novel algorithms for Sensitivity-Based Pruning (SBP), i.e. pruning schemes that prune a weight $W$ based on the magnitude of $|W \frac{\partial \mathcal{L}}{\partial W}|$ at initialization where $\mathcal{L}$ is the loss. Experimentally, compared to other available one-shot pruning schemes, these algorithms provide state-of-the-art results (this might not be true in some regimes). Our work is motivated by a new theoretical analysis of gradient back-propagation relying on the mean-field approximation of deep NN (Hayou et al., 2019; Schoenholz et al., 2017; Poole et al., 2016; Yang and Schoenholz, 2017; Xiao et al., 2018; Lee et al., 2018; Matthews et al., 2018). Our contribution is threefold:

- For deep fully connected FeedForward NN (FFNN) and Convolutional NN (CNN), it has been previously shown that only an initialization on the so-called Edge of Chaos (EOC) make models trainable; see e.g. (Schoenholz et al., 2017; Hayou et al., 2019). For such models, we show that an EOC initialization is also necessary for SBP to be efficient. Outside this regime, one layer can be fully pruned.
- For these models, pruning pushes the NN out of the EOC making the resulting pruned model difficult to train. We introduce a simple rescaling trick to bring the pruned model back in the EOC regime, making the pruned NN easily trainable.
- Unlike FFNN and CNN, we show that Resnets are better suited for pruning at initialization since they 'live' on the EOC by default (Yang and Schoenholz, 2017). However, they can suffer from exploding gradients, which we resolve by introducing a re-parameterization, called 'Stable Resnet' (SR). The performance of the resulting SBP-SR pruning algorithm is illustrated in Table 1: SBP-SR allows for pruning up to $99.5\%$ of ResNet104 on CIFAR10 while still retaining around $87\%$ test accuracy.

The precise statements and proofs of the theoretical results are given in the Supplementary. Appendix H also includes the proof of a weak version of the Lottery Ticket Hypothesis (Frankle and Carbin, 2019) showing that, starting from a randomly initialized NN, there exists a subnetwork initialized on the EOC.

## 2 SENSITIVITY PRUNING FOR FFNN/CNN AND THE RESCALING TRICK

### 2.1 SETUP AND NOTATIONS

Let $x$ be an input in $\mathbb{R}^d$. A NN of depth $L$ is defined by

$$y^l(x) = \mathcal{F}_l(W^l, y^{l-1}(x)) + B^l, \quad 1 \leq l \leq L, \tag{1}$$

where $y^l(x)$ is the vector of pre-activations, $W^l$ and $B^l$ are respectively the weights and bias of the $l^{\text{th}}$ layer and $\mathcal{F}_l$ is a mapping that defines the nature of the layer. The weights and bias are initialized with $W^l \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_w^2/v_l)$, where $v_l$ is a scaling factor used to control the variance of $y^l$, and $B^l \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_b^2)$. Hereafter, $M_l$ denotes the number of weights in the $l^{th}$ layer, $\phi$ the activation function and $[\![m : n]\!] := \{m, m+1, ..., n\}$ for $m \leq n$. Two examples of such architectures are:

- **Fully connected FFNN**. For a FFNN of depth $L$ and widths $(N_l)_{0 \leq l \leq L}$, we have $v_l = N_{l-1}$, $M_l = N_{l-1}N_l$ and

$$y_i^1(x) = \sum_{j=1}^{d} W_{ij}^1 x_j + B_i^1, \quad y_i^l(x) = \sum_{j=1}^{N_{l-1}} W_{ij}^l \phi(y_j^{l-1}(x)) + B_i^l \quad \text{for } l \geq 2. \tag{2}$$

• **CNN**. For a 1D CNN of depth $L$, number of channels $(n_l)_{l\leq L}$, and number of neurons per channel $(N_l)_{l\leq L}$, we have

$$y^1_{i,\alpha}(x) = \sum_{j=1}^{n_{l-1}} \sum_{\beta\in ker_l} W^1_{i,j,\beta} x_{j,\alpha+\beta} + b^1_i,\ y^l_{i,\alpha}(x) = \sum_{j=1}^{n_{l-1}} \sum_{\beta\in ker_l} W^l_{i,j,\beta} \phi(y^{l-1}_{j,\alpha+\beta}(x)) + b^l_i,\ \text{for } l \geq 2, \tag{3}$$

where $i \in [1 : n_l]$ is the channel index, $\alpha \in [0 : N_l - 1]$ is the neuron location, $ker_l = [-k_l : k_l]$ is the filter range, and $2k_l + 1$ is the filter size. To simplify the analysis, we assume hereafter that $N_l = N$ and $k_l = k$ for all $l$. Here, we have $v_l = n_{l-1}(2k + 1)$ and $M_l = n_{l-1}n_l(2k + 1)$. We assume periodic boundary conditions; so $y^l_{i,\alpha} = y^l_{i,\alpha+N} = y^l_{i,\alpha-N}$. Generalization to multidimensional convolutions is straightforward.

When no specific architecture is mentioned, $(W^l_i)_{1\leq i\leq M_l}$ denotes the weights of the $l^{\text{th}}$ layer. In practice, a pruning algorithm creates a binary mask $\delta$ over the weights to force the pruned weights to be zero. The neural network after pruning is given by

$$y^l(x) = \mathcal{F}_l(\delta^l \circ W^l, y^{l-1}(x)) + B^l, \tag{4}$$

where $\circ$ is the Hadamard (i.e. element-wise) product. In this paper, we focus on pruning at initialization. The mask is typically created by using a vector $g^l$ of the same dimension as $W^l$ using a mapping of choice (see below), we then prune the network by keeping the weights that correspond to the top $k$ values in the sequence $(g^l_i)_{i,l}$ where $k$ is fixed by the sparsity that we want to achieve. There are three popular types of criteria in the literature :

• **Magnitude based pruning (MBP)**: We prune weights based on the magnitude $|W|$.

• **Sensitivity based pruning (SBP)**: We prune the weights based on the values of $|W\frac{\partial\mathcal{L}}{\partial W}|$ where $\mathcal{L}$ is the loss. This is motivated by $\mathcal{L}_W \approx \mathcal{L}_{W=0} + W\frac{\partial\mathcal{L}}{\partial W}$ used in SNIP (Lee et al. (2018)).

• **Hessian based pruning (HBP)**: We prune the weights based on some function that uses the Hessian of the loss function as in GraSP (Wang et al., 2020).

In the remainder of the paper, we focus exclusively on SBP while our analysis of MBP is given in Appendix E. We leave HBP for future work. However, we include empirical results with GraSP (Wang et al., 2020) in Section 4.

Hereafter, we denote by $s$ the sparsity, i.e. the fraction of weights we want to prune. Let $A_l$ be the set of indices of the weights in the $l^{\text{th}}$ layer that are pruned, i.e. $A_l = \{i \in [1 : M_l],\ \text{s.t. } \delta^l_i = 0\}$. We define the critical sparsity $s_{cr}$ by

$$s_{cr} = \min\{s \in (0, 1),\ \text{s.t. } \exists l, |A_l| = M_l\},$$

where $|A_l|$ is the cardinality of $A_l$. Intuitively, $s_{cr}$ represents the maximal sparsity we are allowed to choose without fully pruning at least one layer. $s_{cr}$ is random as the weights are initialized randomly. Thus, we study the behaviour of the expected value $\mathbb{E}[s_{cr}]$ where, hereafter, **all expectations are taken w.r.t. to the random initial weights**. This provides theoretical guidelines for pruning at initialization.

For all $l \in [1 : L]$, we define $\alpha_l$ by $v_l = \alpha_l N$ where $N > 0$, and $\zeta_l > 0$ such that $M_l = \zeta_l N^2$, where we recall that $v_l$ is a scaling factor controlling the variance of $y^l$ and $M_l$ is the number of weights in the $l^{\text{th}}$ layer. This notation assumes that, in each layer, the number of weights is quadratic in the number of neurons, which is satisfied by classical FFNN and CNN architectures.

### 2.2 Sensitivity-Based Pruning (SBP)

SBP is a data-dependent pruning method that uses the data to compute the gradient *with* backpropagation at initialization (one-shot pruning).We randomly sample a batch and compute the gradients of the loss with respect to each weight. The mask is then defined by $\delta^l_i = \mathbb{I}(|W^l_i\frac{\partial\mathcal{L}}{\partial W^l_i}| \geq t_s)$, where $t_s = |W\frac{\partial\mathcal{L}}{\partial W}|^{(k_s)}$ and $k_s = (1 - s)\sum_l M_l$ and $|W\frac{\partial\mathcal{L}}{\partial W}|^{(k_s)}$ is the $k^{\text{th}}_s$ order statistics of the sequence $(|W^l_i\frac{\partial\mathcal{L}}{\partial W^l_i}|)_{1\leq l\leq L, 1\leq i\leq M_l}$.

However, this simple approach suffers from the well-known exploding/vanishing gradients problem which renders the first/last few layers respectively susceptible to be completely pruned. We give a formal definition to this problem.

**Definition 1** (Well-conditioned & ill-conditioned NN). *Let $m_l = \mathbb{E}[|W_1^l \frac{\partial \mathcal{L}}{\partial W_1^l}|^2]$ for $l \in [1:L]$. We say that the NN is well-conditioned if there exist $A, B > 0$ such that for all $L \geq 1$ and $l \in [1:L]$ we have $A \leq m_l/m_L \leq B$, and it is ill-conditioned otherwise.*

Understanding the behaviour of gradients at initialization is thus crucial for SBP to be efficient. Using a mean-field approach, such analysis has been carried out in (Schoenholz et al., 2017; Hayou et al., 2019; Xiao et al., 2018; Poole et al., 2016; Yang, 2019) where it has been shown that an initialization known as the EOC is beneficial for DNN training. The mean-field analysis of DNNs relies on two standard approximations that we will also use here.

**Approximation 1** (Mean-Field Approximation). *When $N_l \gg 1$ for FFNN or $n_l \gg 1$ for CNN, we use the approximation of infinitely wide NN. This means infinite number of neurons per layer for fully connected layers and infinite number of channels per layer for convolutional layers.*

**Approximation 2** (Gradient Independence). *The weights used for forward propagation are independent from those used for back-propagation.*

These two approximations are ubiquitous in literature on the mean-field analysis of neural networks. They have been used to derive theoretical results on signal propagation (Schoenholz et al., 2017; Hayou et al., 2019; Poole et al., 2016; Yang, 2019; Yang and Schoenholz, 2017; Yang et al., 2019) and are also key tools in the derivation of the Neural Tangent Kernel (Jacot et al., 2018; Arora et al., 2019; Hayou et al., 2020). Approximation 1 simplifies the analysis of the forward propagation as it allows the derivation of closed-form formulas for covariance propagation. Approximation 2 does the same for back-propagation. See Appendix A for a detailed discussion of these approximations. Throughout the paper, we provide numerical results that substantiate the theoretical results that we derive using these two approximations. We show that these approximations lead to excellent match between theoretical results and numerical experiments.

**Edge of Chaos (EOC):** For inputs $x, x'$, let $c^l(x, x')$ be the correlation between $y^l(x)$ and $y^l(x')$. From (Schoenholz et al., 2017; Hayou et al., 2019), there exists a so-called correlation function $f$ that depends on $(\sigma_w, \sigma_b)$ such that $c^{l+1}(x, x') = f(c^l(x, x'))$. Let $\chi(\sigma_b, \sigma_w) = f'(1)$. The EOC is the set of hyperparameters $(\sigma_w, \sigma_b)$ satisfying $\chi(\sigma_b, \sigma_w) = 1$. When $\chi(\sigma_b, \sigma_w) > 1$, we are in the Chaotic phase, the gradient explodes and $c^l(x, x')$ converges exponentially to some $c < 1$ for $x \neq x'$ and the resulting output function is discontinuous everywhere. When $\chi(\sigma_b, \sigma_w) < 1$, we are in the Ordered phase where $c^l(x, x')$ converges exponentially fast to 1 and the NN outputs constant functions. Initialization on the EOC allows for better information propagation (see Supplementary for more details).

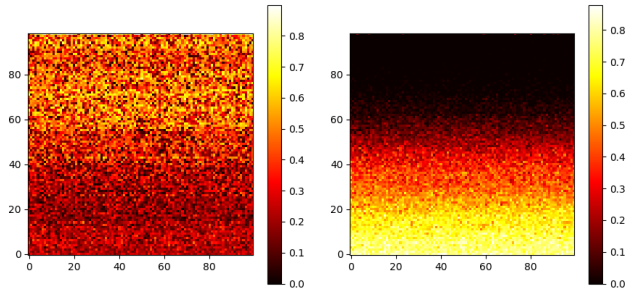
Hence, by leveraging the above results, we show that an initialization outside the EOC will lead to an ill-conditioned NN.

**Theorem 1** (EOC Initialization is crucial for SBP). *Consider a NN of type (2) or (3) (FFNN or CNN). Assume $(\sigma_w, \sigma_b)$ are chosen on the ordered phase, i.e. $\chi(\sigma_b, \sigma_w) < 1$, then the NN is ill-conditioned. Moreover, we have*

$$\mathbb{E}[s_{cr}] \leq \frac{1}{L}\left(1 + \frac{\log(\kappa L N^2)}{\kappa}\right) + \mathcal{O}\left(\frac{1}{\kappa^2 \sqrt{LN^2}}\right),$$

*where $\kappa = |\log \chi(\sigma_b, \sigma_w)|/8$. If $(\sigma_w, \sigma_b)$ are on the EOC, i.e. $\chi(\sigma_b, \sigma_w) = 1$, then the NN is well-conditioned. In this case, $\kappa = 0$ and the above upper bound no longer holds.*

The proof of Theorem 1 relies on the behaviour of the gradient norm at initialization. On the ordered phase, the gradient norm vanishes exponentially quickly as it back-propagates, thus resulting in an ill-conditioned network. We use another approximation for the sake of simplification of the proof (Approximation 3 in the Supplementary) but the result holds without this approximation although the resulting constants would be a bit different. Theorem 1 shows that the upper bound decreases the farther $\chi(\sigma_b, \sigma_w)$ is from 1, i.e. the farther the initialization is from the EOC. For constant width FFNN with $L = 100$, $N = 100$ and $\kappa = 0.2$, the theoretical upper bound is $\mathbb{E}[s_{cr}] \lessapprox 27\%$ while we obtain $\mathbb{E}[s_{cr}] \approx 22\%$ based on 10 simulations. A similar result can be obtained when the NN is initialized on the chaotic phase; in this case too, the NN is ill-conditioned. To illustrate these results, Figure 1 shows the impact of the initialization with sparsity $s = 70\%$. The dark area in Figure 1(b) corresponds to layers that are fully pruned in the chaotic phase due to exploding gradients. Using an EOC initialization, Figure 1(a) shows that pruned weights are well distributed in the NN, ensuring that no layer is fully pruned.



(a) Edge of Chaos (b) Chaotic phase

Figure 1: Percentage of weights kept after SBP applied to a randomly initialized FFNN with depth 100 and width 100 for 70% sparsity on MNIST. Each pixel $(i, j)$ corresponds to a neuron and shows the proportion of connections to neuron $(i, j)$ that have not been pruned. The EOC (a) allows us to preserve a uniform spread of the weights, whereas the Chaotic phase (b), due to exploding gradients, prunes entire layers.

### 2.3 Training Pruned Networks Using the Rescaling Trick

We have shown previously that an initialization on the EOC is crucial for SBP. However, we have not yet addressed the key problem of training the resulting pruned NN. This can be very challenging in practice (Li et al., 2018), especially for deep NN.

Consider as an example a FFNN architecture. After pruning, we have for an input $x$

$$\hat{y}_i^l(x) = \sum_{j=1}^{N_{l-1}} W_{ij}^l \delta_{ij}^l \phi(\hat{y}_j^{l-1}(x)) + B_i^l, \quad \text{for } l \geq 2, \tag{5}$$

where $\delta$ is the pruning mask. While the original NN initialized on the EOC was satisfying $c^{l+1}(x, x') = f(c^l(x, x'))$ for $f'(1) = \chi(\sigma_b, \sigma_w) = 1$, the pruned architecture leads to $\hat{c}^{l+1}(x, x') = f_{\text{pruned}}(\hat{c}^l(x, x'))$ with $f'_{\text{pruned}}(1) \neq 1$, hence *pruning destroys the EOC*. Consequently, the pruned NN will be difficult to train (Schoenholz et al., 2017; Hayou et al., 2019) especially if it is deep. Hence, we propose to bring the pruned NN back on the EOC. This approach consists of rescaling the weights obtained after SBP in each layer by factors that depend on the pruned architecture itself.

**Proposition 1** (Rescaling Trick)**.** *Consider a NN of type (2) or (3) (FFNN or CNN) initialized on the EOC. Then, after pruning, the pruned NN is not initialized on the EOC anymore. However, the rescaled pruned NN*

$$y^l(x) = \mathcal{F}(\rho^l \circ \delta^l \circ W^l, y^{l-1}(x)) + B^l, \quad \textit{for } l \geq 1, \tag{6}$$

*where*

$$\rho_{ij}^l = (\mathbb{E}[N_{l-1}(W_{i1}^l)^2 \delta_{i1}^l])^{-\frac{1}{2}} \textit{ for FFNN}, \quad \rho_{i,j,\beta}^l = (\mathbb{E}[n_{l-1}(W_{i,1,\beta}^l)^2 \delta_{i,1,\beta}^l])^{-\frac{1}{2}} \textit{ for CNN}, \tag{7}$$

*is initialized on the EOC. (The scaling is constant across $j$).*

The scaling factors in equation 7 are easily approximated using the weights kept after pruning. Algorithm 1 (see Appendix I) details a practical implementation of this rescaling technique for FFNN. We illustrate experimentally the benefits of this approach in Section 4.

## 3 Sensitivity-Based Pruning for Stable Residual Networks

Resnets and their variants (He et al., 2015; Huang et al., 2017) are currently the best performing models on various classification tasks (CIFAR10, CIFAR100, ImageNet etc (Kolesnikov et al., 2019)). Thus, understanding Resnet pruning at initialization is of crucial interest. Yang and Schoenholz (2017) showed that Resnets naturally 'live' on the EOC. Using this result, we show that Resnets are actually better suited to SBP than FFNN and CNN. However, Resnets suffer from an exploding gradient problem (Yang and Schoenholz, 2017) which might affect the performance of SBP. We

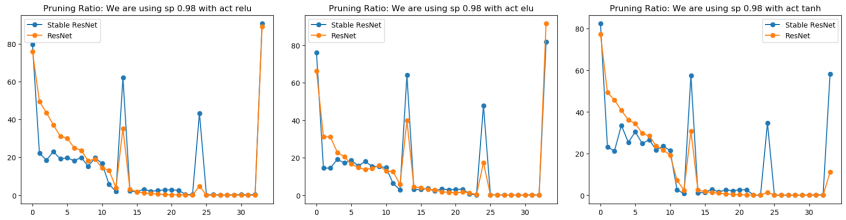



Figure 2: Percentage of non-pruned weights per layer in a ResNet32 for our Stable ResNet32 and standard Resnet32 with Kaiming initialization on CIFAR10. With Stable Resnet, we prune less aggressively weights in the deeper layers than for standard Resnet.

address this issue by introducing a new Resnet parameterization. Let a standard Resnet architecture be given by

$$y^1(x) = \mathcal{F}(W^1, x), \quad y^l(x) \quad = y^{l-1}(x) + \mathcal{F}(W^l, y^{l-1}), \quad \text{for } l \geq 2, \tag{8}$$

where $\mathcal{F}$ defines the blocks of the Resnet. Hereafter, we assume that $\mathcal{F}$ is either of the form (2) or (3) (FFNN or CNN).

The next theorem shows that Resnets are well-conditioned independently from the initialization and are thus well suited for pruning at initialization.

**Theorem 2** (Resnet are Well-Conditioned)**.** *Consider a Resnet with either Fully Connected or Convolutional layers and ReLU activation function. Then for all $\sigma_w > 0$, the Resnet is well-conditioned. Moreover, for all $l \in \{1, ..., L\}$, we have $m^l = \Theta((1 + \frac{\sigma_w^2}{2})^L)$.*

The above theorem proves that Resnets are always well-conditioned. However, taking a closer look at $m^l$, which represents the variance of the pruning criterion (Definition 1), we see that it grows exponentially in the number of layers $L$. Therefore, this could lead to a 'higher variance of pruned networks' and hence high variance test accuracy. To this end, we propose a Resnet parameterization which we call Stable Resnet. Stable Resnets prevent the second moment from growing exponentially as shown below.

