Proceedings of Machine Learning Research vol 107:921–954, 2021 2nd Annual Conference on Mathematical and Scientific Machine Learning

Kernel-Based Smoothness Analysis of Residual Networks

Tom Tirer TOMTIRER@MAIL.TAU.AC.IL
Tel Aviv University

Joan Bruna BRUNA@CIMS.NYU.EDU
New York University

Raja Giryes RAJA@TAUEX.TAU.AC.IL
Tel Aviv University

Editors: Joan Bruna, Jan S Hesthaven, Lenka Zdeborova

Abstract

A major factor in the success of deep neural networks is the use of sophisticated architectures rather than the classical multilayer perceptron (MLP). Residual networks (ResNets) stand out among these powerful modern architectures. Previous works focused on the optimization advantages of deep ResNets over deep MLPs. In this paper, we show another distinction between the two models, namely, a tendency of ResNets to promote smoother interpolations than MLPs. We analyze this phenomenon via the neural tangent kernel (NTK) approach. First, we compute the NTK for a considered ResNet model and prove its stability during gradient descent training. Then, we show by various evaluation methodologies that for ReLU activations the NTK of ResNet, and its kernel regression results, are smoother than the ones of MLP. The better smoothness observed in our analysis may explain the better generalization ability of ResNets and the practice of moderately attenuating the residual blocks.

Keywords: Neural tangent kernel, residual networks, multilayer perceptron, kernel methods

1. Introduction

Deep neural networks have led to a major improvement in various fields. The advance in the network performance is tightly related to the introduction of various novel architectures (Krizhevsky et al., 2012; Simonyan and Zisserman, 2015; He et al., 2016; Huang et al., 2017; Tan and Le, 2019). A prominent model among them, which has led to a major leap in performance, is the deep residual network, known also as ResNet (He et al., 2016). It has introduced the usage of the skip connection, i.e., an identity path in the network that adds to the output features of a given layer its input features. This simple change enables effectively training much deeper networks, which eventually leads to improved results.

Different efforts were dedicated to explaining the success of ResNets. These mainly focused on the optimization aspect of ResNets, namely, e.g., claiming that it is "easier" to train a network with skip-connections as it enjoys a better loss surface (Li et al., 2018) or that ResNets overcome the problem of vanishing gradients (Veit et al., 2016). Yet, analyzing deep networks rather than shallow ones has remained a major challenge.

Recently, Jacot et al. (2018) have shown that, under certain conditions (one of them is strong over-parameterization), training a deep neural network with gradient descent can be characterized by kernel regression with the neural tangent kernel (NTK). Essentially, this approach can be understood as a linearization (first-order Taylor series expansion) of the network's output with respect to its parameters around the initialization. It should be noted that the NTK is not the nominal regime of the non-linear learning capabilities of deep neural networks (e.g., like classical kernels (Schölkopf et al., 2002), its feature mapping does not adapt to the data and it has not been shown to reach the performance of powerful deep networks (Chizat et al., 2019)). Yet, the NTK can be used to identify or provide tractable analyses for phenomena that are also observed in other deep learning settings, such as achieving zero training loss (Jacot et al., 2018; Chizat et al., 2019; Lee et al., 2019; Arora et al., 2019b) and faster learning of lower frequencies (Basri et al., 2019).

As the NTK formulas depend on the network architecture, most of the NTK works consider the classical multilayer perceptron (MLP), a plain feed-forward network with fully connected layers (Jacot et al., 2018; Chizat et al., 2019; Lee et al., 2019; Arora et al., 2019a; Basri et al., 2019; Bietti and Mairal, 2019; Williams et al., 2019; Geifman et al., 2020; Chen and Xu, 2020; Bietti and Bach, 2020). Yet, some recent papers compute NTK expressions for other architectures (Arora et al., 2019b; Yang, 2019a, 2020; Huang et al., 2020; Alemohammad et al., 2020; Hron et al., 2020).

Contribution. In this paper, we develop the NTK for a ResNet model. After obtaining the formulas for the infinite width limit at initialization, we prove the stability of empirical NTK during training with gradient descent (and other common NTK assumptions), which implies that the trained ResNet model is indeed characterized by its NTK. Note that proving stability during training is the key result that allows NTK-based analysis of neural networks. Yet, it is missing in a recent paper (Huang et al., 2020) that also considered a resembling ResNet model (more details are in Section 3).