**Proposition 2** (Stable Resnet)**.** *Consider the following Resnet parameterization*

$$y^l(x) = y^{l-1}(x) + \frac{1}{\sqrt{L}}\mathcal{F}(W^l, y^{l-1}), \quad \text{for } l \geq 2, \tag{9}$$

*then the NN is well-conditioned for all $\sigma_w > 0$. Moreover, for all $l \leq L$ we have $m^l = \Theta(L^{-1})$.*

In Proposition 2, $L$ is not the number of layers but the number of blocks. For example, ResNet32 has 15 blocks and 32 layers, hence $L = 15$. Figure 2 shows the percentage of weights in each layer kept after pruning ResNet32 and Stable ResNet32 at initialization. The jumps correspond to limits between sections in ResNet32 and are caused by max-pooling. Within each section, Stable Resnet tends to have a more uniform distribution of percentages of weights kept after pruning compared to standard Resnet. In Section 4 we show that this leads to better performance of Stable Resnet compared to standard Resnet. Further theoretical and experimental results for Stable Resnets are presented in (Hayou et al., 2021).

In the next proposition, we establish that, unlike FFNN or CNN, we do not need to rescale the pruned Resnet for it to be trainable as it lives naturally on the EOC before and after pruning.

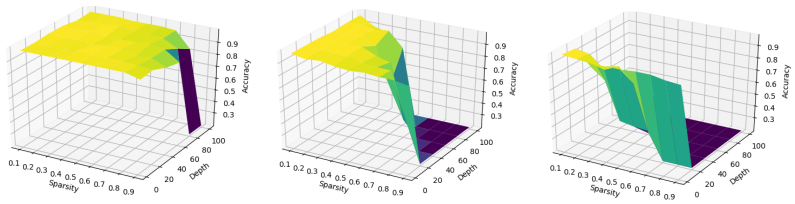
**Proposition 3** (Resnet live on the EOC even after pruning)**.** *Consider a Residual NN with blocks of type FFNN or CNN. Then, after pruning, the pruned Residual NN is initialized on the EOC.*

## 4 Experiments

In this section, we illustrate empirically the theoretical results obtained in the previous sections. We validate the results on MNIST, CIFAR10, CIFAR100 and Tiny ImageNet.

### 4.1 Initialization and Rescaling

According to Theorem 1, an EOC initialization is necessary for the network to be well-conditioned. We train FFNN with tanh activation on MNIST, varying depth $L \in \{2, 20, 40, 60, 80, 100\}$ and




(a) EOC Init & Rescaling (b) EOC Init (c) Ordered phase Init

Figure 3: Accuracy on MNIST with different initialization schemes including EOC with rescaling, EOC without rescaling, Ordered phase, with varying depth and sparsity. This shows that rescaling to be on the EOC allows us to train not only much deeper but also sparser models.

sparsity $s \in \{10\%, 20\%, .., 90\%\}$. We use SGD with batchsize 100 and learning rate $10^{-3}$, which we found to be optimal using a grid search with an exponential scale of 10. Figure 3 shows the test accuracy after 10k iterations for 3 different initialization schemes: *Rescaled EOC*, *EOC*, *Ordered*. On the Ordered phase, the model is untrainable when we choose sparsity $s > 40\%$ and depth $L > 60$ as one layer being fully pruned. For an EOC initialization, the set $(s, L)$ for which NN are trainable becomes larger. However, the model is still untrainable for highly sparse deep networks as the sparse NN is no longer initialized on the EOC (see Proposition 1). As predicted by Proposition 1, after application of the rescaling trick to bring back the pruned NN on the EOC, the pruned NN can be trained appropriately.

Table 2: Classification accuracies for CIFAR10 and CIFAR100 after pruning

| | CIFAR10 | | | CIFAR100 | | |
|---|---|---|---|---|---|---|
| SPARSITY | 90% | 95% | 98% | 90% | 95% | 98% |
| **ResNet32** (NO PRUNING) | 94.80 | - | - | 74.64 | - | - |
| OBD LECUN ET AL. (1990) | 93.74 | 93.58 | 93.49 | 73.83 | 71.98 | 67.79 |
| RANDOM PRUNING | 89.95±0.23 | 89.68±0.15 | 86.13±0.25 | 63.13±2.94 | 64.55±0.32 | 19.83±3.21 |
| MBP | 90.21±0.55 | 88.35±0.75 | 86.83±0.27 | 67.07±0.31 | 64.92±0.77 | 59.53±2.19 |
| SNIP LEE ET AL. (2018) | 92.26 ± 0.32 | 91.18 ± 0.17 | 87.78 ± 0.16 | 69.31 ± 0.52 | 65.63 ± 0.15 | 55.70 ± 1.13 |
| GRASP WANG ET AL. (2020) | 92.20±0.31 | 91.39±0.25 | **88.70±0.42** | 69.24 ± 0.24 | 66.50 ± 0.11 | 58.43 ± 0.43 |
| GRASP-SR | 92.30±0.19 | 91.16±0.13 | 87.8 ± 0.32 | 69.12 ± 0.15 | 65.49 ± 0.21 | 58.63 ± 0.23 |
| SYNFLOW TANAKA ET AL. (2020) | 92.01±0.22 | **91.67±0.17** | 88.10 ± 0.25 | 69.03 ± 0.20 | 65.23 ± 0.31 | 58.73 ± 0.30 |
| SBP-SR (STABLE RESNET) | **92.56 ± 0.06** | 91.21 ± 0.30 | 88.25 ± 0.35 | **69.51 ± 0.21** | **66.72 ± 0.12** | **59.51 ± 0.15** |
| **ResNet50** (NO PRUNING) | 94.90 | - | - | 74.9 | - | - |
| RANDOM PRUNING | 85.11±4.51 | 88.76±0.21 | 85.32±0.47 | 65.67±0.57 | 60.23±2.21 | 28.32±10.35 |
| MBP | 90.11 ± 0.32 | 89.06 ± 0.09 | 87.32 ± 0.16 | 68.51 ± 0.21 | 63.32 ± 1.32 | 55.21 ± 0.35 |
| SNIP | 91.95 ± 0.13 | 92.12 ± 0.34 | 89.26 ± 0.23 | 70.43 ± 0.43 | 67.85 ± 1.02 | 60.38 ± 0.78 |
| GRASP | **92.10 ± 0.21** | 91.74 ± 0.35 | **89.97± 0.25** | 70.53±0.32 | 67.84±0.25 | 63.88±0.45 |
| SYNFLOW | 92.05 ± 0.20 | 91.83 ± 0.23 | 89.61± 0.17 | 70.43±0.30 | 67.95±0.22 | 63.95±0.11 |
| SBP-SR | 92.05 ± 0.06 | **92.74± 0.32** | 89.57 ± 0.21 | **71.79 ± 0.13** | **68.98 ± 0.15** | **64.45 ± 0.34** |
| **ResNet104** (NO PRUNING) | 94.92 | - | - | 75.24 | - | - |
| RANDOM PRUNING | 89.80±0.33 | 87.86±1.22 | 85.52±2.12 | 66.73±1.32 | 64.98±0.11 | 30.31±4.51 |
| MBP | 90.05 ± 1.23 | 88.95±0.65 | 87.83±1.21 | 69.57±0.35 | 64.31±0.78 | 60.21±2.41 |
| SNIP | 93.25 ± 0.53 | 92.98 ± 0.12 | 91.58 ± 0.19 | 71.94 ± 0.22 | 68.73±0.09 | 63.31 ± 0.41 |
| GRASP | 93.08 ± 0.17 | 92.93 ± 0.09 | 91.19±0.35 | 73.33±0.21 | 70.95 ± 1.12 | 66.91±0.33 |
| SYNFLOW | 93.43 ± 0.10 | 92.85 ± 0.18 | 91.03±0.25 | 72.85±0.20 | 70.33 ± 0.15 | 67.02±0.10 |
| SBP-SR | **94.69 ± 0.13** | **93.88 ± 0.17** | **92.08 ± 0.14** | **74.17 ± 0.11** | **71.84 ± 0.13** | **67.73 ± 0.28** |

### 4.2 RESNET AND STABLE RESNET

Although Resnets are adapted to SBP (i.e. they are always well-conditioned for all $\sigma_w > 0$), Theorem 2 shows that the magnitude of the pruning criterion grows exponentially w.r.t. the depth $L$. To resolve this problem we introduced Stable Resnet. We call our pruning algorithm for ResNet SBP-SR (SBP with Stable Resnet). Theoretically, we expect SBP-SR to perform better than other methods for deep Resnets according to Proposition 2. Table 2 shows test accuracies for ResNet32, ResNet50 and ResNet104 with varying sparsities $s \in \{90\%, 95\%, 98\%\}$ on CIFAR10 and CIFAR100. For all our experiments, we use a setup similar to (Wang et al., 2020), i.e. we use SGD for 160 and 250 epochs for CIFAR10 and CIFAR100, respectively. We use an initial learning rate of 0.1 and decay it by 0.1

at $1/2$ and $3/4$ of the number of total epoch. In addition, we run all our experiments 3 times to obtain more stable and reliable test accuracies. As in (Wang et al., 2020), we adopt Resnet architectures where we doubled the number of filters in each convolutional layer. As a baseline, we include pruning results with the classical OBD pruning algorithm (LeCun et al., 1990) for ResNet32 (train $\rightarrow$ prune $\rightarrow$ repeat). We compare our results against other algorithms that prune at initialization, such as SNIP (Lee et al., 2018), which is a SBP algorithm, GraSP (Wang et al., 2020) which is a Hessian based pruning algorithm, and SynFlow (Tanaka et al., 2020), which is an iterative data-agnostic pruning algorithm. As we increase the depth, SBP-SR starts to outperform other algorithms that prune at initialization (SBP-SR outperforms all other algorithms with ResNet104 on CIFAR10 and CIFAR100). Furthermore, using GraSP on Stable Resnet did not improve the result of GraSP on standard Resnet, as our proposed Stable Resnet analysis only applies to gradient based pruning. The analysis of Hessian based pruning could lead to similar techniques for improving trainability, which we leave for future work.

Table 3: Classification accuracies on Tiny ImageNet for Resnet with varying depths

| | ALGORITHM | 85% | 90% | 95% |
|---|---|---|---|---|
| RESNET32 | SBP-SR | **57.25 ± 0.09** | **55.67 ± 0.21** | 50.63±0.21 |
| | SNIP | 56.92± 0.33 | 54.99±0.37 | 49.48±0.48 |
| | GRASP | **57.25±0.11** | 55.53±0.11 | 51.34±0.29 |
| | SYNFLOW | 56.75±0.09 | 55.60±0.07 | **51.50±0.21** |
| RESNET50 | SBP-SR | **59.8±0.18** | **57.74±0.06** | **53.97±0.27** |
| | SNIP | 58.91±0.23 | 56.15±0.31 | 51.19±0.47 |
| | GRASP | 58.46±0.29 | 57.48±0.35 | 52.5±0.41 |
| | SYNFLOW | 59.31±0.17 | **57.67±0.15** | 53.14±0.31 |
| RESNET104 | SBP-SR | **62.84±0.13** | **61.96±0.11** | **57.9±0.31** |
| | SNIP | 59.94±0.34 | 58.14±0.28 | 54.9±0.42 |
| | GRASP | 61.1±0.41 | 60.14±0.38 | 56.36±0.51 |
| | SYNFLOW | 61.71±0.08 | 60.81±0.14 | 55.91±0.43 |

To confirm these results, we also test SBP-SR against other pruning algorithms on Tiny ImageNet. We train the models for 300 training epochs to make sure all algorithms converge. Table 3 shows test accuracies for SBP-SR, SNIP, GraSP, and SynFlow for $s \in \{85\%, 90\%, 95\%\}$. Although SynFlow competes or outperforms GraSP in many cases, SBP-SR has a clear advantage over SynFlow and other algorithms, especially for deep networks as illustrated on ResNet104.

Additional results with ImageNet dataset are provided in Appendix F.

### 4.3 RESCALING TRICK AND CNNS

The theoretical analysis of Section 2 is valid for Vanilla CNN i.e. CNN without pooling layers. With pooling layers, the theory of signal propagation applies to sections between successive pooling layers; each of those section can be seen as Vanilla CNN. This applies to standard CNN architectures such as VGG. As a *toy example*, we show in Table 4 the test accuracy of a pruned V-CNN with sparsity $s = 50\%$ on MNIST dataset. Similar to FFNN results in Figure 3, the combination of the EOC Init and the ReScaling trick allows for pruning deep V-CNN (depth 100) while ensuring their trainability.

Table 4: Test accuracy on MNIST with V-CNN for different depths with sparsity $50\%$ using SBP(SNIP)

| | $L = 10$ | $L = 50$ | $L = 100$ |
|---|---|---|---|
| ORDERED PHASE INIT | 98.12±0.13 | 10.00±0.0 | 10.00±0.0 |
| EOC INIT | 98.20±0.17 | 98.75±0.11 | 10.00±0.0 |
| EOC + RESCALING | 98.18±0.21 | 98.90±0.07 | 99.15±0.08 |

However, V-CNN is a toy example that is generally not used in practice. Standard CNN architectures such as VGG are popular among practitioners since they achieve SOTA accuracy on many tasks. Table 5 shows test accuracies for SNIP, SynFlow, and our EOC+ReScaling trick for VGG16 on CIFAR10. Our results are close to the results presented by Frankle et al. (2020). These three

algorithms perform similarly. From a theoretical point of view, our ReScaling trick applies to vanilla CNNs without pooling layers, hence, adding pooling layers might cause a deterioration. However, we know that the signal propagation theory applies to vanilla blocks inside VGG (i.e. the sequence of convolutional layers between two successive pooling layers). The larger those vanilla blocks are, the better our ReScaling trick performs. We leverage this observation by training a modified version of VGG, called 3xVGG16, which has the same number of pooling layers as VGG16, and 3 times the number of convolutional layers inside each vanilla block. Numerical results in Table 5 show that the EOC initialization with the ReScaling trick outperforms other algorithms, which confirms our hypothesis. However, the architecture 3xVGG16 is not a standard architecture and it does not seem to improve much the test accuracy of VGG16. An adaptation of the ReScaling trick to standard VGG architectures would be of great value and is left for future work.

Table 5: Classification accuracy on CIFAR10 for VGG16 and 3xVGG16 with varying sparsities

| | ALGORITHM | 85% | 90% | 95% |
|---|---|---|---|---|
| VGG16 | SNIP | 93.09±0.11 | 92.97±0.08 | 92.61±0.10 |
| | SYNFLOW | 93.21±0.13 | 93.05±0.11 | 92.19±0.12 |
| | EOC + RESCALING | 93.15±0.12 | 92.90±0.15 | 92.70±0.06 |
| 3XVGG16 | SNIP | 93.30±0.10 | 93.12±0.20 | 92.85±0.15 |
| | SYNFLOW | 92.95±0.13 | 92.91±0.21 | 92.70±0.20 |
| | EOC + RESCALING | **93.97±0.17** | **93.75±0.15** | **93.40±0.16** |

**Summary of numerical results.** We summarize in Table 6 our numerical results. The letter 'C' refers to 'Competition' between algorithms in that setting, and indicates no clear winner is found, while the dash means no experiment has been run with this setting. We observe that our algorithm SBP-SR consistently outperforms other algorithms in a variety of settings.

Table 6: Which algorithm performs better? (according to our results)

| DATASET | ARCHITECTURE | 85% | 90% | 95% | 98% |
|---|---|---|---|---|---|
| CIFAR10 | RESNET32 | - | C | C | GRASP |
| | RESNET50 | - | C | SBP-SR | GRASP |
| | RESNET104 | - | SBP-SR | SBP-SR | SBP-SR |
| | VGG16 | C | C | C | - |
| | 3XVGG16 | EOC+RESC | EOC+RESC | EOC+RESC | - |
| CIFAR100 | RESNET32 | - | SBP-SR | SBP-SR | SBP-SR |
| | RESNET50 | - | SBP-SR | SBP-SR | SBP-SR |
| | RESNET104 | - | SBP-SR | SBP-SR | SBP-SR |
| TINY IMAGENET | RESNET32 | C | C | SYNFLOW | - |
| | RESNET50 | SBP-SR | C | SBP-SR | - |
| | RESNET104 | SBP-SR | SBP-SR | SBP-SR | - |

## 5 CONCLUSION

In this paper, we have formulated principled guidelines for SBP at initialization. For FFNN and CNN, we have shown that an initialization on the EOC is necessary followed by the application of a simple rescaling trick to train the pruned network. For Resnets, the situation is markedly different. There is no need for a specific initialization but Resnets in their original form suffer from an exploding gradient problem. We propose an alternative Resnet parameterization called Stable Resnet, which allows for more stable pruning. Our theoretical results have been validated by extensive experiments on MNIST, CIFAR10, CIFAR100, Tiny ImageNet and ImageNet. Compared to other available one-shot pruning algorithms, we achieve state-of the-art results in many scenarios.

## REFERENCES


Alvarez, J. M. and M. Salzmann (2017). Compression-aware training of deep networks. In *31st Conference in Neural Information Processing Systems*, pp. 856–867.

Arora, S., S. Du, W. Hu, Z. Li, R. Salakhutdinov, and R. Wang (2019). On exact computation with an infinitely wide neural net. In *33rd Conference on Neural Information Processing Systems*.

Carreira-Perpiñán, M. and Y. Idelbayev (2018, June). Learning-compression algorithms for neural net pruning. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Dong, X., S. Chen, and S. Pan (2017). Learning to prune deep neural networks via layer-wise optimal brain surgeon. In *31st Conference on Neural Information Processing Systems*, pp. 4860–4874.

Du, S., X. Zhai, B. Poczos, and A. Singh (2019). Gradient descent provably optimizes over-parameterized neural networks. In *7th International Conference on Learning Representations*.

Frankle, J. and M. Carbin (2019). The lottery ticket hypothesis: Finding sparse, trainable neural networks. In *7th International Conference on Learning Representations*.

Frankle, J., G. Dziugaite, D. Roy, and M. Carbin (2020). Pruning neural networks at initialization: Why are we missing the mark? *arXiv preprint arXiv:2009.08576*.

Hassibi, B., D. Stork, and W. Gregory (1993). Optimal brain surgeon and general network pruning. In *IEEE International Conference on Neural Networks*, pp. 293 – 299 vol.1.

Hayou, S., E. Clerico, B. He, G. Deligiannidis, A. Doucet, and J. Rousseau (2021). Stable resnet. In *24th International Conference on Artificial Intelligence and Statistics*.

Hayou, S., A. Doucet, and J. Rousseau (2019). On the impact of the activation function on deep neural networks training. In *36th International Conference on Machine Learning*.

Hayou, S., A. Doucet, and J. Rousseau (2020). Mean-field behaviour of neural tangent kernel for deep neural networks. *arXiv preprint arXiv:1905.13654*.

He, K., X. Zhang, S. Ren, and J. Sun (2015). Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778.

Huang, G., Z. Liu, L. Maaten, and K. Weinberger (2017). Densely connected convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2261–2269.

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

Kolesnikov, A., L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby (2019). Large scale learning of general visual representations for transfer. *arXiv preprint arXiv:1912.11370*.

LeCun, Y., J. Denker, and S. Solla (1990). Optimal brain damage. In *Advances in Neural Information Processing Sstems*, pp. 598–605.

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

Lee, J., L. Xiao, S. Schoenholz, Y. Bahri, R. Novak, J. Sohl-Dickstein, and J. Pennington (2019). Wide neural networks of any depth evolve as linear models under gradient descent. In *33rd Conference on Neural Information Processing Systems*.

Lee, N., T. Ajanthan, S. Gould, and P. H. S. Torr (2020). A signal propagation perspective for pruning neural networks at initialization. In *8th International Conference on Learning Representations*.

Lee, N., T. Ajanthan, and P. H. Torr (2018). Snip: Single-shot network pruning based on connection sensitivity. In *6th International Conference on Learning Representations*.

Li, H., A. Kadav, I. Durdanovic, H. Samet, and H. Graf (2018). Pruning filters for efficient convnets. In *6th International Conference on Learning Representations*.

Li, Y., S. Gu, C. Mayer, L. V. Gool, and R. Timofte (2020). Group sparsity: The hinge between filter pruning and decomposition for network compression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 8018–8027.

Li, Y., S. Gu, K. Zhang, L. Van Gool, and R. Timofte (2020). Dhp: Differentiable meta pruning via hypernetworks. *arXiv preprint arXiv:2003.13683*.

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

Liu, Z., H. Mu, X. Zhang, Z. Guo, X. Yang, K.-T. Cheng, and J. Sun (2019). Metapruning: Meta learning for automatic neural network channel pruning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 3296–3305.

Louizos, C., M. Welling, and D. Kingma (2018). Learning sparse neural networks through $l_0$ regularization. In *6th International Conference on Learning Representations*.

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

Mozer, M. and P. Smolensky (1989). Skeletonization: A technique for trimming the fat from a network via relevance assessment. In *Advances in Neural Information Processing Systems*, pp. 107–115.

Neal, R. (1995). *Bayesian Learning for Neural Networks*, Volume 118. Springer Science & Business Media.

Neyshabur, B., Z. Li, S. Bhojanapalli, Y. LeCun, and N. Srebro (2019). The role of over-parametrization in generalization of neural networks. In *7th International Conference on Learning Representations*.

Nguyen, Q. and M. Hein (2018). Optimization landscape and expressivity of deep CNNs. In *35th International Conference on Machine Learning*.

Pečarić, J., F. Proschan, and Y. Tong (1992). *Convex Functions, Partial Orderings, and Statistical Applications*. Academic Press.

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

Puri, M. and S. Ralescu (1986). Limit theorems for random central order statistics. *Lecture Notes-Monograph Series Vol. 8, Adaptive Statistical Procedures and Related Topics*.

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

Tanaka, H., D. Kunin, D. L. Yamins, and S. Ganguli (2020). Pruning neural networks without any data by iteratively conserving synaptic flow. In *34th Conference on Neural Information Processing Systems*.

Wang, C., G. Zhang, and R. Grosse (2020). Picking winning tickets before training by preserving gradient flow. In *8th 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. In *35th International Conference on Machine Learning*.

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., J. Pennington, V. Rao, J. Sohl-Dickstein, and S. S. Schoenholz (2019). A mean field theory of batch normalization. In *7th International Conference on Learning Representations*.

Yang, G. and S. Schoenholz (2017). Mean field residual networks: On the edge of chaos. In *31st Conference in Neural Information Processing Systems*, Volume 30, pp. 2869–2869.

Zhang, C., S. Bengio, M. Hardt, B. Recht, and O. Vinyals (2016). Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations*.

# A Discussion about Approximations 1 and 2

## A.1 Approximation 1: Infinite Width Approximation

**FeedForward Neural Network**
Consider a randomly initialized FFNN of depth $L$, widths $(N_l)_{1\leq l\leq L}$, weights $W_{ij}^l \overset{iid}{\sim} \mathcal{N}(0, \frac{\sigma_w^2}{N_{l-1}})$ and bias $B_i^l \overset{iid}{\sim} \mathcal{N}(0, \sigma_b^2)$, where $\mathcal{N}(\mu, \sigma^2)$ denotes the normal distribution of mean $\mu$ and variance $\sigma^2$. For some input $x \in \mathbb{R}^d$, the propagation of this input through the network is given by

$$y_i^1(x) = \sum_{j=1}^{d} W_{ij}^1 x_j + B_i^1, \tag{10}$$

$$y_i^l(x) = \sum_{j=1}^{N_{l-1}} W_{ij}^l \phi(y_j^{l-1}(x)) + B_i^l, \quad \text{for } l \geq 2. \tag{11}$$

Where $\phi : \mathbb{R} \to \mathbb{R}$ is the activation function. When we take the limit $N_{l-1} \to \infty$, the Central Limit Theorem implies that $y_i^l(x)$ is a Gaussian variable for any input $x$. This approximation by infinite width solution results in an error of order $\mathcal{O}(1/\sqrt{N_{l-1}})$ (standard Monte Carlo error). More generally, an approximation of the random process $y_i^l(.)$ by a Gaussian process was first proposed by Neal (1995) in the single layer case and has been recently extended to the multiple layer case by Lee et al. (2018) and Matthews et al. (2018). We recall here the expressions of the limiting Gaussian process kernels. For any input $x \in \mathbb{R}^d$, $\mathbb{E}[y_i^l(x)] = 0$ so that for any inputs $x, x' \in \mathbb{R}^d$

$$\begin{aligned}\kappa^l(x, x') &= \mathbb{E}[y_i^l(x) y_i^l(x')] \\ &= \sigma_b^2 + \sigma_w^2 \mathbb{E}[\phi(y_i^{l-1}(x))\phi(y_i^{l-1}(x'))] \\ &= \sigma_b^2 + \sigma_w^2 F_\phi(\kappa^{l-1}(x, x), \kappa^{l-1}(x, x'), \kappa^{l-1}(x', x')),\end{aligned}$$

where $F_\phi$ 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**
Similar to the FFNN case, the infinite width approximation with 1D CNN (introduced in the main paper) yields a recursion for the kernel. However, the infinite width here means infinite number of channels, and results in an error $\mathcal{O}(1/\sqrt{n_{l-1}})$. The kernel in this case depends on the choice of the neurons in the channel and is given by

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

so that

$$\kappa_{\alpha,\alpha'}^l(x, x') = \sigma_b^2 + \frac{\sigma_w^2}{2k+1} \sum_{\beta \in ker} F_\phi(\kappa_{\alpha+\beta,\alpha'+\beta}^{l-1}(x, x), \kappa_{\alpha+\beta,\alpha'+\beta}^{l-1}(x, x'), \kappa_{\alpha+\beta,\alpha'+\beta}^{l-1}(x', x')).$$

The convolutional kernel $\kappa_{\alpha,\alpha'}^l$ 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 $\hat{\kappa}^l = \frac{1}{N^2} \sum_{\alpha,\alpha'} \kappa_{\alpha,\alpha'}^l$. Indeed, by summing terms in the previous equation and using the fact that we use circular padding, we obtain

$$\hat{\kappa}^l(x, x') = \sigma_b^2 + \sigma_w^2 \frac{1}{N^2} \sum_{\alpha,\alpha'} F_\phi(\kappa_{\alpha,\alpha'}^{l-1}(x, x), \kappa_{\alpha,\alpha'}^{l-1}(x, x'), \kappa_{\alpha,\alpha'}^{l-1}(x', x')).$$

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

Note that our analysis only requires the approximation that, in the infinite width limit, for any two inputs $x, x'$, the variables $y_i^l(x)$ and $y_i^l(x')$ are Gaussian with covariance $\kappa^l(x, x')$ for FFNN, and

$y^l_{i,\alpha}(x)$ and $y^l_{i,\alpha'}(x')$ are Gaussian with covariance $\kappa^l_{\alpha,\alpha'}(x,x')$ for CNN. We do not need the much stronger approximation that the process $y^l_i(x)$ ($y^l_{i,\alpha}(x)$ for CNN) is a Gaussian process.

**Residual Neural Networks**
The infinite width limit approximation for ResNet yields similar results with an additional residual terms. It is straighforward to see that, in the case 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

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

### A.2 Approximation 2: Gradient Independence

For gradient back-propagation, an essential assumption in prior literature in Mean-Field analysis of DNNs is that of the gradient independence which is similar in nature to the practice of feedback alignment (Lillicrap et al., 2016). This approximation allows for derivation of recursive formulas for gradient back-propagation, and it has been extensively used in literature and verified empirically; see references below.

**Gradient Covariance back-propagation:** this approximation was used to derive analytical formulas for gradient covariance back-propagation in (Hayou et al., 2019; Schoenholz et al., 2017; Yang and Schoenholz, 2017; Lee et al., 2018; Poole et al., 2016; Xiao et al., 2018; Yang, 2019). It was shown empirically through simulations that it is an excellent approximation for FFNN in Schoenholz et al. (2017), for Resnets in Yang and Schoenholz (2017) and for CNN in Xiao et al. (2018).

**Neural Tangent Kernel (NTK):** this approximation was implicitly used by Jacot et al. (2018) to derive the recursive formula of the infinite width Neural Tangent Kernel (See Jacot et al. (2018), Appendix A.1). Authors have found that this approximation yields excellent match with exact NTK. It was also exploited later in (Arora et al., 2019; Hayou et al., 2020) to derive the infinite NTK for different architectures. The difference between the infinite width NTK $\Theta$ and the empirical (exact) NTK $\hat{\Theta}$ was studied in Lee et al. (2019) where authors have shown that $\|\Theta - \hat{\Theta}\|_F = \mathcal{O}(N^{-1})$ where $N$ is the width of the NN.

More precisely, we use the approximation 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 gradient covariance and Neural Tangent Kernel, this approximation was proven to give the exact computation for standard architectures such as FFNN, CNN and ResNets, without BatchNorm in Yang (2019) (section D.5). Even with BatchNorm, in Yang et al. (2019), authors have found that the Gradient Independence approximation matches empirical results.

This approximation can be alternatively formulated as an assumption instead of an approximation as in Yang and Schoenholz (2017).
**Assumption 1 (Gradient Independence):** The gradients are computed using an i.i.d. version of the weights used for forward propagation.

## B Preliminary results

Let $x$ be an input in $\mathbb{R}^d$. In its general form, a neural network of depth $L$ is given by the following set of forward propagation equations

$$y^l(x) = \mathcal{F}_l(W^l, y^{l-1}(x)) + B^l, \quad 1 \leq l \leq L, \tag{12}$$

where $y^l(x)$ is the vector of pre-activations and $W^l$ and $B^l$ are respectively the weights and bias of the $l^{th}$ layer. $F_l$ is a mapping that defines the nature of the layer. The weights and bias are initialized with $W^l \overset{\text{iid}}{\sim} \mathcal{N}(0, \frac{\sigma_w^2}{v_l})$ where $v_l$ is a scaling factor used to control the variance of $y^l$, and $B^l \overset{\text{iid}}{\sim} \mathcal{N}(0, \sigma_b^2)$. Hereafter, we denote by $M_l$ the number of weights in the $l^{th}$ layer, $\phi$ the activation function and $[n:m]$ the set of integers $\{n, n+1, ..., m\}$ for $n \leq m$. Two examples of such architectures are:

- **Fully-connected FeedForward Neural Network (FFNN)**
  For a fully connected feedforward neural network of depth $L$ and widths $(N_l)_{l\leq L}$, the forward propagation of the input through the network is given by

$$\begin{aligned} y_i^1(x) &= \sum_{j=1}^{d} W_{ij}^1 x_j + B_i^1, \\ y_i^l(x) &= \sum_{j=1}^{N_{l-1}} W_{ij}^l \phi(y_j^{l-1}(x)) + B_i^l, \quad \text{for } l \geq 2. \end{aligned} \tag{13}$$

  Here, we have $v_l = N_{l-1}$ and $M_l = N_{l-1}N_l$.

- **Convolutional Neural Network (CNN/ConvNet)**
  For a 1D convolutional neural network of depth $L$, number of channels $(n_l)_{l\leq L}$ and number of neurons per channel $(N_l)_{l\leq L}$. we have

$$\begin{aligned} y_{i,\alpha}^1(x) &= \sum_{j=1}^{n_{l-1}} \sum_{\beta \in ker_l} W_{i,j,\beta}^1 x_{j,\alpha+\beta} + b_i^1, \\ y_{i,\alpha}^l(x) &= \sum_{j=1}^{n_{l-1}} \sum_{\beta \in ker_l} W_{i,j,\beta}^l \phi(y_{j,\alpha+\beta}^{l-1}(x)) + b_i^l, \ \text{ for } l \geq 2, \end{aligned} \tag{14}$$

  where $i \in [1 : n_l]$ is the channel index, $\alpha \in [0 : N_l - 1]$ is the neuron location, $ker_l = [-k_l : k_l]$ is the filter range and $2k_l + 1$ is the filter size. To simplify the analysis, we assume hereafter that $N_l = N$ and $k_l = k$ for all $l$. Here, we have $v_l = n_{l-1}(2k + 1)$ and $M_l = n_{l-1}n_l(2k + 1)$. We assume periodic boundary conditions, so $y_{i,\alpha}^l = y_{i,\alpha+N}^l = y_{i,\alpha-N}^l$. Generalization to multidimensional convolutions is straighforward.

**Notation:** Hereafter, for FFNN layers, we denote by $q^l(x)$ the variance of $y_1^l(x)$ (the choice of the index 1 is not crucial since, by the mean-field approximation, the random variables $(y_i^l(x))_{i\in[1:N_l]}$ are iid Gaussian variables). We denote by $q^l(x, x')$ the covariance between $y_1^l(x)$ and $y_1^l(x')$, and $c_1^l(x, x')$ the corresponding correlation. For gradient back-propagation, for some loss function $\mathcal{L}$, we denote by $\tilde{q}^l(x, x')$ the gradient covariance defined by $\tilde{q}^l(x, x') = \mathbb{E}\left[\frac{\partial \mathcal{L}}{\partial y_1^l}(x)\frac{\partial \mathcal{L}}{\partial y_1^l}(x')\right]$. Similarly, $\tilde{q}^l(x)$ denotes the gradient variance at point $x$.
For CNN layers, we use similar notation accross channels. More precisely, we denote by $q_\alpha^l(x)$ the variance of $y_{1,\alpha,}^l(x)$ (the choice of the index 1 is not crucial here either since, by the mean-field approximation, the random variables $(y_{i,\alpha}^l(x))_{i\in[1:N_l]}$ are iid Gaussian variables). We denote by $q_{\alpha,\alpha'}^l(x, x')$ the covariance between $y_{1,\alpha}^l(x)$ and $y_{1,\alpha'}^l(x')$, and $c_{\alpha,\alpha'}^l(x, x')$ the corresponding correlation.
As in the FFNN case, we define the gradient covariance by $\tilde{q}_{\alpha,\alpha'}^l(x, x') = \mathbb{E}\left[\frac{\partial \mathcal{L}}{\partial y_{1,\alpha}^l}(x)\frac{\partial \mathcal{L}}{\partial y_{1,\alpha'}^l}(x')\right]$.

### B.1 Warmup : Some results from the Mean-Field theory of DNNs

We start by recalling some results from the mean-field theory of deep NNs.

#### B.1.1 Covariance propagation

**Covariance propagation for FFNN:**
In Section A.1, we presented the recursive formula for covariance propagation in a FFNN, which we derive using the Central Limit Theorem. More precisely, for two inputs $x, x' \in \mathbb{R}^d$, we have

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

This can be rewritten as

$$q^l(x, x') = \sigma_b^2 + \sigma_w^2 \mathbb{E}\left[\phi\left(\sqrt{q^l(x)}Z_1\right)\phi\left(\sqrt{q^l(x')}(c^{l-1}Z_1 + \sqrt{1-(c^{l-1})^2}Z_2\right)\right],$$

where $c^{l-1} := c^{l-1}(x, x')$.
With a ReLU activation function, we have

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

where $f$ is the ReLU correlation function given by (Hayou et al. (2019))

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

**Covariance propagation for CNN:**
Similar to the FFNN case, it is straightforward to derive recusive formula for the covariance. However, in this case, the independence is across channels and not neurons. Simple calculus yields

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

Using a ReLU activation function, this becomes

$$q^l_{\alpha,\alpha'}(x, x') = \sigma_b^2 + \frac{\sigma_w^2}{2k+1}\sum_{\beta\in ker}\sqrt{q^l_{\alpha+\beta}(x)}\sqrt{q^l_{\alpha'+\beta}(x')}f(c^{l-1}_{\alpha+\beta,\alpha'+\beta}(x, x')).$$

**Covariance propagation for ResNet with ReLU :**
This case is similar to the non residual case. However, an added residual term shows up in the recursive formula. For ResNet with FFNN layers, we have

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

and for ResNet with CNN layers, we have

$$q^l_{\alpha,\alpha'}(x, x') = q^{l-1}_{\alpha,\alpha'}(x, x') + \sigma_b^2 + \frac{\sigma_w^2}{2k+1}\sum_{\beta\in ker}\sqrt{q^l_{\alpha+\beta}(x)}\sqrt{q^l_{\alpha'+\beta}(x')}f(c^{l-1}_{\alpha+\beta,\alpha'+\beta}(x, x')).$$

### B.1.2 Gradient Covariance back-propagation

**Gradiant Covariance back-propagation for FFNN:**
Let $\mathcal{L}$ be the loss function. Let $x$ be an input. The back-propagation of the gradient is given by the set of equations

$$\frac{\partial \mathcal{L}}{\partial y^l_i} = \phi'(y^l_i)\sum_{j=1}^{N_{l+1}}\frac{\partial \mathcal{L}}{\partial y^{l+1}_j}W^{l+1}_{ji}.$$

Using the approximation that the weights used for forward propagation are independent from those used in backpropagation, we have as in Schoenholz et al. (2017)

$$\tilde{q}^l(x) = \tilde{q}^{l+1}(x)\frac{N_{l+1}}{N_l}\chi(q^l(x)),$$

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

**Gradient Covariance back-propagation for CNN:**
Similar to the FFNN case, we have that

$$\frac{\partial \mathcal{L}}{\partial W^l_{i,j,\beta}} = \sum_\alpha \frac{\partial \mathcal{L}}{\partial y^l_{i,\alpha}}\phi(y^{l-1}_{j,\alpha+\beta})$$

and

$$\frac{\partial \mathcal{L}}{\partial y^l_{i,\alpha}} = \sum_{j=1}^n\sum_{\beta\in ker}\frac{\partial \mathcal{L}}{\partial y^{l+1}_{j,\alpha-\beta}}W^{l+1}_{i,j,\beta}\phi'(y^l_{i,\alpha}).$$

Using the approximation of Gradient independence and averaging over the number of channels (using CLT) we have that

$$\mathbb{E}[\frac{\partial \mathcal{L}}{\partial y^l_{i,\alpha}}^2] = \frac{\sigma_w^2 \mathbb{E}[\phi'(\sqrt{q^l_\alpha(x)}Z)^2]}{2k+1} \sum_{\beta \in ker} \mathbb{E}[\frac{\partial \mathcal{L}}{\partial y^{l+1}_{i,\alpha-\beta}}^2].$$

We can get similar recursion to that of the FFNN case by summing over $\alpha$ and using the periodic boundary condition, this yields

$$\sum_\alpha \mathbb{E}[\frac{\partial \mathcal{L}}{\partial y^l_{i,\alpha}}^2] = \chi(q^l_\alpha(x)) \sum_\alpha \mathbb{E}[\frac{\partial \mathcal{L}}{\partial y^{l+1}_{i,\alpha}}^2].$$

### B.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 $c^l(x, x')$ between $y^l(x)$ and $y^l(x')$ for any two inputs $x$ and $x'$ is also driven by $(\sigma_b, \sigma_w)$: 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 $\sigma_w^2 \mathbb{E}[\phi'(\sqrt{q(\sigma_b, \sigma_w)}Z)^2] = 1$ where $Z \sim \mathcal{N}(0, 1)$. Similarly the Ordered, resp. Chaotic, phase is defined by $\sigma_w^2 \mathbb{E}[\phi'(\sqrt{q(\sigma_b, \sigma_w)}Z)^2] < 1$, resp. $\sigma_w^2 \mathbb{E}[\phi'(\sqrt{q(\sigma_b, \sigma_w)}Z)^2] > 1$. On the Ordered phase, the gradient will vanish as it backpropagates through the network, and the correlation $c^l(x, x')$ converges exponentially to 1. 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**.

### B.1.4 Some results from the Mean-Field theory of Deep FFNNs

Let $\epsilon \in (0, 1)$ and $B_\epsilon = \{(x, x') \in (\mathbb{R}^d)^2 : c^1(x, x') < 1 - \epsilon\}$ (For now $B_\epsilon$ is defined only for FFNN).