By comparing the ResNet NTK and the MLP NTK for ReLU activations, we find that ResNet promotes smoother interpolations than MLP (without using any explicit regularization), which adds to other advantages of ResNet described in previous works. Our smoothness findings are based on different evaluation methodologies, such as visualizing the kernel of each model (which is data-independent), comparing uniform upper bounds on the norm of the models' Jacobians after training, and comparing kernel regression results (specifically, interpolations) for the different NTKs. In the latter methodology, we use approximate $\mathcal{L}^2$-norm of the second derivative of a function as a quantitative measure for its smoothness. Finally, we show that for ReLU-based networks the smoothness advantage of ResNet is also observed outside the NTK regime.

Our analysis may be related to the better generalization ability of ResNets over MLP, as there is prior work that connects fitting the training data with a smoother function to better generalization error (under some smoothness assumption on the target function) (Lu et al., 2019; Giryes, 2020; Xie et al., 2020). We also show that the smoothness distinction between the two models can be increased by moderately attenuating the residual blocks when summing them with the skip connections. Indeed, this practice has been shown to improve the training and generalization robustness of ResNets in a recent empirical classification study (Zhang et al., 2019) and its follow-up work (Yang et al., 2020).

2. Background and Related Work

This section presents the NTK of a plain MLP with some of its results. Consider an MLP model with L hidden layers, input $\mathbf{x} \in \mathbb{R}^d$, parameter vector $\boldsymbol{\theta} := \mathrm{vec}(\{\mathbf{W}^{(\ell)}\})$, and output $\mathbf{f}(\mathbf{x}; \boldsymbol{\theta}) \in \mathbb{R}^k$, given by

$$\begin{aligned}
&\mathbf{g}^{(\ell)} = \frac{\sigma_w}{\sqrt{n_{\ell-1}}}\mathbf{W}^{(\ell)}\mathbf{x}^{(\ell-1)}, \qquad \ell = 1, \dots, L \\
&\mathbf{x}^{(\ell)} = \phi(\mathbf{g}^{(\ell)}), \qquad \ell = 1, \dots, L \\
&\mathbf{x}^{(0)} = \mathbf{x}, \qquad \mathbf{f}(\mathbf{x}, \boldsymbol{\theta}) = \mathbf{g}^{(L+1)} = \frac{\sigma_w}{\sqrt{n_L}}\mathbf{W}^{(L+1)}\mathbf{x}^{(L)},
\end{aligned} \tag{1}$$

where $\phi(\cdot)$ is an element-wise activation function, σ_w is a positive hyperparameter that scales the standard deviation of $\{\mathbf{W}^{(\ell)}\}$, $\mathbf{W}^{(\ell)} \in \mathbb{R}^{n_\ell \times n_{\ell-1}}$, $n_0 = d$, $n_{L+1} = k$, and all the weights are initialized by the standard normal distribution $W_{ij}^{(\ell)} \sim \mathcal{N}(0, 1)$. It is assumed that the input is bounded $\|\mathbf{x}\|_2 \le B$.

At initialization, when $n_1, \dots, n_L \to \infty$ each pre-activation $g_i^{(\ell)}(\mathbf{x})$, and thus also $f_i(\mathbf{x}) = g_i^{(L+1)}(\mathbf{x})$, is a stochastic Gaussian Process (GP) with zero mean (Neal, 2012; Lee et al., 2017). Denote the GP kernel (covariance) of this process by $K^{(L+1)}(\mathbf{x}, \tilde{\mathbf{x}}) := \mathbb{E}_{\boldsymbol{\theta}}\left[g_i^{(L+1)}(\mathbf{x})g_i^{(L+1)}(\tilde{\mathbf{x}})\right]$ (note its independence of the entry index i). We have that

$$f_i(\mathbf{x})f_i(\tilde{\mathbf{x}}) \xrightarrow{n_{1:L}\to\infty} K^{(L+1)}(\mathbf{x}, \tilde{\mathbf{x}}),$$