Using Approximation 1, the following results have been derived by Schoenholz et al. (2017) and Hayou et al. (2019):

- There exist $q, \lambda > 0$ such that $\sup_{x \in \mathbb{R}^d} |q^l(x) - q| \leq e^{-\lambda l}$.
- On the Ordered phase, there exists $\gamma > 0$ such that $\sup_{x, x' \in \mathbb{R}^d} |c^l(x, x') - 1| \leq e^{-\gamma l}$.
- On the Chaotic phase, For all $\epsilon \in (0, 1)$ there exist $\gamma > 0$ and $c < 1$ such that $\sup_{(x,x') \in B_\epsilon} |c^l(x, x') - c| \leq e^{-\gamma l}$.
- For ReLU network on the EOC, we have

$$f(x) \underset{x \to 1-}{=} x + \frac{2\sqrt{2}}{3\pi}(1-x)^{3/2} + O((1-x)^{5/2}).$$

- In general, we have

$$f(x) = \frac{\sigma_b^2 + \sigma_w^2 \mathbb{E}[\phi(\sqrt{q}Z_1)\phi(\sqrt{q}Z(x))]}{q}, \tag{15}$$

  where $Z(x) = xZ_1 + \sqrt{1-x^2}Z_2$ and $Z_1, Z_2$ are iid standard Gaussian variables.
- On the EOC, we have $f'(1) = 1$
- On the Ordered, resp. Chaotic, phase we have that $f'(1) < 1$, resp. $f'(1) > 1$.
- For non-linear activation functions, $f$ is strictly convex and $f(1) = 1$.
- $f$ is increasing on $[-1, 1]$.

- On the Ordered phase and EOC, $f$ has one fixed point which is 1. On the chaotic phase, $f$ has two fixed points: 1 which is unstable, and $c \in (0,1)$ which is a stable fixed point.
- On the Ordered/Chaotic phase, the correlation between gradients computed with different inputs converges exponentially to 0 as we back-progapagate the gradients.

Similar results exist for CNN. Xiao et al. (2018) show that, similarly to the FFNN case, there exists $q$ such that $q^l_\alpha(x)$ converges exponentially to $q$ for all $x, \alpha$, and studied the limiting behaviour of correlation between neurons at the same channel $c^l_{\alpha,\alpha'}(x,x)$ (same input $x$). These correlations describe how features are correlated for the same input. However, they do not capture the behaviour of these features for different inputs (i.e. $c^l_{\alpha,\alpha'}(x,x')$ where $x \neq x'$). We establish this result in the next section.

### B.2 Correlation behaviour in CNN in the limit of large depth

**Appendix Lemma 1** (Asymptotic behaviour of the correlation in CNN with smooth activation functions)**.** *We consider a 1D CNN. Let $(\sigma_b, \sigma_w) \in (\mathbb{R}^+)^2$ and $x \neq x'$ be two inputs $\in \mathbb{R}^d$. If $(\sigma_b, \sigma_w)$ are either on the Ordered or Chaotic phase, then there exists $\beta > 0$ such that*

$$\sup_{\alpha,\alpha'} |c^l_{\alpha,\alpha'}(x,x') - c| = \mathcal{O}(e^{-\beta l}),$$

*where $c = 1$ if $(\sigma_b, \sigma_w)$ is in the Ordered phase, and $c \in (0,1)$ if $(\sigma_b, \sigma_w)$ is in the Chaotic phase.*

*Proof.* Let $x \neq x'$ be two inputs and $\alpha, \alpha'$ two nodes in the same channel $i$. From Section B.1, we have that

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

This yields

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

where $f$ is the correlation function.
We prove the result in the Ordered phase, the proof in the Chaotic phase is similar. Let $(\sigma_b, \sigma_w)$ be in the Ordered phase and $c^l_m = \min_{\alpha,\alpha'} c^l_{\alpha,\alpha'}(x,x')$. Using the fact that $f$ is non-decreasing (section B.1), we have that $c^l_{\alpha,\alpha'}(x,x') \geq \frac{1}{2k+1} \sum_{\beta \in ker} c^{l-1}_{\alpha+\beta,\alpha'+\beta}(x,x')) \geq f(c^{l-1}_m)$. Taking the min again over $\alpha, \alpha'$, we have $c^l_m \geq f(c^{l-1}_m)$, therefore $c^l_m$ is non-decreasing and converges to a stable fixed point of $f$. By the convexity of $f$, the limit is 1 (in the Chaotic phase, $f$ has two fixed point, a stable point $c_1 < 1$ and $c_2 = 1$ unstable). Moreover, the convergence is exponential using the fact that $0 < f'(1) < 1$. We conclude using the fact that $\sup_{\alpha,\alpha'} |c^l_{\alpha,\alpha'}(x,x') - 1| = 1 - c^l_m$. □

## C Proofs for Section 2 : SBP for FFNN/CNN and the Rescaling Trick

In this section, we prove Theorem 1 and Proposition 1. Before proving Theorem 1, we state the degeneracy approximation.

**Approximation 3** (Degeneracy on the Ordered phase)**.** *On the Ordered phase, the correlation $c^l$ and the variance $q^l$ converge exponentially quickly to their limiting values 1 and $q$ respectively. The degeneracy approximation for FFNN states that*

- $\forall x \neq x', c^l(x,x') \approx 1$
- $\forall x, q^l(x) \approx q$

*For CNN,*

- $\forall x \neq x', \alpha, \alpha', c^l_{\alpha,\alpha'}(x,x') \approx 1$

- $\forall x, q_\alpha^l(x) \approx q$

The degeneracy approximation is essential in the proof of Theorem 1 as it allows us to avoid many unnecessary complications. However, the results hold without this approximation although the constants are different.

**Theorem 1** (Initialization is crucial for SBP)**.** *We consider a FFNN (2) or a CNN (3). Assume $(\sigma_w, \sigma_b)$ are chosen on the ordered phase, i.e. $\chi(\sigma_b, \sigma_w) < 1$, then the NN is ill-conditioned. Moreover, we have*

$$\mathbb{E}[s_{cr}] \leq \frac{1}{L}\left(1 + \frac{\log(\kappa L N^2)}{\kappa}\right) + \mathcal{O}\left(\frac{1}{\kappa^2\sqrt{LN^2}}\right),$$

*where $\kappa = |\log \chi(\sigma_b, \sigma_w)|/8$. If $(\sigma_w, \sigma_b)$ are on the EOC, i.e. $\chi(\sigma_b, \sigma_w) = 1$, then the NN is well-conditioned. In this case, $\kappa = 0$ and the above upper bound no longer holds.*

*Proof.* We prove the result using Approximation 3.

1. **Case 1 : Fully connected Feedforward Neural Networks**
   To simplify the notation, we assume that $N_l = N$ and $M_l = N^2$ (i.e. $\alpha_l = 1$ and $\zeta_l = 1$) for all $l$. We prove the result for the Ordered phase, the proof for the Chaotic phase is similar. Let $L_0 \gg 1$, $\epsilon \in (0, 1 - \frac{1}{L_0})$, $L \geq L_0$ and $x \in (\frac{1}{L} + \epsilon, 1)$. With sparsity $x$, we keep $k_x = \lfloor (1-x)LN^2 \rfloor$ weights. We have

   $$\mathbb{P}(s_{cr} \leq x) \geq \mathbb{P}(\max_{i,j} |W_{ij}^1| \Big| \frac{\partial \mathcal{L}}{\partial W_{ij}^1}\Big| < t^{(k_x)})$$

   where $t^{(k_x)}$ is the $k_x^{th}$ order statistic of the sequence $\{|W_{ij}^l| \Big| \frac{\partial \mathcal{L}}{\partial W_{ij}^l}\Big|, l > 0, (i,j) \in [1:N]^2\}$.

   We have

   $$\begin{aligned}\frac{\partial \mathcal{L}}{\partial W_{ij}^l} &= \frac{1}{|\mathcal{D}|}\sum_{x\in\mathcal{D}} \frac{\partial \mathcal{L}}{\partial y_i^l(x)} \frac{\partial y_i^l(x)}{\partial W_{ij}^l} \\ &= \frac{1}{|\mathcal{D}|}\sum_{x\in\mathcal{D}} \frac{\partial \mathcal{L}}{\partial y_i^l(x)} \phi(y_j^{l-1}(x)).\end{aligned}$$

   On the Ordered phase, the variance $q^l(x)$ and the correlation $c^l(x, x')$ converge exponentially to their limiting values $q$, 1 (Section B.1). Under the degeneracy Approximation 3, we have

   - $\forall x \neq x', c^l(x, x') \approx 1$
   - $\forall x, q^l(x) \approx q$

   Let $\tilde{q}^l(x) = \mathbb{E}[\frac{\partial \mathcal{L}}{\partial y_i^l(x)}^2]$ (the choice of $i$ is not important since $(y_i^l(x))_i$ are iid ). Using these approximations, we have that $y_i^l(x) = y_i^l(x')$ almost surely for all $x, x'$. Thus

   $$\mathbb{E}\Big[\frac{\partial \mathcal{L}}{\partial W_{ij}^l}^2\Big] = \mathbb{E}[\phi(\sqrt{q}Z)^2]\tilde{q}^l(x),$$

   where $x$ is an input. The choice of $x$ is not important in our approximation.
   From Section B.1.2, we have

   $$\tilde{q}^l(x) = \tilde{q}^{l+1}(x)\frac{N_{l+1}}{N_l}\chi.$$

   Then we obtain

   $$\tilde{q}^l(x) = \frac{N_L}{N_l}\tilde{q}^L(x)\chi^{L-l} = \tilde{q}^L(x)\chi^{L-l},$$

   where $\chi = \sigma_w^2 \mathbb{E}[\phi(\sqrt{q}Z)^2]$ as we have assumed $N_l = N$. Using this result, we have

   $$\mathbb{E}\Big[\frac{\partial \mathcal{L}}{\partial W_{ij}^l}^2\Big] = A\,\chi^{L-l},$$

where $A = \mathbb{E}[\phi(\sqrt{q}Z)^2]\tilde{q}_x^L$ for an input $x$. Recall that by definition, one has $\chi < 1$ on the Ordered phase.

In the general case, i.e. without the degeneracy approximation on $c^l$ and $q^l$, we can prove that

$$\mathbb{E}\big[\frac{\partial \mathcal{L}}{\partial W_{ij}^l}^2\big] = \Theta(\chi^{L-l})$$

which suffices for the rest of the proof. However, the proof of this result requires many unnecessary complications that do not add any intuitive value to the proof.

In the general case where the widths are different, $\tilde{q}^l$ will also scale as $\chi^{L-l}$ up to a different constant.
Now we want to lower bound the probability

$$\mathbb{P}(\max_{i,j} |W_{ij}^1| \big| \frac{\partial \mathcal{L}}{\partial W_{ij}^1}\big| < t^{(k_x)}).$$

Let $t_\epsilon^{(k_x)}$ be the $k_x^{\text{th}}$ order statistic of the sequence $\{|W_{ij}^l| \big| \frac{\partial \mathcal{L}}{\partial W_{ij}^l}\big|, l > 1 + \epsilon L, (i,j) \in [1:N]^2\}$. It is clear that $t^{(k_x)} > t_\epsilon^{(k_x)}$, therefore

$$\mathbb{P}(\max_{i,j} |W_{ij}^1| \big| \frac{\partial \mathcal{L}}{\partial W_{ij}^1}\big| < t^{(k_x)}) \geq \mathbb{P}(\max_{i,j} |W_{ij}^1| \big| \frac{\partial \mathcal{L}}{\partial W_{ij}^1}\big| < t_\epsilon^{(k_x)}).$$

Using Markov's inequality, we have that

$$\mathbb{P}(\big| \frac{\partial \mathcal{L}}{\partial W_{ij}^1}\big| \geq \alpha) \leq \frac{\mathbb{E}\big[\big| \frac{\partial \mathcal{L}}{\partial W_{ij}^1}\big|^2\big]}{\alpha^2}. \tag{16}$$

Note that $\text{Var}(\chi^{\frac{l-L}{2}} \big| \frac{\partial \mathcal{L}}{\partial W_{ij}^l}\big|) = A$. In general, the random variables $\chi^{\frac{l-L}{2}} \big| \frac{\partial \mathcal{L}}{\partial W_{ij}^l}\big|$ have a density $f_{ij}^l$ for all $l > 1 + \epsilon L, (i,j) \in [1:N]^2$, such that $f_{ij}^l(0) \neq 0$. Therefore, there exists a constant $\lambda$ such that for $x$ small enough,

$$\mathbb{P}(\chi^{\frac{l-L}{2}} \big| \frac{\partial \mathcal{L}}{\partial W_{ij}^l}\big| \geq x) \geq 1 - \lambda x.$$

By selecting $x = \chi^{\frac{(1-\epsilon/2)L-1}{2}}$, we obtain

$$\chi^{\frac{l-L}{2}} \times x \leq \chi^{\frac{(1+\epsilon L)-L}{2}} \chi^{\frac{(1-\epsilon/2)L-1}{2}} = \chi^{\epsilon L/2}.$$

Therefore, for $L$ large enough, and all $l > 1 + \epsilon L$, $(i,j) \in [1:N_l] \times [1:N_{l-1}] = [1:N]^2$, we have

$$\mathbb{P}(\big| \frac{\partial \mathcal{L}}{\partial W_{ij}^l}\big| \geq \chi^{\frac{(1-\epsilon/2)L-1}{2}}) \geq 1 - \lambda\, \chi^{\frac{l-(\epsilon L/2+1)}{2}} \geq 1 - \lambda\, \chi^{\epsilon L/2}.$$

Now choosing $\alpha = \chi^{\frac{(1-\epsilon/4)L-1}{2}}$ in inequality (16) yields

$$\mathbb{P}(\big| \frac{\partial \mathcal{L}}{\partial W_{ij}^1}\big| \geq \chi^{\frac{(1-\epsilon/4)L-1}{2}}) \geq 1 - A\, \chi^{\epsilon L/4}.$$

Since we do not know the exact distribution of the gradients, the trick is to bound them using the previous concentration inequalities. We define the event $B := \{\forall (i,j) \in [1:N] \times [1:d], \big| \frac{\partial \mathcal{L}}{\partial W_{ij}^1}\big| \leq \chi^{\frac{(1-\epsilon/4)L-1}{2}}\} \cap \{\forall l > 1 + \epsilon L, (i,j) \in [1:N]^2, \big| \frac{\partial \mathcal{L}}{\partial W_{ij}^l}\big| \geq \chi^{\frac{(1-\epsilon/2)L-1}{2}}\}$.

We have

$$\mathbb{P}(\max_{i,j} |W_{ij}^1| \big| \frac{\partial \mathcal{L}}{\partial W_{ij}^1}\big| < t_\epsilon^{(k_x)}) \geq \mathbb{P}(\max_{i,j} |W_{ij}^1| \big| \frac{\partial \mathcal{L}}{\partial W_{ij}^1}\big| < t_\epsilon^{(k_x)} \big| B)\mathbb{P}(B).$$

But, by conditioning on the event $B$, we also have

$$\mathbb{P}(\max_{i,j}|W_{ij}^1||\frac{\partial \mathcal{L}}{\partial W_{ij}^1}| < t_\epsilon^{(k_x)}|B) \geq \mathbb{P}(\max_{i,j}|W_{ij}^1| < \chi^{-\epsilon L/8} t_\epsilon'^{(k_x)}),$$

where $t_\epsilon'^{(k_x)}$ is the $k_x^{\text{th}}$ order statistic of the sequence $\{|W_{ij}^l|, l > 1+\epsilon L, (i,j) \in [1:N]^2\}$.

Now, as in the proof of Proposition 4 in Appendix E (MBP section), define $x_{\zeta,\gamma_L} = \min\{y \in (0,1) : \forall x > y, \gamma_L Q_x > Q_{1-(1-x)^{\gamma_L^{2-\zeta}}}\}$, where $\gamma_L = \chi^{-\epsilon L/8}$. Since $\lim_{\zeta \to 2} x_{\zeta,\gamma_L} = 0$, then there exists $\zeta_\epsilon < 2$ such that $x_{\zeta_\epsilon,\gamma_L} = \epsilon + \frac{1}{L}$.

As $L$ grows, $t_\epsilon'^{(k_x)}$ converges to the quantile of order $\frac{x-\epsilon}{1-\epsilon}$. Therefore, using classic Berry-Essen bounds on the cumulative distribution function, it is easy to obtain

$$\begin{aligned}\mathbb{P}(\max_{i,j}|W_{ij}^1| < \chi^{-\epsilon L/8} t_\epsilon'^{(k_x)}) &\geq \mathbb{P}(\max_{i,j}|W_{ij}^1| < Q_{1-(1-\frac{x-\epsilon}{1-\epsilon})^{\gamma_L^{2-\zeta_\epsilon}}}) + \mathcal{O}(\frac{1}{\sqrt{LN^2}}) \\ &\geq 1 - N^2(\frac{x-\epsilon}{1-\epsilon})^{\gamma_L^{2-\zeta_\epsilon}} + \mathcal{O}(\frac{1}{\sqrt{LN^2}}).\end{aligned}$$

Using the above concentration inequalities on the gradient, we obtain

$$\mathbb{P}(B) \geq (1 - A\,\chi^{\epsilon L/4})^{N^2}(1 - \lambda\,\chi^{\epsilon L/2})^{LN^2}.$$

Therefore there exists a constant $\eta > 0$ independent of $\epsilon$ such that

$$\mathbb{P}(B) \geq 1 - \eta L N^2 \chi^{\epsilon L/4}.$$

Hence, we obtain

$$\mathbb{P}(s_{cr} \geq x) \leq N^2(\frac{x-\epsilon}{1-\epsilon})^{\gamma_L^{2-\zeta_\epsilon}} + \eta L N^2 \chi^{\epsilon L/4} + \mathcal{O}(\frac{1}{\sqrt{LN^2}}).$$

Integration of the previous inequality yields

$$\mathbb{E}[s_{cr}] \leq \epsilon + \frac{1}{L} + \frac{N^2}{1+\gamma_L^{2-\zeta_\epsilon}} + \eta L N^2 \chi^{\epsilon L/4} + \mathcal{O}(\frac{1}{\sqrt{LN^2}}).$$

Now let $\kappa = \frac{|\log(\chi)|}{8}$ and set $\epsilon = \frac{\log(\kappa L N^2)}{\kappa L}$. By the definition of $x_{\zeta_\epsilon,\gamma_L}$, we have

$$\gamma_L Q_{x_{\zeta_\epsilon,\gamma_L}} = Q_{1-(1-x_{\zeta_\epsilon,\gamma_L})^{\gamma_L^{2-\zeta_\epsilon}}}.$$

For the left hand side, we have

$$\gamma_L Q_{x_{\zeta_\epsilon,\gamma_L}} \sim \alpha \gamma_L \frac{\log(\kappa L N^2)}{\kappa L}$$

where $\alpha > 0$ is the derivative at 0 of the function $x \to Q_x$ (the derivative at zero of the cdf of $|\mathcal{N}(0,1)|$ is positive). Since $\gamma_L = \kappa L N^2$, we have

$$\gamma_L Q_{x_{\zeta_\epsilon,\gamma_L}} \sim \alpha N^2 \log(\kappa L N^2)$$

Which diverges as $L$ goes to infinity. In particular this proves that the right hand side diverges and therefore we have that $(1 - x_{\zeta_\epsilon,\gamma_L})^{\gamma_L^{2-\zeta_\epsilon}}$ converges to 0 as $L$ goes to infinity. Using the asymptotic equivalent of the right hand side as $L \to \infty$, we have $Q_{1-(1-x_{\zeta_\epsilon,\gamma_L})^{\gamma_L^{2-\zeta_\epsilon}}} \sim \sqrt{-2\log((1-x_{\zeta_\epsilon,\gamma_L})^{\gamma_L^{2-\zeta_\epsilon}})} = \gamma_L^{1-\zeta_\epsilon/2}\sqrt{-2\log(1-x_{\zeta_\epsilon,\gamma_L})}$. Therefore, we obtain

$$Q_{1-(1-x_{\zeta_\epsilon,\gamma_L})^{\gamma_L^{2-\zeta_\epsilon}}} \sim \gamma_L^{1-\zeta_\epsilon/2}\sqrt{\frac{2\log(\kappa L N^2)}{\kappa L}}.$$

Combining this result to the fact that $\gamma_L Q_{x_{\zeta_\epsilon},\gamma_L} \sim \alpha\gamma_L \frac{\log(\kappa LN^2)}{\kappa L}$ we obtain

$$\gamma_L^{-\zeta_\epsilon} \sim \beta \frac{\log(\kappa LN^2)}{\kappa L},$$

where $\beta$ is a positive constant. This yields

$$\begin{aligned}\mathbb{E}[s_{cr}] &\leq \frac{\log(\kappa LN^2)}{\kappa L} + \frac{1}{L} + \frac{\mu}{\kappa LN^2 \log(\kappa LN^2)}(1+o(1)) + \eta\frac{1}{\kappa^2 LN^2} + \mathcal{O}(\frac{1}{\sqrt{LN^2}}) \\ &= \frac{1}{L}(1 + \frac{\log(\kappa LN^2)}{\kappa}) + \mathcal{O}(\frac{1}{\kappa^2\sqrt{LN^2}}),\end{aligned}$$

where $\kappa = \frac{|\log(\chi)|}{8}$ and $\mu$ is a constant.