where the limit should be interpreted in the almost surely sense. Since $\mathbb{E}_{\boldsymbol{\theta}}\left[g_i^{(L+1)}(\mathbf{x})g_j^{(L+1)}(\tilde{\mathbf{x}})\right] = 0$ for $i \ne j$, for multidimensional output, we simply have

$$\mathbf{f}(\mathbf{x})\mathbf{f}^\top(\tilde{\mathbf{x}}) \xrightarrow{n_{1:L}\to\infty} K^{(L+1)}(\mathbf{x}, \tilde{\mathbf{x}}) \otimes \mathbf{I}_k,$$

where $\otimes$ is the Kronecker product. The GP kernel of the MLP above can be computed using the following recursive expression (Jacot et al., 2018; Lee et al., 2017; Yang, 2019b)

$$\begin{aligned}
K^{(L+1)}(\mathbf{x}, \tilde{\mathbf{x}}) &= \sigma_w^2 T\left(\begin{bmatrix} K^{(L)}(\mathbf{x}, \mathbf{x}) & K^{(L)}(\mathbf{x}, \tilde{\mathbf{x}}) \\ K^{(L)}(\mathbf{x}, \tilde{\mathbf{x}}) & K^{(L)}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}) \end{bmatrix}\right), \\
K^{(1)}(\mathbf{x}, \tilde{\mathbf{x}}) &= \frac{\sigma_w^2}{d}\mathbf{x}^\top\tilde{\mathbf{x}},
\end{aligned} \tag{2}$$

where $T(\boldsymbol{\Sigma}) := \mathbb{E}_{(u,v)\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma})}\left[\phi(u)\phi(v)\right]$.

Recently, a new type of kernel has received much attention, namely, the NTK (Jacot et al., 2018), which is defined as $\Theta^{(L+1)}(\mathbf{x}, \tilde{\mathbf{x}}) := \mathbb{E}_{\boldsymbol{\theta}}\left\langle \frac{\partial f_i(\mathbf{x};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}, \frac{\partial f_i(\tilde{\mathbf{x}};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} \right\rangle$ where $n_{1:L} \to \infty$. Similarly to the GP kernel, it can be shown that at initialization $\left\langle \frac{\partial f_i(\mathbf{x};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}}, \frac{\partial f_i(\tilde{\mathbf{x}};\boldsymbol{\theta})}{\partial\boldsymbol{\theta}} \right\rangle \xrightarrow{n_{1:L}\to\infty} \Theta^{(L+1)}(\mathbf{x}, \tilde{\mathbf{x}})$ (Jacot et al., 2018; Arora et al., 2019b; Yang, 2019a, 2020). (The extension to multidimensional output is done again by Kronecker product with $\mathbf{I}_k$). Note that the significant impact of NTK is mainly due to the fact that it can be used to characterize DNN training with gradient flow or gradient descent (with small enough learning rate) (Jacot et al., 2018; Lee et al., 2019; Arora et al., 2019b).

Specifically, given the training data $\mathcal{D} = (\mathcal{X}, \mathcal{Y})$ (where $(\mathcal{X}, \mathcal{Y}) = ((\mathbf{x}_1, \mathbf{y}_1), \dots, (\mathbf{x}_{|\mathcal{X}|}, \mathbf{y}_{|\mathcal{X}|}))$ are the training samples $\{\mathbf{x}_i\} \in \mathbb{R}^d$ and their associated labels $\{\mathbf{y}_i\} \in \mathbb{R}^k$) and a loss function $\ell(\cdot, \cdot) : \mathbb{R}^k \times \mathbb{R}^k \to \mathbb{R}$, consider learning $\boldsymbol{\theta}$ by minimizing the empirical loss $\mathcal{L} = \sum_{(\mathbf{x}_i, \mathbf{y}_i)\in\mathcal{D}} \ell(\mathbf{f}(\mathbf{x}_i; \boldsymbol{\theta}), \mathbf{y}_i)$ using gradient descent with learning rate η. This can be written in continuous time (for simplicity) as

$\dot{\boldsymbol{\theta}}_t = -\eta \frac{\partial \mathbf{f}(\mathcal{X};\boldsymbol{\theta}_t)}{\partial \boldsymbol{\theta}}^\top \nabla_{\mathbf{f}(\mathcal{X};\boldsymbol{\theta}_t)}\mathcal{L}$, where $\mathbf{f}(\mathcal{X};\boldsymbol{\theta}_t) = \mathrm{vec}(\{\mathbf{f}(\mathbf{x}_i;\boldsymbol{\theta}_t)\}_{\mathbf{x}_i\in\mathcal{X}}) \in \mathbb{R}^{k|\mathcal{X}|\times 1}$. In the function space, we have

$$\begin{aligned}\dot{\mathbf{f}}(\mathcal{X};\boldsymbol{\theta}_t) &= \frac{\partial \mathbf{f}(\mathcal{X};\boldsymbol{\theta}_t)}{\partial \boldsymbol{\theta}}\dot{\boldsymbol{\theta}}_t \\ &= -\eta \frac{\partial \mathbf{f}(\mathcal{X};\boldsymbol{\theta}_t)}{\partial \boldsymbol{\theta}}\frac{\partial \mathbf{f}(\mathcal{X};\boldsymbol{\theta}_t)}{\partial \boldsymbol{\theta}}^\top \nabla_{\mathbf{f}(\mathcal{X};\boldsymbol{\theta}_t)}\mathcal{L}.\end{aligned} \tag{3}$$

Under appropriate conditions, it can be shown that $\frac{\partial \mathbf{f}(\mathcal{X};\boldsymbol{\theta}_t)}{\partial \boldsymbol{\theta}}\frac{\partial \mathbf{f}(\mathcal{X};\boldsymbol{\theta}_t)}{\partial \boldsymbol{\theta}}^\top \xrightarrow{n_{1:L}\to\infty} \boldsymbol{\Theta}\otimes\mathbf{I}_k$, where $\boldsymbol{\Theta}\in\mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}$ with $\Theta_{ij} = \Theta^{(L+1)}(\mathbf{x}_i,\mathbf{x}_j)$. In other words, the dynamics of the output function in (3) turns into a simple ODE with respect to $\mathbf{f}(\mathcal{X};\boldsymbol{\theta}_t)$ based on the constant NTK. For squared ℓ_2 loss this is even a linear ODE with a close-from solution. In this case, fitting scalar labels $\{y_i\}$ is reduced to ℓ_2 kernel regression, which has the following closed-form solution

$$f(\mathbf{x}) = \mathbf{k}(\mathbf{x})^\top \boldsymbol{\Theta}^{-1}\mathbf{y}, \tag{4}$$

where $k_i(\mathbf{x}) = \Theta^{(L+1)}(\mathbf{x},\mathbf{x}_i)$ and $\mathbf{y} = [y_1,\ldots,y_{|\mathcal{D}|}]^\top$. A more general study on linear behaviour of non-linear models, which takes into account under-parameterized models and the effect of the scaling used in initialization, appears in (Chizat et al., 2019).

Finally, the NTK of the MLP model in (1) can be computed using the following recursive expression (Jacot et al., 2018)

$$\begin{aligned}\Theta^{(L+1)}(\mathbf{x},\tilde{\mathbf{x}}) &= K^{(L+1)}(\mathbf{x},\tilde{\mathbf{x}}) + \Theta^{(L)}(\mathbf{x},\tilde{\mathbf{x}})\cdot\sigma_w^2\dot{T}\left(\begin{bmatrix}K^{(L)}(\mathbf{x},\mathbf{x}) & K^{(L)}(\mathbf{x},\tilde{\mathbf{x}})\\ K^{(L)}(\mathbf{x},\tilde{\mathbf{x}}) & K^{(L)}(\tilde{\mathbf{x}},\tilde{\mathbf{x}})\end{bmatrix}\right), \\ \Theta^{(1)}(\mathbf{x},\tilde{\mathbf{x}}) &= K^{(1)}(\mathbf{x},\tilde{\mathbf{x}}),\end{aligned} \tag{5}$$

where $\dot{T}(\boldsymbol{\Sigma}) := \mathbb{E}_{(u,v)\sim\mathcal{N}(\mathbf{0},\boldsymbol{\Sigma})}[\phi'(u)\phi'(v)]$. Note that $T(\boldsymbol{\Sigma})$ and $\dot{T}(\boldsymbol{\Sigma})$ have closed-form expressions for the ReLU and erf activation functions. For completeness, their expressions for ReLU, which are due to (Cho and Saul, 2009), are provided in Appendix D.