2. **Case 2 : Convolutional Neural Networks**
The proof for CNNs is similar to that of FFNN once we prove that

$$\mathbb{E}[\frac{\partial \mathcal{L}}{\partial W^l_{i,j,\beta}}^2] = A\,\chi^{L-l}$$

where $A$ is a constant. We have that

$$\frac{\partial \mathcal{L}}{\partial W^l_{i,j,\beta}} = \sum_\alpha \frac{\partial \mathcal{L}}{\partial y^l_{i,\alpha}} \phi(y^{l-1}_{j,\alpha+\beta})$$

and

$$\frac{\partial \mathcal{L}}{\partial y^l_{i,\alpha}} = \sum_{j=1}^{n} \sum_{\beta \in ker} \frac{\partial \mathcal{L}}{\partial y^{l+1}_{j,\alpha-\beta}} W^{l+1}_{i,j,\beta} \phi'(y^l_{i,\alpha}).$$

Using the approximation of Gradient independence and averaging over the number of channels (using CLT) we have that

$$\mathbb{E}[\frac{\partial \mathcal{L}}{\partial y^l_{i,\alpha}}^2] = \frac{\sigma_w^2 \mathbb{E}[\phi'(\sqrt{q}Z)^2]}{2k+1} \sum_{\beta \in ker} \mathbb{E}[\frac{\partial \mathcal{L}}{\partial y^{l+1}_{i,\alpha-\beta}}^2].$$

Summing over $\alpha$ and using the periodic boundary condition, this yields

$$\sum_\alpha \mathbb{E}[\frac{\partial \mathcal{L}}{\partial y^l_{i,\alpha}}^2] = \chi \sum_\alpha \mathbb{E}[\frac{\partial \mathcal{L}}{\partial y^{l+1}_{i,\alpha}}^2].$$

Here also, on the Ordered phase, the variance $q^l$ and the correlation $c^l$ converge exponentially to their limiting values $q$ and 1 respectively. As for FFNN, we use the degeneracy approximation that states

- $\forall x \neq x', \alpha, \alpha', c^l_{\alpha,\alpha'}(x,x') \approx 1,$
- $\forall x, q^l_\alpha(x) \approx q.$

Using these approximations, we have

$$\mathbb{E}[\frac{\partial \mathcal{L}}{\partial W^l_{i,j,\beta}}^2] = \mathbb{E}[\phi(\sqrt{q}Z)^2]\tilde{q}^l(x),$$

where $\tilde{q}^l(x) = \sum_\alpha \mathbb{E}[\frac{\partial \mathcal{L}}{\partial y^l_{i,\alpha}(x)}^2]$ for an input $x$. The choice of $x$ is not important in our approximation.

From the analysis above, we have

$$\tilde{q}^l(x) = \tilde{q}^L(x)\chi^{L-l},$$

so we conclude that

$$\mathbb{E}\Big[\frac{\partial \mathcal{L}}{\partial W^l_{i,j,\beta}}^2\Big] = A\,\chi^{L-l}$$

where $A = \mathbb{E}[\phi(\sqrt{q}Z)^2]\tilde{q}^L(x)$.

With EOC initialization, classic results from Schoenholz et al. (2017); Hayou et al. (2019) show that the second moment of the gradient is the same for all layers. It follows that $m_l$ is the same for all layers up to some constants that do not depend on $L, l$. This concludes the proof. □

After pruning, the network is usually 'deep' in the Ordered phase in the sense that $\chi = f'(1) \ll 1$. To re-place it on the Edge of Chaos, we use the Rescaling Trick.

**Proposition 1** (Rescaling Trick)**.** *Consider a NN of the form (2) or (3) (FFNN or CNN) initialized on the EOC. Then, after pruning, the sparse network is not initialized on the EOC. However, the rescaled sparse network*

$$y^l(x) = \mathcal{F}(\rho^l \circ \delta^l \circ W^l, y^{l-1}(x)) + B^l, \quad \text{for } l \geq 1, \tag{17}$$

*where*

- $\rho^l_{ij} = \frac{1}{\sqrt{\mathbb{E}[N_{l-1}(W^l_{i1})^2\delta^l_{i1}]}}$ *for FFNN of the form (2),*
- $\rho^l_{i,j,\beta} = \frac{1}{\sqrt{\mathbb{E}[n_{l-1}(W^l_{i,1,\beta})^2\delta^l_{i,1,\beta}]}}$ *for CNN of the form (3),*

*is initialized on the EOC.*

*Proof.* For two inputs $x, x'$, the forward propagation of the covariance is given by

$$\begin{aligned}\hat{q}^l(x,x') &= \mathbb{E}[y^l_i(x)y^l_i(x')] \\ &= \mathbb{E}[\sum_{j,k}^{N_{l-1}} W^l_{ij}W^l_{ik}\delta^l_{ij}\delta^l_{ik}\phi(\hat{y}^{l-1}_j(x))\phi(\hat{y}^{l-1}_j(x'))] + \sigma_b^2.\end{aligned}$$

We have

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial W^l_{ij}} &= \frac{1}{|\mathcal{D}|}\sum_{x\in\mathcal{D}} \frac{\partial \mathcal{L}}{\partial y^l_i(x)}\frac{\partial y^l_i(x)}{\partial W^l_{ij}} \\ &= \frac{1}{|\mathcal{D}|}\sum_{x\in\mathcal{D}} \frac{\partial \mathcal{L}}{\partial y^l_i(x)}\phi(y^{l-1}_j(x)).\end{aligned}$$

Under the assumption that the weights used for forward propagation are independent from the weights used for back-propagation, $W^l_{ij}$ and $\frac{\partial \mathcal{L}}{\partial y^l_i(x)}$ are independent for all $x \in \mathcal{D}$. We also have that $W^l_{ij}$ and $\phi(y^{l-1}_j(x))$ are independent for all $x \in \mathcal{D}$. Therefore, $W^l_{ij}$ and $\frac{\partial \mathcal{L}}{\partial W^l_{ij}}$ are independent for all $l, i, j$. This yields

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

where $\alpha_l = \mathbb{E}[N_{l-1}(W^l_{11})^2\delta^l_{11}]$ (the choice of $i, j$ does not matter because they are iid). Unless we do not prune any weights from the $l^{th}$ layer, we have that $\alpha_l < 1$.
These dynamics are the same as a FFNN with the variance of the weights given by $\hat{\sigma}_w^2 = \sigma_w^2\alpha_l$. Since the EOC equation is given by $\sigma_w^2\mathbb{E}[\phi'(\sqrt{q}Z)^2] = 1$, with the new variance, it is clear that $\hat{\sigma}_w^2\mathbb{E}[\phi'(\sqrt{\hat{q}}Z)^2] \neq 1$ in general. Hence, the network is no longer on the EOC and this could be problematic for training.
With the rescaling, this becomes

$$\begin{aligned}\hat{q}^l(x,x') &= \sigma_w^2\rho_l^2\alpha_l\mathbb{E}[\phi(\tilde{y}^{l-1}_1(x))\phi(\tilde{y}^{l-1}_1(x'))] + \sigma_b^2 \\ &= \sigma_w^2\mathbb{E}[\phi(\tilde{y}^{l-1}_1(x))\phi(\tilde{y}^{l-1}_1(x'))] + \sigma_b^2.\end{aligned}$$

Therefore, the new variance after re-scaling is $\tilde{\sigma}_w^2 = \sigma_w^2$, and the limiting variance $\tilde{q} = q$ remains also unchanged since the dynamics are the same. Therefore $\tilde{\sigma}_w^2 \mathbb{E}[\phi'(\sqrt{\tilde{q}}Z)^2] = \sigma_w^2 \mathbb{E}[\phi'(\sqrt{q}Z)^2] = 1$. Thus, the re-scaled network is initialized on the EOC. The proof is similar for CNNs.

□

## D PROOF FOR SECTION 3 : SBP FOR STABLE RESIDUAL NETWORKS

**Theorem 2** (Resnet is well-conditioned)**.** *Consider a Resnet with either Fully Connected or Convolutional layers and ReLU activation function. Then for all $\sigma_w > 0$, the Resnet is well-conditioned. Moreover, for all $l \in \{1, ..., L\}$, $m^l = \Theta((1 + \frac{\sigma_w^2}{2})^L)$.*

*Proof.* Let us start with the case of a Resnet with Fully Connected layers. we have that

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial W_{ij}^l} &= \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \frac{\partial \mathcal{L}}{\partial y_i^l(x)} \frac{\partial y_i^l(x)}{\partial W_{ij}^l} \\ &= \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \frac{\partial \mathcal{L}}{\partial y_i^l(x)} \phi(y_j^{l-1}(x))\end{aligned}$$

and the backpropagation of the gradient is given by the set of equations

$$\frac{\partial \mathcal{L}}{\partial y_i^l} = \frac{\partial \mathcal{L}}{\partial y_i^{l+1}} + \phi'(y_i^l) \sum_{j=1}^{N_{l+1}} \frac{\partial \mathcal{L}}{\partial y_j^{l+1}} W_{ji}^{l+1}.$$

Recall that $q^l(x) = \mathbb{E}[y_i^l(x)^2]$ and $\tilde{q}^l(x, x') = \mathbb{E}[\frac{\partial \mathcal{L}}{\partial y_i^l(x)} \frac{\partial \mathcal{L}}{\partial y_i^l(x')}]$ for some inputs $x, x'$. We have that

$$q^l(x) = \mathbb{E}[y_i^{l-1}(x)^2] + \sigma_w^2 \mathbb{E}[\phi(y_1^{l-1})^2] = (1 + \frac{\sigma_w^2}{2}) q^{l-1}(x),$$

and

$$\tilde{q}^l(x, x') = (1 + \sigma_w^2 \mathbb{E}[\phi'(y_i^l(x))\phi'(y_i^l(x'))]) \tilde{q}^{l+1}(x, x').$$

We also have

$$\mathbb{E}[\frac{\partial \mathcal{L}}{\partial W_{ij}^l}^2] = \frac{1}{|\mathcal{D}|^2} \sum_{x, x'} t_{x,x'}^l,$$

where $t_{x,x'}^l = \tilde{q}^l(x, x') \sqrt{q^l(x) q^l(x')} f(c^{l-1}(x, x'))$ and $f$ is defined in the preliminary results (Eq 15).
Let $k \in \{1, 2, ..., L\}$ be fixed. We compare the terms $t_{x,x'}^l$ for $l = k$ and $l = L$. The ratio between the two terms is given by (after simplification)

$$\frac{t_{x,x'}^k}{t_{x,x'}^L} = \frac{\prod_{l=k}^{L-1} (1 + \frac{\sigma_w^2}{2} f'(c^l(x, x')))}{(1 + \frac{\sigma_w^2}{2})^{L-k}} \frac{f(c^{k-1}(x, x'))}{f(c^{L-1}(x, x'))}.$$

We have that $f'(c^l(x, x)) = f'(1) = 1$. A Taylor expansion of $f$ near 1 yields $f'(c^l(x, x')) = 1 - l^{-1} + o(l^{-1})$ and $f(c^l(x, x)) = 1 - sl^{-2} + o(l^{-2})$ (see Hayou et al. (2019) for more details). Therefore, there exist two constants $A, B > 0$ such that $A < \frac{\prod_{l=k}^{L-1} (1 + \frac{\sigma_w^2}{2} f'(c^l(x, x')))}{(1 + \frac{\sigma_w^2}{2})^{L-k}} < B$ for all $L$ and $k \in \{1, 2, ..., L\}$. This yields

$$A \leq \frac{\mathbb{E}[\frac{\partial \mathcal{L}}{\partial W_{ij}^l}^2]}{\mathbb{E}[\frac{\partial \mathcal{L}}{\partial W_{ij}^L}^2]} \leq B,$$

which concludes the proof.

For Resnet with convolutional layers, we have

$$\frac{\partial \mathcal{L}}{\partial W^l_{i,j,\beta}} = \frac{1}{|\mathcal{D}|}\sum_{x\in\mathcal{D}}\sum_{\alpha}\frac{\partial \mathcal{L}}{\partial y^l_{i,\alpha}(x)}\phi(y^{l-1}_{j,\alpha+\beta}(x))$$

and

$$\frac{\partial \mathcal{L}}{\partial y^l_{i,\alpha}} = \frac{\partial \mathcal{L}}{\partial y^{l+1}_{i,\alpha}} + \sum_{j=1}^{n}\sum_{\beta\in ker}\frac{\partial \mathcal{L}}{\partial y^{l+1}_{j,\alpha-\beta}}W^{l+1}_{i,j,\beta}\phi'(y^l_{i,\alpha}).$$

Recall the notation $\tilde{q}^l_{\alpha,\alpha'}(x,x') = \mathbb{E}[\frac{\partial \mathcal{L}}{\partial y^l_{i,\alpha}(x)}\frac{\partial \mathcal{L}}{\partial y^l_{i,\alpha'}(x')}]$. Using the hypothesis of independence of forward and backward weights and averaging over the number of channels (using CLT), we have

$$\tilde{q}^l_{\alpha,\alpha'}(x,x') = \tilde{q}^{l+1}_{\alpha,\alpha'}(x,x') + \frac{\sigma_w^2 f'(c^l_{\alpha,\alpha'}(x,x'))}{2(2k+1)}\sum_{\beta}\tilde{q}^{l+1}_{\alpha+\beta,\alpha'+\beta}(x,x').$$

Let $K_l = ((\tilde{q}^l_{\alpha,\alpha+\beta}(x,x'))_{\alpha\in[0:N-1]})_{\beta\in[0:N-1]}$ be a vector in $\mathbb{R}^{N^2}$. Writing this previous equation in matrix form, we obtain

$$K_l = (I + \frac{\sigma_w^2 f'(c^l_{\alpha,\alpha'}(x,x'))}{2(2k+1)}U)K_{l+1}$$

and

$$\mathbb{E}[\frac{\partial \mathcal{L}}{\partial W^l_{i,j,\beta}}^2] = \frac{1}{|\mathcal{D}|^2}\sum_{x,x'\in\mathcal{D}}\sum_{\alpha,\alpha'}t^l_{\alpha,\alpha'}(x,x'),$$

where $t^l_{\alpha,\alpha'}(x,x') = \tilde{q}^l_{\alpha,\alpha'}(x,x')\sqrt{q^l_{\alpha+\beta}(x)q^l_{\alpha'+\beta}(x')}f(c^{l-1}_{\alpha+\beta,\alpha'+\beta}(x,x'))$. Since we have $f'(c^l_{\alpha,\alpha'}(x,x')) \to 1$, then by fixing $l$ and letting $L$ goes to infinity, it follows that

$$K_l \sim_{L\to\infty} (1+\frac{\sigma_w^2}{2})^{L-l}e_1e_1^TK_L$$

and, from Lemma 2, we know that

$$\sqrt{q^l_{\alpha+\beta}(x)q^l_{\alpha'+\beta}(x')} = (1+\frac{\sigma_w^2}{2})^{l-1}\sqrt{q_{0,x}q_{0,x'}}.$$

Therefore, for a fixed $k < L$, we have $t^k_{\alpha,\alpha'}(x,x') \sim (1+\frac{\sigma_w^2}{2})^{L-1}f(c^{k-1}_{\alpha+\beta,\alpha'+\beta}(x,x'))(e_1^TK_L) = \Theta(t^L_{\alpha,\alpha'}(x,x'))$. This concludes the proof.

□

**Proposition 2** (Stable Resnet)**.** *Consider the following Resnet parameterization*

$$y^l(x) = y^{l-1}(x) + \frac{1}{\sqrt{L}}\mathcal{F}(W^l, y^{l-1}), \quad \textit{for } l \geq 2, \tag{18}$$

*then the network is well-conditioned for all choices of $\sigma_w > 0$. Moreover, for all $l \in \{1, ..., L\}$ we have $m^l = \Theta(L^{-1})$.*

*Proof.* The proof is similar to that of Theorem 2 with minor differences. Let us start with the case of a Resnet with fully connected layers, we have

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial W^l_{ij}} &= \frac{1}{|\mathcal{D}|\sqrt{L}}\sum_{x\in\mathcal{D}}\frac{\partial \mathcal{L}}{\partial y^l_i(x)}\frac{\partial y^l_i(x)}{\partial W^l_{ij}} \\ &= \frac{1}{|\mathcal{D}|\sqrt{L}}\sum_{x\in\mathcal{D}}\frac{\partial \mathcal{L}}{\partial y^l_i(x)}\phi(y^{l-1}_j(x))\end{aligned}$$

and the backpropagation of the gradient is given by

$$\frac{\partial \mathcal{L}}{\partial y_i^l} = \frac{\partial \mathcal{L}}{\partial y_i^{l+1}} + \frac{1}{\sqrt{L}}\phi'(y_i^l)\sum_{j=1}^{N_{l+1}} \frac{\partial \mathcal{L}}{\partial y_j^{l+1}} W_{ji}^{l+1}.$$

Recall that $q^l(x) = \mathbb{E}[y_i^l(x)^2]$ and $\tilde{q}^l(x,x') = \mathbb{E}[\frac{\partial \mathcal{L}}{\partial y_i^l(x)}\frac{\partial \mathcal{L}}{\partial y_i^l(x')}]$ for some inputs $x, x'$. We have

$$q^l(x) = \mathbb{E}[y_i^{l-1}(x)^2] + \frac{\sigma_w^2}{L}\mathbb{E}[\phi(y_1^{l-1}(x))^2] = (1+\frac{\sigma_w^2}{2L})q^{l-1}(x)$$

and

$$\tilde{q}^l(x,x') = (1+\frac{\sigma_w^2}{L}\mathbb{E}[\phi'(y_i^l(x))\phi'(y_i^l(x'))])\tilde{q}^{l+1}(x,x').$$

We also have

$$\mathbb{E}[\frac{\partial \mathcal{L}}{\partial W_{ij}^l}^2] = \frac{1}{L|\mathcal{D}|^2}\sum_{x,x'} t_{x,x'}^l,$$

where $t_{x,x'}^l = \tilde{q}^l(x,x')\sqrt{q^l(x)q^l(x')}f(c^{l-1}(x,x'))$ and $f$ is defined in the preliminary results (Eq. 15).
Let $k \in \{1,2,...,L\}$ be fixed. We compare the terms $t_{x,x'}^l$ for $l=k$ and $l=L$. The ratio between the two terms is given after simplification by

$$\frac{t_{x,x'}^k}{t_{x,x'}^L} = \frac{\prod_{l=k}^{L-1}(1+\frac{\sigma_w^2}{2L}f'(c^l(x,x')))}{(1+\frac{\sigma_w^2}{2L})^{L-k}}\frac{f(c^{k-1}(x,x'))}{f(c^{L-1}(x,x'))}.$$

As in the proof of Theorem 2, we have that $f'(c^l(x,x)) = 1$, $f'(c^l(x,x')) = 1 - l^{-1} + o(l^{-1})$ and $f(c^l(x,x)) = 1 - sl^{-2} + o(l^{-2})$. Therefore, there exist two constants $A, B > 0$ such that $A < \frac{\prod_{l=k}^{L-1}(1+\frac{\sigma_w^2}{2L}f'(c^l(x,x')))}{(1+\frac{\sigma_w^2}{2L})^{L-k}} < B$ for all $L$ and $k \in \{1,2,...,L\}$. This yields

$$A \leq \frac{\mathbb{E}[\frac{\partial \mathcal{L}}{\partial W_{ij}^l}^2]}{\mathbb{E}[\frac{\partial \mathcal{L}}{\partial W_{ij}^L}^2]} \leq B.$$

Moreover, since $(1+\frac{\sigma_w^2}{2L})^L \to e^{\sigma_w^2/2}$, then $m^l = \Theta(1)$ for all $l \in \{1,...,L\}$. This concludes the proof.