3. NTK for ResNet

We turn now to develop the ResNet NTK. Consider a ResNet model with L non-linear hidden layers, input $\mathbf{x}\in\mathbb{R}^d$, parameter vector $\boldsymbol{\theta} := \mathrm{vec}(\mathbf{w}^{(L+1)}, \{\mathbf{W}^{(\ell)}\}, \{\mathbf{V}^{(\ell)}\}, \mathbf{U})$, and output $f(\mathbf{x};\boldsymbol{\theta})\in\mathbb{R}$ given by

$$\begin{aligned}\mathbf{g}^{(\ell)} &= \frac{\sigma_w}{\sqrt{n}}\mathbf{W}^{(\ell)}\mathbf{x}^{(\ell-1)}, \quad \ell = 1,\ldots,L \\ \mathbf{x}^{(\ell)} &= \mathbf{x}^{(\ell-1)} + \alpha\frac{\sigma_v}{\sqrt{n}}\mathbf{V}^{(\ell)}\phi(\mathbf{g}^{(\ell)}), \quad \ell = 1,\ldots,L \\ \mathbf{x}^{(0)} &= \frac{1}{\sqrt{d}}\mathbf{U}\mathbf{x}, \quad f(\mathbf{x},\boldsymbol{\theta}) = g^{(L+1)} = \frac{\sigma_w}{\sqrt{n}}\mathbf{w}^{(L+1)\top}\mathbf{x}^{(L)},\end{aligned} \tag{6}$$

where $\phi(\cdot)$ is an element-wise activation function, σ_w and σ_v are positive hyperparameters that scale the standard deviation of $\{\mathbf{W}^{(\ell)}\}$ and $\{\mathbf{V}^{(\ell)}\}$, respectively, and α is a positive hyperparameter that weighs the residual block. $\mathbf{W}^{(\ell)}, \mathbf{V}^{(\ell)}\in\mathbb{R}^{n\times n}$, $\mathbf{w}^{(L+1)}\in\mathbb{R}^n$, $\mathbf{U}\in\mathbb{R}^{n\times d}$, and all the weights are initialized by the standard normal distribution $w_i^{(L+1)}, W_{ij}^{(\ell)}, V_{ij}^{(\ell)}, U_{ij} \sim \mathcal{N}(0,1)$. It is assumed that the input is bounded $\|\mathbf{x}\|_2 \leq B$.

A few remarks are in place. First, we assume scalar output for simplification, and the extension to multidimensional output is straightforward as shown in Section 2. Second, note that we lift the input dimension from $\mathbb{R}^d$ to $\mathbb{R}^n$ using $\mathbf{x}^{(0)} = \frac{1}{\sqrt{d}}\mathbf{U}\mathbf{x}$. This lifting is unavoidable as the NTK analysis requires that the width of all intermediate layers approaches infinity. Third, the weights $\{\mathbf{V}^{(\ell)}\}$ are necessary for the proof technique and for obtaining closed-form expressions for the ResNet NTK. Specifically, $\mathbf{V}^{(\ell)}$ breaks the correlation between $\mathbf{x}^{(\ell-1)}$ and $\phi(\mathbf{g}^{(\ell)})$ which leads to (relatively nice) formulas for the GP kernel and NTK with closed-form expressions in the case of ReLU activations. Also, as explained in Appendix A, the multiplication of $\phi(\mathbf{g}^{(\ell)})$ by $\mathbf{V}^{(\ell)}$ allows us to use general convergence results from (Yang, 2019b,a). In contrast, ResNet models that do not multiply the nonlinear activations by $\{\mathbf{V}^{(\ell)}\}$ get complicated recursive equations for their kernel with *no closed-form analytical expressions even for ReLU activations* (Du et al., 2019), or do not provide kernel expressions at all (Zhang, 2019). Thus, they have no possibility to observe the shape of this kernel and obtain its kernel regression results (contrary to our ResNet NTK).