For Resnet with convolutional layers, the proof is similar. With the scaling, we have

$$\frac{\partial \mathcal{L}}{\partial W_{i,j,\beta}^l} = \frac{1}{\sqrt{L}|\mathcal{D}|}\sum_{x\in\mathcal{D}}\sum_{\alpha}\frac{\partial \mathcal{L}}{\partial y_{i,\alpha}^l(x)}\phi(y_{j,\alpha+\beta}^{l-1}(x))$$

and

$$\frac{\partial \mathcal{L}}{\partial y_{i,\alpha}^l} = \frac{\partial \mathcal{L}}{\partial y_{i,\alpha}^{l+1}} + \frac{1}{\sqrt{L}}\sum_{j=1}^{n}\sum_{\beta\in ker}\frac{\partial \mathcal{L}}{\partial y_{j,\alpha-\beta}^{l+1}}W_{i,j,\beta}^{l+1}\phi'(y_{i,\alpha}^l).$$

Let $\tilde{q}_{\alpha,\alpha'}^l(x,x') = \mathbb{E}[\frac{\partial \mathcal{L}}{\partial y_{i,\alpha}^l(x)}\frac{\partial \mathcal{L}}{\partial y_{i,\alpha'}^l(x')}]$. Using the hypothesis of independence of forward and backward weights and averaging over the number of channels (using CLT) we have

$$\tilde{q}_{\alpha,\alpha'}^l(x,x') = \tilde{q}_{\alpha,\alpha'}^{l+1}(x,x') + \frac{\sigma_w^2 f'(c_{\alpha,\alpha'}^l(x,x'))}{2(2k+1)L}\sum_{\beta}\tilde{q}_{\alpha+\beta,\alpha'+\beta}^{l+1}(x,x').$$

Let $K_l = ((\tilde{q}_{\alpha,\alpha+\beta}^l(x,x'))_{\alpha\in[0:N-1]})_{\beta\in[0:N-1]}$ is a vector in $\mathbb{R}^{N^2}$. Writing this previous equation in matrix form, we have

$$K_l = (I + \frac{\sigma_w^2 f'(c_{\alpha,\alpha'}^l(x,x'))}{2(2k+1)L}U)K_{l+1},$$

and

$$\mathbb{E}[\frac{\partial \mathcal{L}}{\partial W^l_{i,j,\beta}}^2] = \frac{1}{L|\mathcal{D}|^2}\sum_{x,x'\in\mathcal{D}}\sum_{\alpha,\alpha'} t^l_{\alpha,\alpha'}(x,x'),$$

where $t^l_{\alpha,\alpha'}(x,x') = \tilde{q}^l_{\alpha,\alpha'}(x,x')\sqrt{q^l_{\alpha+\beta}(x)q^l_{\alpha'+\beta}(x')}f(c^{l-1}_{\alpha+\beta,\alpha'+\beta}(x,x'))$. Since we have $f'(c^l_{\alpha,\alpha'}(x,x')) \to 1$, then by fixing $l$ and letting $L$ goes to infinity, we obtain

$$K_l \sim_{L\to\infty} (1+\frac{\sigma_w^2}{2L})^{L-l} e_1 e_1^T K_L$$

and we know from Appendix Lemma 2 (using $\alpha_\beta = \frac{\sigma_w^2}{2L}$ for all $\beta$) that

$$\sqrt{q^l_{\alpha+\beta}(x)q^l_{\alpha'+\beta}(x')} = (1+\frac{\sigma_w^2}{2L})^{l-1}\sqrt{q_{0,x}q_{0,x'}}.$$

Therefore, for a fixed $k < L$, we have $t^k_{\alpha,\alpha'}(x,x') \sim (1+\frac{\sigma_w^2}{2L})^{L-1} f(c^{k-1}_{\alpha+\beta,\alpha'+\beta}(x,x'))(e_1^T K_L) = \Theta(t^L_{\alpha,\alpha'}(x,x'))$ which proves that the stable Resnet is well conditioned. Moreover, since $(1+\frac{\sigma_w^2}{2L})^{L-1} \to e^{\sigma_w^2/2}$, then $m^l = \Theta(L^{-1})$ for all $l$.

□

In the next Lemma, we study the asymptotic behaviour of the variance $q^l_\alpha$. We show that, as $l \to \infty$, a phenomenon of self averaging shows that $q^l_\alpha$ becomes independent of $\alpha$.

**Appendix Lemma 2.** *Let $x \in \mathbb{R}^d$. Assume the sequence $(a_{l,\alpha})_{l,\alpha}$ is given by the recursive formula*

$$a_{l,\alpha} = a_{l-1,\alpha} + \sum_{\beta\in ker} \lambda_\beta a_{l-1,\alpha+\beta}$$

*where $\lambda_\beta > 0$ for all $\beta$. Then, there exists $\zeta > 0$ such that for all $x \in \mathbb{R}^d$ and $\alpha$,*

$$a_{l,\alpha}(x) = (1+\sum_\beta \alpha_\beta)^l a_0 + \mathcal{O}((1+\sum_\beta \alpha_\beta)^l e^{-\zeta l}),$$

*where $a_0$ is a constant and the $\mathcal{O}$ is uniform in $\alpha$.*

*Proof.* Recall that

$$a_{l,\alpha} = a_{l-1,\alpha} + \sum_{\beta\in ker} \lambda_\beta a_{l-1,\alpha+\beta}.$$

We rewrite this expression in a matrix form

$$A_l = UA_{l-1},$$

where $A_l = (a_{l,\alpha})_\alpha$ is a vector in $\mathbb{R}^N$ and $U$ is the is the convolution matrix. As an example, for $k = 1$, $U$ given by

$$U = \begin{bmatrix} 1+\lambda_0 & \lambda_1 & 0 & ... & 0 & \lambda_{-1} \\ \lambda_{-1} & 1+\lambda_0 & \lambda_1 & 0 & \ddots & 0 \\ 0 & \lambda_{-1} & 1+\lambda_0 & \lambda_1 & \ddots & 0 \\ 0 & 0 & \lambda_{-1} & 1+\lambda_0 & \ddots & 0 \\ & \ddots & \ddots & \ddots & \ddots & \\ \lambda_1 & 0 & \dots & 0 & \lambda_{-1} & 1+\lambda_0 \end{bmatrix}.$$

$U$ is a circulant symmetric matrix with eigenvalues $b_1 > b_2 \geq b_3 ... \geq b_N$. The largest eigenvalue of $U$ is given by $b_1 = 1 + \sum_\beta \lambda_\beta$ and its equivalent eigenspace is generated by the vector $e_1 = \frac{1}{\sqrt{N}}(1,1,...,1) \in \mathbb{R}^N$. This yields

$$b_1^{-l} U^l = e_1 e_1^T + O(e^{-\zeta l}),$$

where $\zeta = \log(\frac{b_1}{b_2})$. Using this result, we obtain

$$b_1^{-l} A_l = (b_1^{-l} U^l) A_0 = e_1 e_1^T A_0 + O(e^{-\zeta l}).$$

This concludes the proof. □

Unlike FFNN or CNN, we do not need to rescale the pruned network. The next proposition establishes that a Resnet lives on the EOC in the sense that the correlation between $y_i^l(x)$ and $y_i^l(x')$ converges to 1 at a sub-exponential $\mathcal{O}(l^{-2})$ rate.

**Proposition 3** (Resnet live on the EOC even after pruning)**.** *Let $x \neq x'$ be two inputs. The following statments hold*

1. *For Resnet with Fully Connected layers, let $\hat{c}^l(x, x')$ be the correlation between $\hat{y}_i^l(x)$ and $\hat{y}_i^l(x')$ after pruning the network. Then we have*

$$1 - \hat{c}^l(x, x') \sim \frac{\kappa}{l^2},$$

   *where $\kappa > 0$ is a constant.*

2. *For Resnet with Convolutional layers, let $\hat{c}^l(x, x') = \frac{\sum_{\alpha,\alpha'} \mathbb{E}[y_{1,\alpha}^l(x) y_{1,\alpha'}^l(x')]}{\sum_{\alpha,\alpha'} \sqrt{q_\alpha^l(x)} \sqrt{q_\alpha^{l\prime}(x')}}$ be an 'average' correlation after pruning the network. Then we have*

$$1 - \hat{c}^l(x, x') \gtrsim l^{-2}.$$

*Proof.* It is sufficient to prove the result when $\alpha = \mathbb{E}[N_{l-1} {W_{11}^l}^2 \delta_{11}^l]$ is the same for all $l$. The proof for the general case is straightforward using the same techniques.

1. Let $x$ and $x'$ be two inputs. The covariance of $\hat{y}_i^l(x)$ and $\hat{y}_i^l(x')$ is given by

$$\hat{q}^l(x, x') = \hat{q}^{l-1}(x, x') + \alpha \mathbb{E}_{(Z_1, Z_2) \sim \mathcal{N}(0, Q^{l-1})}[\phi(Z_1)\phi(Z_2)]$$

   where $Q^{l-1} = \begin{bmatrix} \hat{q}^{l-1}(x) & \hat{q}^{l-1}(x, x') \\ \hat{q}^{l-1}(x, x') & \hat{q}^{l-1}(x') \end{bmatrix}$ and $\alpha = \mathbb{E}[N_{l-1} {W_{11}^l}^2 \delta_{11}^l]$.

   Consequently, we have $\hat{q}^l(x) = (1 + \frac{\alpha}{2}) \hat{q}^{l-1}(x)$. Therefore, we obtain

$$\hat{c}^l(x, x') = \frac{1}{1+\lambda} \hat{c}^{l-1}(x, x') + \frac{\lambda}{1+\lambda} f(\hat{c}^{l-1}(x, x')),$$

   where $\lambda = \frac{\alpha}{2}$ and $f(x) = 2\mathbb{E}[\phi(Z_1)\phi(xZ_1 + \sqrt{1-x^2} Z_2)]$ and $Z_1$ and $Z_2$ are iid standard normal variables.

   Using the fact that $f$ is increasing (Section B.1), it is easy to see that $\hat{c}^l(x, x') \to 1$. Let $\zeta_l = 1 - \hat{c}^l(x, x')$. Moreover, using a Taylor expansion of $f$ near 1 (Section B.1) $f(x) \underset{x \to 1^-}{=} x + \beta(1-x)^{3/2} + O((1-x)^{5/2})$, it follows that

$$\zeta_l = \zeta_{l-1} - \eta \zeta_{l-1}^{3/2} + O(\zeta_{l-1}^{5/2}),$$

   where $\eta = \frac{\lambda\beta}{1+\lambda}$. Now using the asymptotic expansion of $\zeta_l^{-1/2}$ given by

$$\zeta_l^{-1/2} = \zeta_{l-1}^{-1/2} + \frac{\eta}{2} + O(\zeta_{l-1}),$$

   this yields $\zeta_l^{-1/2} \underset{l \to \infty}{\sim} \frac{\eta}{2} l$. We conclude that $1 - \hat{c}^l(x, x') \sim \frac{4}{\eta^2 l^2}$.

2. Let $x$ be an input. Recall the forward propagation of a pruned 1D CNN

$$y_{i,\alpha}^l(x) = y_{i,\alpha}^{l-1}(x) + \sum_{j=1}^{c} \sum_{\beta \in ker} \delta_{i,j.\beta}^l W_{i,j,\beta}^l \phi(y_{j,\alpha+\beta}^{l-1}(x)) + b_i^l.$$

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

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

where $\alpha_\beta = \mathbb{E}[\delta^l_{i,1.\beta}{W^l_{i,1,\beta}}^2 n_{l-1}]$.

Let $q^l_\alpha(x) = \mathbb{E}[y^l_{1,\alpha}(x)^2]$. The choice of the channel is not important since for a given $\alpha$, neurons $(y^l_{i,\alpha}(x))_{i\in[c]}$ are iid. Using the previous formula, we have

$$\begin{aligned} q^l_\alpha(x) &= q^{l-1}_\alpha(x) + \frac{1}{2k+1}\sum_{\beta\in ker}\alpha_\beta \mathbb{E}[\phi(y^{l-1}_{1,\alpha+\beta}(x))^2] \\ &= q^{l-1}_\alpha(x) + \frac{1}{2k+1}\sum_{\beta\in ker}\alpha_\beta \frac{q^{l-1}_{\alpha+\beta}(x)}{2}. \end{aligned}$$

Therefore, letting $q^l(x) = \frac{1}{N}\sum_{\alpha\in[N]} q^l_\alpha(x)$ and $\sigma = \frac{\sum_\beta \alpha_\beta}{2k+1}$, we obtain

$$\begin{aligned} q^l(x) &= q^{l-1}(x) + \frac{1}{2k+1}\sum_{\beta\in ker}\alpha_\beta \sum_{\alpha\in[n]}\frac{q^{l-1}_{\alpha+\beta}(x)}{2} \\ &= (1+\frac{\sigma}{2})q^{l-1}(x) = (1+\frac{\sigma}{2})^{l-1}q^1(x), \end{aligned}$$

where we have used the periodicity $q^{l-1}_\alpha = q^{l-1}_{\alpha-N} = q^{l-1}_{\alpha+N}$. Moreover, we have $\min_\alpha q^l_\alpha(x) \geq (1+\frac{\sigma}{2})\min_\alpha q^{l-1}_\alpha(x) \geq (1+\frac{\sigma}{2})^{l-1}\min_\alpha q^1_\alpha(x)$.

The convolutional structure makes it hard to analyse the correlation between the values of a neurons for two different inputs. Xiao et al. (2018) studied the correlation between the values of two neurons in the same channel for the same input. Although this could capture the propagation of the input structure (say how different pixels propagate together) inside the network, it does not provide any information on how different structures from different inputs propagate. To resolve this situation, we study the 'average' correlation per channel defined as

$$c^l(x,x') = \frac{\sum_{\alpha,\alpha'}\mathbb{E}[y^l_{1,\alpha}(x)y^l_{1,\alpha'}(x')]}{\sum_{\alpha,\alpha'}\sqrt{q^l_\alpha(x)}\sqrt{{q^l_\alpha}'(x')}},$$

for any two inputs $x \neq x'$. We also define $\breve{c}^l(x,x')$ by

$$\breve{c}^l(x,x') = \frac{\frac{1}{N^2}\sum_{\alpha,\alpha'}\mathbb{E}[y^l_{1,\alpha}(x)y^l_{1,\alpha'}(x')]}{\sqrt{\frac{1}{N}\sum_\alpha q^l_\alpha(x)}\sqrt{\frac{1}{N}\sum_\alpha q^l_\alpha(x')}}.$$

Using the concavity of the square root function, we have

$$\begin{aligned} \sqrt{\frac{1}{N}\sum_\alpha q^l_\alpha(x)}\sqrt{\frac{1}{N}\sum_\alpha q^l_\alpha(x')} &= \sqrt{\frac{1}{N^2}\sum_{\alpha,\alpha'} q^l_\alpha(x)q^l_\alpha(x')} \\ &\geq \frac{1}{N^2}\sum_{\alpha,\alpha'}\sqrt{q^l_\alpha(x)}\sqrt{q^l_\alpha(x')} \\ &\geq \frac{1}{N^2}\sum_{\alpha,\alpha'}|\mathbb{E}[y^l_{1,\alpha}(x)y^l_{1,\alpha'}(x')]|. \end{aligned}$$

This yields $\breve{c}^l(x,x') \leq c^l(x,x') \leq 1$. Using Appendix Lemma 2 twice with $a_{l,\alpha} = q^l_\alpha(x)$, $a_{l,\alpha} = q^l_\alpha(x')$, and $\lambda_\beta = \frac{\alpha_\beta}{2(2k+1)}$, there exists $\zeta > 0$ such that

$$c^l(x,x') = \breve{c}^l(x,x')(1+\mathcal{O}(e^{-\zeta l})). \tag{19}$$

This result shows that the limiting behaviour of $c^l(x,x')$ is equivalent to that of $\breve{c}^l(x,x')$ up to an exponentially small factor. We study hereafter the behaviour of $\breve{c}^l(x,x')$ and use this result to conclude. Recall that

$$\mathbb{E}[y^l_{i,\alpha}(x)y^l_{i,\alpha'}(x')] = \mathbb{E}[y^{l-1}_{i,\alpha}(x)y^{l-1}_{i,\alpha'}(x')] + \frac{1}{2k+1}\sum_{\beta\in ker}\alpha_\beta\mathbb{E}[\phi(y^{l-1}_{1,\alpha+\beta}(x))\phi(y^{l-1}_{1,\alpha'+\beta}(x'))].$$

Therefore,

$$\begin{aligned}
&\sum_{\alpha,\alpha'}\mathbb{E}[y^l_{1,\alpha}(x)y^l_{1,\alpha'}(x')]\\
&=\sum_{\alpha,\alpha'}\mathbb{E}[y^{l-1}_{1,\alpha}(x)y^{l-1}_{1,\alpha'}(x')] + \frac{1}{2k+1}\sum_{\alpha,\alpha'}\sum_{\beta\in ker}\alpha_\beta\mathbb{E}[\phi(y^{l-1}_{1,\alpha+\beta}(x))\phi(y^{l-1}_{1,\alpha'+\beta}(x'))]\\
&=\sum_{\alpha,\alpha'}\mathbb{E}[y^{l-1}_{1,\alpha}(x)y^{l-1}_{1,\alpha'}(x')] + \sigma\sum_{\alpha,\alpha'}\mathbb{E}[\phi(y^{l-1}_{1,\alpha}(x))\phi(y^{l-1}_{1,\alpha'}(x'))]\\
&=\sum_{\alpha,\alpha'}\mathbb{E}[y^{l-1}_{1,\alpha}(x)y^{l-1}_{1,\alpha'}(x')] + \frac{\sigma}{2}\sum_{\alpha,\alpha'}\sqrt{q^{l-1}_\alpha(x)}\sqrt{q^{l-1\prime}_\alpha(x')}f(c^{l-1}_{\alpha,\alpha'}(x,x')),
\end{aligned}$$

where $f$ is the correlation function of ReLU.

Let us first prove that $\breve{c}^l(x,x')$ converges to 1. Using the fact that $f(z)\geq z$ for all $z\in(0,1)$ (Section B.1), we have that

$$\begin{aligned}
\sum_{\alpha,\alpha'}\mathbb{E}[y^l_{1,\alpha}(x)y^l_{1,\alpha'}(x')] &\geq \sum_{\alpha,\alpha'}\mathbb{E}[y^{l-1}_{1,\alpha}(x)y^{l-1}_{1,\alpha'}(x')] + \frac{\sigma}{2}\sum_{\alpha,\alpha'}\sqrt{q^{l-1}_\alpha(x)}\sqrt{q^{l-1\prime}_\alpha(x')}c^{l-1}_{\alpha,\alpha'}(x,x')\\
&=\sum_{\alpha,\alpha'}\mathbb{E}[y^{l-1}_{1,\alpha}(x)y^{l-1}_{1,\alpha'}(x')] + \frac{\sigma}{2}\sum_{\alpha,\alpha'}\mathbb{E}[y^{l-1}_{1,\alpha}(x)y^{l-1}_{1,\alpha'}(x')]\\
&=(1+\frac{\sigma}{2})\sum_{\alpha,\alpha'}\mathbb{E}[y^{l-1}_{1,\alpha}(x)y^{l-1}_{1,\alpha'}(x')].
\end{aligned}$$

Combining this result with the fact that $\sum_\alpha q^l_\alpha(x) = (1+\frac{\sigma}{2})\sum_\alpha q^{l-1}_\alpha(x)$, we have $\breve{c}^l(x,x')\geq\breve{c}^{l-1}(x,x')$. Therefore $\breve{c}^l(x,x')$ is non-decreasing and converges to a limiting point $c$.
Let us prove that $c=1$. By contradiction, assume the limit $c<1$. Using equation (19), we have that $\frac{c^l(x,x')}{\breve{c}^l(x,x')}$ converge to 1 as $l$ goes to infinity. This yields $c^l(x,x')\to c$. Therefore, there exists $\alpha_0,\alpha_0'$ and a constant $\delta<1$ such that for all $l$, $c^l_{\alpha_0,\alpha_0'}(x,x')\leq\delta<1$. Knowing that $f$ is strongly convex and that $f'(1)=1$, we have that $f(c^l_{\alpha_0,\alpha_0'}(x,x'))\geq c^l_{\alpha_0,\alpha_0'}(x,x')+f(\delta)-\delta$. Therefore,

$$\begin{aligned}
\breve{c}^l(x,x') &\geq \breve{c}^{l-1}(x,x') + \frac{\frac{\sigma}{2}\sqrt{q^{l-1}_{\alpha_0}(x)q^{l-1}_{\alpha_0'}(x')}}{N^2\sqrt{q^l(x)}\sqrt{q^l(x')}}(f(\delta)-\delta)\\
&\geq \breve{c}^{l-1}(x,x') + \frac{\frac{\sigma}{2}\sqrt{\min_\alpha q^1_\alpha(x)\min_{\alpha'}q^1_{\alpha'}(x')}}{N^2\sqrt{q^1(x)}\sqrt{q^1(x')}}(f(\delta)-\delta).
\end{aligned}$$

By taking the limit $l\to\infty$, we find that $c\geq c+\frac{\frac{\sigma}{2}\sqrt{\min_\alpha q^1_\alpha(x)\min_{\alpha'}q^1_{\alpha'}(x')}}{N^2\sqrt{q^1(x)}\sqrt{q^1(x')}}(f(\delta)-\delta)$. This cannot be true since $f(\delta)>\delta$. Thus we conclude that $c=1$.