Lastly, a similar ResNet model is considered by (Huang et al., 2020). Yet, they assume that the weights of the first and last layers are fixed, and they do not include a proof that when training this ResNet model with gradient descent/flow the limiting NTK stays the same as the one in the initialization (i.e., $\frac{\partial \mathbf{f}(\mathcal{X};\boldsymbol{\theta}_t)}{\partial \boldsymbol{\theta}} \frac{\partial \mathbf{f}(\mathcal{X};\boldsymbol{\theta}_t)}{\partial \boldsymbol{\theta}}^\top \xrightarrow{n\to\infty} \boldsymbol{\Theta}$). Note that this missing result (proven here in Theorem 5, using Lemma 4) is perhaps the most important property of NTK-based analysis of neural networks. Another difference between the works is that we use the derived NTK to compare the ResNet with MLP in terms of smoothness for a similar *fixed* depth (i.e., fixed L), while Huang et al. (2020) examine the NTKs expressions for $L \to \infty$. In fact, note that the NTK regime for MLP requires L to be finite (Hanin and Nica, 2019; Littwin et al., 2020).

Denote the *empirical* (random, finite-width) GP kernel and NTK at initialization by $\hat{K}_0^{(L+1)}(\mathbf{x}, \tilde{\mathbf{x}}) := f(\mathbf{x};\boldsymbol{\theta}_0) f(\tilde{\mathbf{x}};\boldsymbol{\theta}_0)$ and $\hat{\Theta}_0^{(L+1)}(\mathbf{x}, \tilde{\mathbf{x}}) := \left\langle \frac{\partial f(\mathbf{x};\boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}}, \frac{\partial f(\tilde{\mathbf{x}};\boldsymbol{\theta}_0)}{\partial \boldsymbol{\theta}} \right\rangle$, respectively. Our first results state the GP kernel and NTK at initialization when $n \to \infty$. While these results are asymptotic, in Section 4 we present numerical experiments where the behavior of practical finite width networks correlates with the asymptotic NTK analysis. Yet, non-asymptotic results may be obtained using concentration bounds, as done for other kernel approximation techniques (Sriperumbudur and Szabó, 2015; Koppel et al., 2019).

Theorem 1 (GP kernel at initialization) *Consider the ResNet model in* (6). *We have* $\hat{K}_0^{(L+1)}(\mathbf{x}, \tilde{\mathbf{x}}) \xrightarrow{n\to\infty} K^{(L+1)}(\mathbf{x}, \tilde{\mathbf{x}}) := \mathbb{E}_{\boldsymbol{\theta}}\left[f(\mathbf{x};\boldsymbol{\theta}) f(\tilde{\mathbf{x}};\boldsymbol{\theta})\right]$, *where* $K^{(L+1)}(\mathbf{x}, \tilde{\mathbf{x}})$ *can be computed recursively as following:*

$$K^{(L+1)}(\mathbf{x}, \tilde{\mathbf{x}}) = K^{(L)}(\mathbf{x}, \tilde{\mathbf{x}}) + \alpha^2 \sigma_v^2 \sigma_w^2 T\left(\begin{bmatrix} K^{(L)}(\mathbf{x}, \mathbf{x}) & K^{(L)}(\mathbf{x}, \tilde{\mathbf{x}}) \\ K^{(L)}(\mathbf{x}, \tilde{\mathbf{x}}) & K^{(L)}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}) \end{bmatrix}\right), \tag{7}$$

$$K^{(1)}(\mathbf{x}, \tilde{\mathbf{x}}) = \frac{\sigma_w^2}{d} \mathbf{x}^\top \tilde{\mathbf{x}}.$$

Theorem 2 (NTK at initialization) *Consider the ResNet model in* (6) *and let the element-wise non-linearities be bounded uniformly by* $\mathrm{e}^{(cx^2-\epsilon)}$ *for some* $c, \epsilon > 0$. *We have that* $\hat{\Theta}_0^{(L+1)}(\mathbf{x}, \tilde{\mathbf{x}}) \xrightarrow{n\to\infty}$