Now we study the asymptotic convergence rate. From Section B.1, we have that

$$f(x)\underset{x\to1^-}{=}x+\frac{2\sqrt{2}}{3\pi}(1-x)^{3/2}+O((1-x)^{5/2}).$$

Therefore, there exists $\kappa > 0$ such that, close to $1^-$ we have that

$$f(x) \leq x + \kappa(1-x)^{3/2}.$$

Using this result, we can upper bound $\breve{c}^l(x, x')$

$$\breve{c}^l(x,x') \leq \breve{c}^{l-1}(x,x') + \kappa \sum_{\alpha,\alpha'} \frac{\frac{1}{N^2}\sqrt{q_\alpha^{l-1}(x)}\sqrt{q_{\alpha'}^{l-1}(x')}}{\sqrt{q^l(x)}\sqrt{q^l(x')}}(1 - c^l_{\alpha,\alpha'}(x,x'))^{3/2}.$$

To get a polynomial convergence rate, we should have an upper bound of the form $\breve{c}^l \leq \breve{c}^{l-1} + \zeta(1-\breve{c}^{l-1})^{1+\epsilon}$ (see below). However, the function $x^{3/2}$ is convex, so the sum cannot be upper-bounded directly using Jensen's inequality. We use here instead (Pečarić et al., 1992, Theorem 1) which states that for any $x_1, x_2, ...x_n > 0$ and $s > r > 0$, we have

$$\big(\sum_i x_i^s\big)^{1/s} < \big(\sum_i x_i^r\big)^{1/r}. \tag{20}$$

Let $z^l_{\alpha,\alpha'} = \frac{\frac{1}{N^2}\sqrt{q_\alpha^{l-1}(x)}\sqrt{q_{\alpha'}^{l-1}(x')}}{\sqrt{q^l(x)}\sqrt{q^l(x')}}$, we have

$$\sum_{\alpha,\alpha'} z^l_{\alpha,\alpha'}(1 - c^l_{\alpha,\alpha'}(x,x'))^{3/2} \leq \zeta_l \sum_{\alpha,\alpha'} [z^l_{\alpha,\alpha'}(1 - c^l_{\alpha,\alpha'}(x,x'))]^{3/2},$$

where $\zeta_l = \max_{\alpha,\alpha'} \frac{1}{{z^l_{\alpha,\alpha'}}^{1/2}}$. Using the inequality (20) with $s = 3/2$ and $r = 1$, we have

$$\begin{aligned}\sum_{\alpha,\alpha'} [z^l_{\alpha,\alpha'}(1 - c^l_{\alpha,\alpha'}(x,x'))]^{3/2} &\leq \big(\sum_{\alpha,\alpha'} z^l_{\alpha,\alpha'}(1 - c^l_{\alpha,\alpha'}(x,x'))\big)^{3/2} \\ &= \big(\sum_{\alpha,\alpha'} z^l_{\alpha,\alpha'} - \breve{c}^l(x,x')\big)^{3/2}.\end{aligned}$$

Moreover, using the concavity of the square root function, we have $\sum_{\alpha,\alpha'} z^l_{\alpha,\alpha'} \leq 1$. This yields

$$\breve{c}^l(x,x') \leq \breve{c}^{l-1}(x,x') + \zeta(1 - \breve{c}^{l-1}(x,x'))^{3/2},$$

where $\zeta$ is constant. Letting $\gamma_l = 1 - \breve{c}^l(x,x')$, we can conclude using the following inequality (we had an equality in the case of FFNN)

$$\gamma_l \geq \gamma_{l-1} - \zeta\gamma_{l-1}^{3/2}$$

which leads to

$$\gamma_l^{-1/2} \leq \gamma_{l-1}^{-1/2}(1 - \zeta\gamma_{l-1}^{1/2})^{-1/2} = \gamma_{l-1}^{-1/2} + \frac{\zeta}{2} + o(1).$$

Hence we have

$$\gamma_l \gtrsim l^{-2}.$$

Using this result combined with (19) again, we conclude that

$$1 - c^l(x,x') \gtrsim l^{-2}.$$

□

## E THEORETICAL ANALYSIS OF MAGNITUDE BASED PRUNING (MBP)

In this section, we provide a theoretical analysis of MBP. The two approximations from Appendix A are not used here.

MBP is a data independent pruning algorithm (zero-shot pruning). The mask is given by

$$\delta_i^l = \begin{cases} 1 & \text{if} \quad |W_i^l| \geq t_s, \\ 0 & \text{if} \quad |W_i^l| < t_s, \end{cases}$$

where $t_s$ is a threshold that depends on the sparsity $s$. By defining $k_s = (1-s)\sum_l M_l$, $t_s$ is given by $t_s = |W|^{(k_s)}$ where $|W|^{(k_s)}$ is the $k_s^{\text{th}}$ order statistic of the network weights $(|W_i^l|)_{1\leq l\leq L, 1\leq i\leq M_l}$ ($|W|^{(1)} > |W|^{(2)} > ...$).

With MBP, changing $\sigma_w$ does not impact the distribution of the resulting sparse architecture since it is a common factor for all the weights. However, in the case of different scaling factors $v_l$, the variances $\frac{\sigma_w^2}{v_l}$ used to initialize the weights vary across layers. This gives potentially the erroneous intuition that the layer with the smallest variance will be highly likely fully pruned before others as we increase the sparsity $s$. This is wrong in general since layers with small variances might have more weights compared to other layers. However, we can prove a similar result by considering the limit of large depth with fixed widths.

**Proposition 4** (MBP in the large depth limit)**.** *Assume $N$ is fixed and there exists $l_0 \in [1:L]$ such that $\alpha_{l_0} > \alpha_l$ for all $l \neq l_0$. Let $Q_x$ be the $x^{th}$ quantile of $|X|$ where $X \overset{iid}{\sim} \mathcal{N}(0,1)$ and $\gamma = \min_{l\neq l_0} \frac{\alpha_{l_0}}{\alpha_l}$. For $\epsilon \in (0,2)$, define $x_{\epsilon,\gamma} = \inf\{y \in (0,1) : \forall x > y, \gamma Q_x > Q_{1-(1-x)^{\gamma^{2-\epsilon}}}\}$ and $x_{\epsilon,\gamma} = \infty$ for the null set. Then, for all $\epsilon \in (0,2)$, $x_{\epsilon,\gamma}$ is finite and there exists a constant $\nu > 0$ such that*

$$\mathbb{E}[s_{cr}] \leq \inf_{\epsilon\in(0,2)} \{x_{\epsilon,\gamma} + \frac{\zeta_{l_0} N^2}{1+\gamma^{2-\epsilon}}(1-x_{\epsilon,\gamma})^{1+\gamma^{2-\epsilon}}\} + \mathcal{O}(\frac{1}{\sqrt{LN^2}}).$$

Proposition 4 gives an upper bound on $\mathbb{E}[s_{cr}]$ in the large depth limit. The upper bound is easy to approximate numerically. Table 7 compares the theoretical upper bound in Proposition 4 to the empirical value of $\mathbb{E}[s_{cr}]$ over 10 simulations for a FFNN with depth $L = 100$, $N = 100$, $\alpha_1 = \gamma$ and $\alpha_2 = \alpha_3 = \cdots = \alpha_L = 1$. Our experiments reveal that this bound can be tight.

Table 7: Theoretical upper bound of Proposition 4 and empirical observations for a FFNN with $N = 100$ and $L = 100$

| GAMMA | $\gamma = 2$ | $\gamma = 5$ | $\gamma = 10$ |
|---|---|---|---|
| UPPER BOUND | 5.77 | 0.81 | 0.72 |
| EMPIRICAL OBSERVATION | $\approx 1$ | 0.79 | 0.69 |

*Proof.* Let $x \in (0,1)$ and $k_x = (1-x)\Gamma_L N^2$, where $\Gamma_L = \sum_{l\neq l_0} \zeta_l$. We have

$$\mathbb{P}(s_{cr} \leq x) \geq \mathbb{P}(\max_i |W_i^{l_0}| < |W|^{(k_x)}),$$

where $|W|^{(k_x)}$ is the $k_x^{\text{th}}$ order statistic of the sequence $\{|W_i^l|, l \neq l_0, i \in [1:M_l]\}$; i.e $|W|^{(1)} > |W|^{(2)} > ... > |W|^{(k_x)}$.

Let $(X_i)_{i\in[1:M_{l_0}]}$ and $(Z_i)_{i\in[1:\Gamma_L N^2]}$ be two sequences of iid standard normal variables. It is easy to see that

$$\mathbb{P}(\max_{i,j} |W_{ij}^{l_0}| < |W|^{(k_x)}) \geq \mathbb{P}(\max_i |X_i| < \gamma |Z|^{(k_x)})$$

where $\gamma = \min_{l\neq l_0} \frac{\alpha_{l_0}}{\alpha_l}$.

Moreover, we have the following result from the theory of order statistics, which is a weak version of Theorem 3.1. in Puri and Ralescu (1986)

**Appendix Lemma 3.** *Let $X_1, X_2, ..., X_n$ be iid random variables with a cdf $F$. Assume $F$ is differentiable and let $p \in (0,1)$ and let $Q_p$ be the order $p$ quantile of the distribution $F$, i.e. $F(Q_p) = p$. Then we have*

$$\sqrt{n}(X^{(pn)} - Q_p)F'(Q_p)\sigma_p^{-1} \underset{D}{\to} \mathcal{N}(0,1),$$

*where the convergence is in distribution and $\sigma_p = p(1-p)$.*

Using this result, we obtain

$$\mathbb{P}(\max_i |X_i| < \gamma |Z|^{(k_x)}) = \mathbb{P}(\max_i |X_i| < \gamma Q_x) + \mathcal{O}(\frac{1}{\sqrt{LN^2}}),$$

where $Q_x$ is the $x$ quantile of the folded standard normal distribution.

The next result shows that $x_{\epsilon,\gamma}$ is finite for all $\epsilon \in (0,2)$.

**Appendix Lemma 4.** *Let $\gamma > 1$. For all $\epsilon \in (0,2)$, there exists $x_\epsilon \in (0,1)$ such that, for all $x > x_\epsilon$, $\gamma Q_x > Q_{1-(1-x)^{\gamma^{2-\epsilon}}}$.*

*Proof.* Let $\epsilon > 0$, and recall the asymptotic equivalent of $Q_{1-x}$ given by

$$Q_{1-x} \sim_{x \to 0} \sqrt{-2\log(x)}$$

Therefore, $\frac{\gamma Q_x}{Q_{1-(1-x)^{\gamma^{2-\epsilon}}}} \sim_{x \to 1} \sqrt{\gamma^\epsilon} > 1$. Hence $x_\epsilon$ exists and is finite. □

Let $\epsilon > 0$. Using Appendix Lemma 4, there exists $x_\epsilon > 0$ such that

$$\begin{aligned}
\mathbb{P}(\max_i |X_i| < \gamma Q_x) &\geq \mathbb{P}(\max_i |X_i| < Q_{1-(1-x)^{\gamma^{2-\epsilon}}}) \\
&= (1-(1-x)^{\gamma^{2-\epsilon}})^{\zeta_{l_0} N^2} \\
&\geq 1 - \zeta_{l_0} N^2 (1-x)^{\gamma^{2-\epsilon}},
\end{aligned}$$

where we have used the inequality $(1-t)^z \geq 1 - zt$ for all $(t,z) \in [0,1] \times (1,\infty)$.

Using the last result, we have

$$\mathbb{P}(s_{cr} \geq x) \leq \zeta_{l_0} N^2 (1-x)^{\gamma^{2-\epsilon}} + \mathcal{O}(\frac{1}{\sqrt{LN^2}}).$$

Now we have

$$\begin{aligned}
\mathbb{E}[s_{cr}] &= \int_0^1 \mathbb{P}(s_{cr} \geq x) dx \\
&\leq x_\epsilon + \int_{x_\epsilon}^1 \mathbb{P}(s_{cr} \geq x) dx \\
&\leq x_\epsilon + \frac{\zeta_{l_0} N^2}{1+\gamma^{2-\epsilon}} (1-x_\epsilon)^{\gamma^{2-\epsilon}+1} + \mathcal{O}(\frac{1}{\sqrt{LN^2}}).
\end{aligned}$$

This is true for all $\epsilon \in (0,2)$, and the additional term $\mathcal{O}(\frac{1}{\sqrt{LN^2}})$ does not depend on $\epsilon$. Therefore there exists a constant $\nu \in \mathbb{R}$ such that for all $\epsilon$

$$\mathbb{E}[s_{cr}] \leq x_\epsilon + \frac{\zeta_{l_0} N^2}{1+\gamma^{2-\epsilon}} (1-x_\epsilon)^{\gamma^{2-\epsilon}+1} + \frac{\nu}{\sqrt{LN^2}}.$$

We conclude by taking the infimum over $\epsilon$. □

## F IMAGENET EXPERIMENTS

To validate our results on large scale datasets, we prune ResNet50 using SNIP, GraSP, SynFlow and our algorithm SBP-SR, and train the pruned network on ImageNet. We train the pruned model for 90 epochs with SGD. The training starts with a learning rate 0.1 and it drops by a factor of 10 at epochs $30, 60, 80$. We report in table 8 Top-1 test accuracy for different sparsities. Our algorithm SBP-SR has a clear advantage over other algorithms. We are currently running extensive simulations on ImageNet to confirm these results.

Table 8: Classification accuracy on ImageNet (Top-1) for ResNet50 with varying sparsities (TODO: These results will be updated to include confidence intervals)

| ALGORITHM | 85% | 90% | 95% |
|---|---|---|---|
| SNIP | 69.05 | 64.25 | 44.90 |
| GRASP | 69.45 | 66.41 | 62.10 |
| SYNFLOW | 69.50 | 66.20 | 62.05 |
| SBP-SR | **69.75** | **67.02** | **62.66** |

## G ADDITIONAL EXPERIMENTS

In Table 10, we present additional experiments with varying Resnet Architectures (Resnet32/50), and sparsities (up to 99.9%) with Relu and Tanh activation functions on Cifar10. We see that overall, using our proposed Stable Resnet performs overall better that standard Resnets.

In addition, we also plot the remaining weights for each layer to get a better understanding on the different pruning strategies and well as understand why some of the Resnets with Tanh activation functions are untrainable. Furthermore, we added additional MNIST experiments with different activation function (ELU, Tanh) and note that our rescaled version allows us to prune significantly more for deeper networks.

Table 9: Test accuracy of pruned neural network on CIFAR10 with different activation functions

| Resnet32 | Algo | 90 | 98 | 99.5 | 99.9 |
|---|---|---|---|---|---|
| Relu | SBP-SR | **92.56(0.06)** | **88.25(0.35)** | **79.54(1.12)** | **51.56(1.12)** |
| | SNIP | 92.24(0.25) | 87.63(0.16) | 77.56(0.36) | 10(0) |
| Tanh | SBP-SR | **90.97(0.2)** | **86.62(0.38)** | **75.04(0.49)** | **51.88(0.56)** |
| | SNIP | 90.69(0.28) | 85.47(0.18) | 10(0) | 10(0) |
| Resnet50 | | | | | |
| Relu | SBP-SR | **92.05(0.06)** | **89.57(0.21)** | **82.68(0.52)** | **58.76(1.82)** |
| | SNIP | 91.64(0.14) | 89.20(0.54) | 80.49(2.41) | 19.98(14.12) |
| Tanh | SBP-SR | **90.43(0.32)** | **88.18(0.10)** | **80.09(0.0.55)** | **58.21(1.61)** |
| | SNIP | 89.55(0.10) | 10(0) | 10(0) | 10(0) |

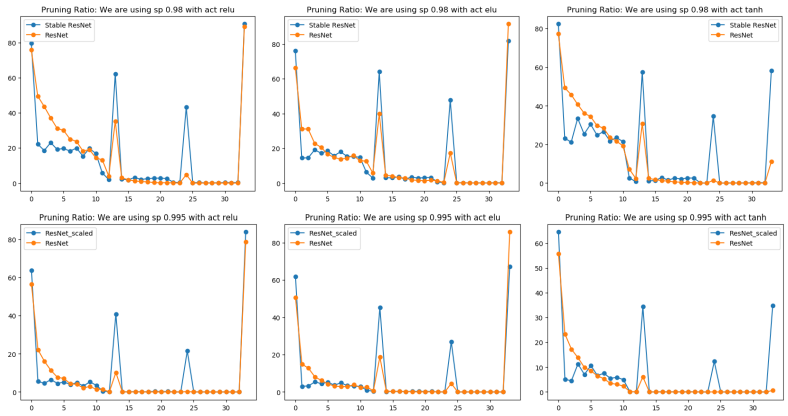
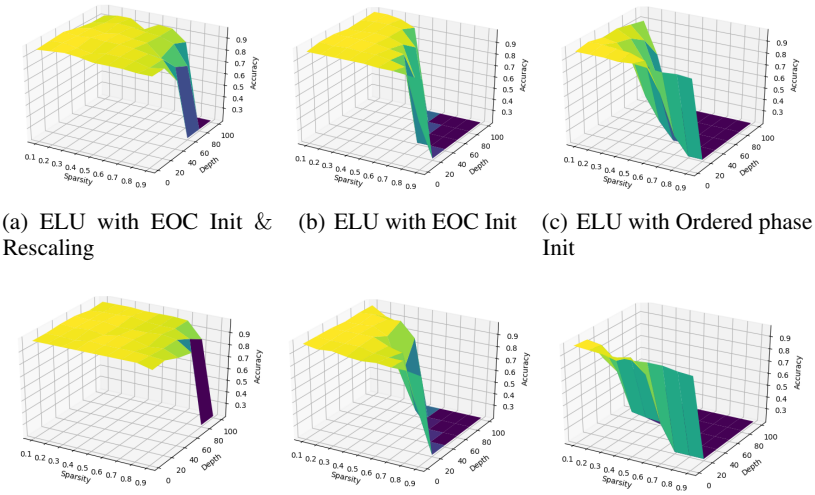



Figure 4: Percentage of pruned weights per layer in a ResNet32 for our scaled ResNet32 and standard Resnet32 with Kaiming initialization




(a) ELU with EOC Init & Rescaling

(b) ELU with EOC Init

(c) ELU with Ordered phase Init

(d) Tanh with EOC Init & Rescaling

(e) Tanh with EOC Init

(f) Tanh with Ordered phase Init

Figure 5: Accuracy on MNIST with different initialization schemes including EOC with rescaling, EOC without rescaling, Ordered phase, with varying depth and sparsity. This figure clearly illustrates the benefits of rescaling very deep and sparse FFNN.

Table 10: Test accuracy of pruned vanilla-CNN on CIFAR10 with different depth/sparsity levels

| Resnet32 | Algo | 90 | 98 | 99.5 | 99.9 |
|---|---|---|---|---|---|
| Relu | SBP-SR | **92.56(0.06)** | **88.25(0.35)** | **79.54(1.12)** | **51.56(1.12)** |
| | SNIP | 92.24(0.25) | 87.63(0.16) | 77.56(0.36) | 10(0) |
| Tanh | SBP-SR | **90.97(0.2)** | **86.62(0.38)** | **75.04(0.49)** | **51.88(0.56)** |
| | SNIP | 90.69(0.28) | 85.47(0.18) | 10(0) | 10(0) |
| Resnet50 | | | | | |
| Relu | SBP-SR | **92.05(0.06)** | **89.57(0.21)** | **82.68(0.52)** | **58.76(1.82)** |
| | SNIP | 91.64(0.14) | 89.20(0.54) | 80.49(2.41) | 19.98(14.12) |
| Tanh | SBP-SR | **90.43(0.32)** | **88.18(0.10)** | **80.09(0.0.55)** | **58.21(1.61)** |
| | SNIP | 89.55(0.10) | 10(0) | 10(0) | 10(0) |

## H ON THE LOTTERY TICKET HYPOTHESIS

The Lottery Ticket Hypothesis (LTH) (Frankle and Carbin, 2019) states that "randomly initialized networks contain subnetworks that when trained in isolation reach test accuracy comparable to the original network". We have shown so far that pruning a NN initialized on the EOC will output sparse NNs that can be trained after rescaling. Conversely, if we initialize a random NN with any hyperparameters $(\sigma_w, \sigma_b)$, then intuitively, we can prune this network in a way that ensures that the pruned NN is on the EOC. This would theoretically make the sparse architecture trainable. We formalize this intuition as follows.

**Weak Lottery Ticket Hypothesis (WLTH):** *For any randomly initialized network, there exists a subnetwork that is initialized on the Edge of Chaos.*

In the next theorem, we prove that the WLTH is true for FFNN and CNN architectures that are initialized with Gaussian distribution.

**Theorem 3.** *Consider a FFNN or CNN with layers initialized with variances $\sigma_w^2 > 0$ for weights and variance $\sigma_b^2$ for bias. Let $\sigma_{w,EOC}$ be the value of $\sigma_w$ such that $(\sigma_{w,EOC}, \sigma_b) \in EOC$. Then, for all $\sigma_w > \sigma_{w,EOC}$, there exists a subnetwork that is initialized on the EOC. Therefore WLTH is true.*

The idea behind the proof of Theorem 3 is that by removing a fraction of weights from each layer, we are changing the covariance structure in the next layer. By doing so in a precise way, we can find a subnetwork that is initialized on the EOC.

We prove a slightly more general result than the one stated.

**Theorem 4** (Winning Tickets on the Edge of Chaos)**.** *Consider a neural network with layers initialized with variances $\sigma_{w,l} \in \mathbb{R}^+$ for each layer and variance $\sigma_b > 0$ for bias. We define $\sigma_{w,EOC}$ to be the value of $\sigma_w$ such that $(\sigma_{w,EOC}, \sigma_b) \in EOC$. Then, for all sequences $(\sigma_{w,l})_l$ such that $\sigma_{w,l} > \sigma_{w,EOC}$ for all $l$, there exists a distribution of subnetworks initialized on the Edge of Chaos.*

*Proof.* We prove the result for FFNN. The proof for CNN is similar. Let $x, x'$ be two inputs. For all $l$, let $(\delta^l)_{ij}$ be a collection of Bernoulli variables with probability $p_l$. The forward propagation of the covariance is given by

$$\begin{aligned}\hat{q}^l(x, x') &= \mathbb{E}[y_i^l(x) y_i^l(x')] \\ &= \mathbb{E}[\sum_{j,k}^{N_{l-1}} W_{ij}^l W_{ik}^l \delta_{ij}^l \delta_{ik}^l \phi(\hat{y}_j^{l-1}(x)) \phi(\hat{y}_j^{l-1}(x'))] + \sigma_b^2.\end{aligned}$$

This yields

$$\hat{q}^l(x, x') = \sigma_{w,l}^2 p_l \mathbb{E}[\phi(\hat{y}_1^{l-1}(x)) \phi(\hat{y}_1^{l-1}(x'))] + \sigma_b^2.$$

By choosing $p_l = \frac{\sigma_{w,EOC}^2}{\sigma_{w,l}^2}$, this becomes

$$\hat{q}^l(x, x') = \sigma_{w,EOC}^2 \mathbb{E}[\phi(\tilde{y}_1^{l-1}(x)) \phi(\tilde{y}_1^{l-1}(x'))] + \sigma_b^2.$$

Therefore, the new variance after pruning with the Bernoulli mask $\delta$ is $\tilde{\sigma}_w^2 = \sigma_{w,EOC}^2$. Thus, the subnetwork defined by $\delta$ is initialized on the EOC. The distribution of these subnetworks is directly linked to the distribution of $\delta$. We can see this result as layer-wise pruning, i.e. pruning each layer aside. The proof is similar for CNNs. □

Theorem 3 is a special case of the previous result where the variances $\sigma_{w,l}$ are the same for all layers.

# I ALGORITHM FOR SECTION 2.3

**Algorithm 1** Rescaling trick for FFNN

**Input:** Pruned network, size $m$
**for** $L = 1$ **to** $L$ **do**
  **for** $i = 1$ **to** $N_l$ **do**
    $\alpha_i^l \leftarrow \sum_{j=1}^{N_{l-1}} (W_{ij})^2 \delta_{ij}^l$
    $\rho_{ij}^l \leftarrow 1/\sqrt{\alpha_i^l}$ **for all** $j$
  **end for**
**end for**

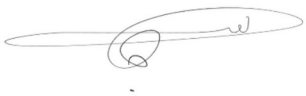
## Statement of Authorship for joint/multi-authored papers for PGR thesis

To appear at the end of each thesis chapter submitted as an article/paper

The statement shall describe the candidate's and co-authors' independent research contributions in the thesis publications. For each publication there should exist a complete statement that is to be filled out and signed by the candidate and supervisor **(only required where there isn't already a statement of contribution within the paper itself).**

| | |
|---|---|
| Title of Paper | Robust Pruning at Initialization |
| Publication Status | ■ Published □ Accepted for Publication<br>□ Submitted for Publication □Unpublished and unsubmitted work written in a manuscript style |
| Publication Details | Hayou, S., Ton, J.F., Doucet, A., Teh, Y.W. (2020). Robust Pruning at Initialization. Proceedings of the 9th International Conference on Learning Representations (ICLR 2021). |

### Student Confirmation

| | | | |
|---|---|---|---|
| Student Name: | Soufiane Hayou | | |
| Contribution to the Paper | I was inspired by my previous work on signal propagation to initiate this project. I worked on the theory and proofs behind this paper while on the experiments were done by Jean Francois Ton. Arnaud Doucet contributed to this work by providing valuable insights and helpful remarks and helped a lot to the writing of the draft, checking the proofs, and proof-reading. Yee Whye Teh gave valuable insights on ResNet section. | | |
| Signature |  | Date | 21/01/2021 |

### Supervisor Confirmation

By signing the Statement of Authorship, you are certifying that the candidate made a substantial contribution to the publication, and that the description described above is accurate.

| | | |
|---|---|---|
| Supervisor name and title: Arnaud Doucet | | |
| Supervisor comments | | |
| Signature | Date | |

This completed form should be included in the thesis, at the end of the relevant chapter.

# Discussion

# 6
# Conclusion and Discussion

## Contents



This chapter concludes the dissertation by summarizing the presented work and outlining its limitations and some potential avenues for future research.

## 6.1 Contributions

This dissertation provided a comprehensive analysis of the theory of the Edge of Chaos; it leveraged a duality between Gaussian processes and randomly initialized wide neural networks to characterize information propagation inside neural networks. Furthermore, it introduced a new class of smooth activation functions that have the advantage of letting the information propagate deeper, compared to ReLU activation function. Through the articles presented, we have shown that the residual architecture is by default suitable for deep neural networks, and that it is further improved by introducing scaling factors that ensure universality of the

kernel of the Neural Network Gaussian Process in the limit of infinite depth, thus allowing training such architectures for arbitrarily large depth. These new residual architectures were termed "Stable ResNet" as they stabilize the gradient while preserving expressiveness. The theoretical analysis of this dissertation was further refined by studying the Neural Tangent Kernel and showing that the so-called "NTK regime" (linear regime) cannot capture the performance of deep Neural Networks. More precisely, this regime becomes degenerate in the limit of infinite depth and its performance in classification tasks matches that of a random classifier.
Using an analysis of gradient back-propagation, principled guidelines for neural network pruning (model compression) which avoid the 'layer-collapse' issue (when one layer is fully pruned) were derived. Finally, we have shown that the created algorithm SBP-SR, formulated exclusively for residual neural networks, achieves state of the art performance on many datasets, among algorithms that consider pruning at initialization.

## 6.2 Limitations and open questions

*I began my doctoral journey with some questions that I aimed to address, and I am ending it with many new ones to go.*

In this section, we present a critical point of view of the results included in this thesis. We discuss the limitations of the theories and concepts used in our analysis and the ensuing questions that arise from our results.

### 6.2.1 The Edge of Chaos

**Characterization of the EOC.** In Chapter 2, we characterized the EOC for the activation functions that belong to class $\mathcal{A}$. We have also shown that other activation functions, such as S-Softplus (shifted Softplus), have a trivial EOC, and others, such as Swish, have a partial EOC. However, to the best of our knowledge, a full characterization of the set of activation functions that admit a non trivial EOC

remains an open question. We believe that such a characterization would have a significant impact on the choice of the activation functions used by practitioners.

**Non-Gaussian EOC.** The theory of the Edge of Chaos is based on two key ingredients: the infinite width limit and the Gaussian distribution. Thus, it is natural to question whether a different choice of the initialization distribution would impact the output function of the network. [Neal, 1995] proposed an $\alpha$-stable distribution to initialize the weights. This idea was motivated by the observation that as the width grows, the variance of the weights (i.e. $\sigma_w^2/N$, where $N$ is the width) converges to zero, and thus the weights converge to zero in $L_2$ norm. As a result, the contribution of each neuron in the next layer becomes negligible in the large width limit. Intuitively, by sampling from a heavy-tailed distribution, such as the $\alpha$-stable distribution, we would obtain large weights from the tail of the distribution more frequently compared to the Gaussian distribution. In the case of a single layer network, [Neal, 1995] conjectures that having some large weights in the hidden layer would give the corresponding neurons a significant contribution to the next layer. This conjecture can be generalized to the multilayer case.
The extension of the EOC theory to such distributions is not straightforward since the second moment (and thus the covariance) is not defined. However, in the case of $\alpha$-stable distributions, there are some alternatives for the covariance. One is the *covariation* which is similar to the covariance but with exponentiated random variables. At this point, it is not clear whether the covariation kernel would also be degenerate in the infinite depth limit; however, we believe that an extension of the EOC theory to heavy-tailed distributions would be a major contribution to the signal propagation theory at initialization.

**Beyond the EOC.** We have shown in Chapter 2 that the EOC initialization is beneficial for the propagation of the correlation and the backpropagation of the gradients. Particularly, the EOC initialization preserves the trace of the covariance matrix of the gradients as they backpropagate through the network at initialization. This has a direct impact on the first step of any gradient-based optimization

algorithm. However, it is still not clear how the EOC influences the whole training procedure. We conjecture that this might be related to some properties of local minima. Indeed, in our experiments, the EOC initialization always yields better performance compared to the Ordered/Chaotic phase, which might indicate that the EOC provides a "good" quality local minima. More research is needed in this topic to fully understand how the EOC impacts the training of finite-width neural networks.

### 6.2.2 The Neural Tangent Kernel

The theory of signal propagation yields tractable formulas for the NTK kernel in the infinite width limit. In this limit, the NTK is deterministic and depends only on the initialization hyperparameters. This is generally not the case for finite width neural networks (the NTK for a randomly initialized finite width network is generally random as it depends on the initialization weights), especially if the depth $L$ is of the same order as the width $N$. Indeed, in the simple case of an FFNN with ReLU activation function, [Hanin and Nica, 2019] showed that the corresponding NTK is random in the limit of large $L$ and $N$ with fixed ratio $\gamma = L/N$. These findings suggest that the NTK regime cannot capture the gradient flow dynamics in when the depth of the same order as the width. An interesting phenomenon that occurs in this case is that of *feature learning*, which simply means that the NTK also changes as we train the network; this implies that the model is also learning the features of the NTK kernel. There is growing consensus amongst researchers that feature learning regimes are better suited to describe what happens during training. In a recent work by [Yang and Hu, 2020], the authors showed that the feature learning regime yields better generalization compared to the NTK regime. Similarly, [Baratin et al., 2021] empirically showed that the NTK of a deep neural network learns features that align with the data labels. The dynamics of feature learning are yet to be fully understood, and we believe that this framework is the next promising topic in the analysis of deep neural networks training.

### 6.2.3 Stable ResNet

**Choice of scaling factors.** In Chapter 4, we have shown that a simple scaling of the residual blocks preserves the universality of the NNGP/NTK, even in the infinite depth limit. This can be achieved using any scaling factors such that $\lim_{L\to\infty} \sum_{l=1}^{L} \lambda_{l,L}^2 < \infty$. This condition is satisfied by a variety of scaling sequences; the uniform and the decreasing scaling factors introduced in Chapter 4 are examples of such sequences. Our experiments show that the decreasing scaling is often better than the uniform scaling. This can be explained by an RKHS hierarchy argument, i.e. the RKHS of the NNGP/NTK with uniform scaling is included in that of NNGP/NTK with the decreasing scaling. We are currently investigating this topic, as we believe it is a natural extension to Chapter 4.

**Stable ResNet and Feature learning.** Our theoretical results on Stable ResNet are valid in the infinite width limit (NTK/NNGP). In this limit, the NTK is fixed at initialization, and thus, the NTK features are also fixed before the training. In the case of finite width, we have shown empirically that Stable ResNet are superior to standard ResNet in terms of performance when trained with Stochastic Gradient Descent on a variety of large scale datasets. We know from section 6.2.2 that feature learning occurs when training finite width neural networks. However, it remains to be determined whether the scaling factors impact the feature learning, e.g. does introducing the scaling factors make feature learning more efficient?

# Bibliography

[Alvarez and Salzmann, 2017] Alvarez, J. M. and Salzmann, M. (2017). Compression-aware training of deep networks. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, Advances in Neural Information Processing Systems 30, pages 856–867. Curran Associates, Inc.

[Baratin et al., 2021] Baratin, A., George, T., Laurent, C., Devon Hjelm, R., Lajoie, G., and Vincent, P.and Lacoste-Julien, S. (2021). Implicit regularization via neural feature alignment. In Proceedings of The 24th International Conference on Artificial Intelligence and Statistics, pages 2269–2277.

[Carreira-Perpiñán and Idelbayev, 2018] Carreira-Perpiñán, M. and Idelbayev, Y. (2018). Learning-compression algorithms for neural net pruning. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[Dong et al., 2017] Dong, X., Chen, S., and Pan, S. (2017). Learning to prune deep neural networks via layer-wise optimal brain surgeon. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, Advances in Neural Information Processing Systems, volume 30, pages 4857–4867. Curran Associates, Inc.

[Du et al., 2019] Du, S., Zhai, X., Poczos, B., and Singh, A. (2019). Gradient descent provably optimizes over-parameterized neural networks. In 7th International Conference on Learning Representations.

[Frankle and Carbin, 2019] Frankle, J. and Carbin, M. (2019). The lottery ticket hypothesis: Finding sparse, trainable neural networks. In 7th International Conference on Learning Representations.

[Goodfellow et al., 2016] Goodfellow, I., Bengio, Y., and Courville, A. (2016). Deep Learning. MIT Press. `http://www.deeplearningbook.org`.

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

[Hassibi et al., 1993] Hassibi, B., Stork, D., and Gregory, W. (1993). Optimal brain surgeon and general network pruning. In IEEE International Conference on Neural Networks, pages 293 – 299 vol.1.

[Hayou et al., 2019] Hayou, S., Doucet, A., and Rousseau, J. (2019). On the impact of the activation function on deep neural networks training. International Conference on Machine Learning.

[Jacot et al., 2018] Jacot, A., Gabriel, F., and Hongler, C. (2018). Neural tangent kernel: Convergence and generalization in neural networks. In Bengio, S., Wallach,

H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., and Garnett, R., editors, Advances in Neural Information Processing Systems, volume 31, pages 8571–8580. Curran Associates, Inc.

[LeCun et al., 1990] LeCun, Y., Denker, J., and Solla, S. (1990). Optimal brain damage. In Touretzky, D., editor, Advances in Neural Information Processing Systems, volume 2, pages 598–605. Morgan-Kaufmann.

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

[Lee et al., 2019] Lee, J., Xiao, L., Schoenholz, S., Bahri, Y., Sohl-Dickstein, J., and Pennington, J. (2019). Wide neural networks of any depth evolve as linear models under gradient descent. 32:8572–8583.

[Lee et al., 2020] Lee, N., Ajanthan, T., Gould, S., and Torr, P. H. S. (2020). A signal propagation perspective for pruning neural networks at initialization. In 8th International Conference on Learning Representations.

[Lee et al., 2018b] Lee, N., Ajanthan, T., and Torr, P. H. (2018b). Snip: Single-shot network pruning based on connection sensitivity. In 6th International Conference on Learning Representations.

[Li et al., 2018] Li, H., Kadav, A., Durdanovic, I., Samet, H., and Graf, H. (2018). Pruning filters for efficient convnets. In 6th International Conference on Learning Representations.

[Li et al., 2020] Li, Y., Gu, S., Zhang, K., Van Gool, L., and Timofte, R. (2020). Dhp: Differentiable meta pruning via hypernetworks. arXiv preprint arXiv:2003.13683.

[Liu et al., 2019] Liu, Z., Mu, H., Zhang, X., Guo, Z., Yang, X., Cheng, K.-T., and Sun, J. (2019). Metapruning: Meta learning for automatic neural network channel pruning. In Proceedings of the IEEE International Conference on Computer Vision, pages 3296–3305.

[Louizos et al., 2018] Louizos, C., Welling, M., and Kingma, D. (2018). Learning sparse neural networks through $l_0$ regularization. In 6th International Conference on Learning Representations.

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

[Montufar et al., 2014] Montufar, G., Pascanu, R., Cho, K., and Bengio, Y. (2014). On the number of linear regions of deep neural networks. 27:2924–2932.

[Mozer and Smolensky, 1989] Mozer, M. and Smolensky, P. (1989). Skeletonization: A technique for trimming the fat from a network via relevance assessment. In Touretzky, D., editor, Advances in Neural Information Processing Systems, volume 1, pages 107–115. Morgan-Kaufmann.

[Neal, 1995] Neal, R. (1995). Bayesian Learning for Neural Networks, volume 118. Springer Science & Business Media.

[Neyshabur et al., 2019] Neyshabur, B., Li, Z., Bhojanapalli, S., LeCun, Y., and Srebro, N. (2019). The role of over-parametrization in generalization of neural networks. In 7th International Conference on Learning Representations.

[Nguyen and Hein, 2018] Nguyen, Q. and Hein, M. (2018). Optimization landscape and expressivity of deep CNNs. International Conference on Machine Learning.

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

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

[Tanaka et al., 2020] Tanaka, H., Kunin, D., Yamins, D. L., and Ganguli, S. (2020). Pruning neural networks without any data by iteratively conserving synaptic flow. arXiv preprint arXiv:2006.05467.

[Wang et al., 2020] Wang, C., Zhang, G., and Grosse, R. (2020). Picking winning tickets before training by preserving gradient flow. In 8th International Conference on Learning Representations.

[Xiao et al., 2018] Xiao, L., Bahri, Y., Sohl-Dickstein, J., S. Schoenholz, S., and Pennington, P. (2018). Dynamical isometry and a mean field theory of cnns: How to train 10,000-layer vanilla convolutional neural networks. International Conference on Machine Learning.

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

[Yang and Hu, 2020] Yang, G. and Hu, E. (2020). Feature learning in infinite-width neural networks. arXivn preprint 2011.14522.

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

[Zhang et al., 2016] Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. In 5th International Conference on Learning Representations.