$\Theta^{(L+1)}(\mathbf{x},\tilde{\mathbf{x}}) := \mathbb{E}_{\boldsymbol{\theta}}\left\langle \frac{\partial f(\mathbf{x};\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}, \frac{\partial f(\tilde{\mathbf{x}};\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right\rangle$, where $\Theta^{(L+1)}(\mathbf{x},\tilde{\mathbf{x}})$ is given by

$$\begin{aligned}\Theta^{(L+1)}(\mathbf{x},\tilde{\mathbf{x}}) &= K^{(L+1)}(\mathbf{x},\tilde{\mathbf{x}}) + \Pi^{(0)}(\mathbf{x},\tilde{\mathbf{x}}) \cdot K^{(1)}(\mathbf{x},\tilde{\mathbf{x}}) \\ &\quad + \alpha^2 \sum_{\ell=1}^{L} \Pi^{(\ell)}(\mathbf{x},\tilde{\mathbf{x}}) \cdot \left(\Sigma^{(\ell+1)}(\mathbf{x},\tilde{\mathbf{x}}) + K^{(\ell)}(\mathbf{x},\tilde{\mathbf{x}}) \cdot \dot{\Sigma}^{(\ell+1)}(\mathbf{x},\tilde{\mathbf{x}}) \right) \end{aligned} \tag{8}$$

such that

$$\begin{aligned}\Sigma^{(\ell+1)}(\mathbf{x},\tilde{\mathbf{x}}) &:= \sigma_v^2 \sigma_w^2 T\left(\begin{bmatrix} K^{(\ell)}(\mathbf{x},\mathbf{x}) & K^{(\ell)}(\mathbf{x},\tilde{\mathbf{x}}) \\ K^{(\ell)}(\mathbf{x},\tilde{\mathbf{x}}) & K^{(\ell)}(\tilde{\mathbf{x}},\tilde{\mathbf{x}}) \end{bmatrix} \right), \\ \dot{\Sigma}^{(\ell+1)}(\mathbf{x},\tilde{\mathbf{x}}) &:= \sigma_v^2 \sigma_w^2 \dot{T}\left(\begin{bmatrix} K^{(\ell)}(\mathbf{x},\mathbf{x}) & K^{(\ell)}(\mathbf{x},\tilde{\mathbf{x}}) \\ K^{(\ell)}(\mathbf{x},\tilde{\mathbf{x}}) & K^{(\ell)}(\tilde{\mathbf{x}},\tilde{\mathbf{x}}) \end{bmatrix} \right), \end{aligned} \tag{9}$$

$\{K^{(\ell)}(\mathbf{x},\tilde{\mathbf{x}})\}$ are given in (7), and $\{\Pi^{(\ell)}(\mathbf{x},\tilde{\mathbf{x}})\}$ can be computed using the following recursive expression

$$\begin{aligned}\Pi^{(\ell)}(\mathbf{x},\tilde{\mathbf{x}}) &= \Pi^{(\ell+1)}(\mathbf{x},\tilde{\mathbf{x}}) \left(1 + \alpha^2 \dot{\Sigma}^{(\ell+2)}(\mathbf{x},\tilde{\mathbf{x}})\right), \\ \Pi^{(L)}(\mathbf{x},\tilde{\mathbf{x}}) &= 1. \end{aligned} \tag{10}$$

The proofs of the theorems can be found in Appendix A. Theorems 1 and 2 provide kernels that are associated with the considered ResNet model. Both of these kernels can be used in applications of kernel methods (Schölkopf et al., 2002). Yet, in what follows we show that the special property of the NTK holds also for our ResNet. Namely, that under appropriate conditions, the limiting NTK stays constant even during gradient descent training of the ResNet. Therefore, as discussed below (3), the network function during and after training can be characterized by kernel regression with the NTK.

Let $\boldsymbol{\theta}_t$ denote the parameters at time step t. Given training data $\mathcal{D} = (\mathcal{X},\mathcal{Y})$ (where $(\mathcal{X},\mathcal{Y}) = ((\mathbf{x}_1,y_1),\ldots,(\mathbf{x}_{|\mathcal{X}|},y_{|\mathcal{X}|}))$ are the training samples $\{\mathbf{x}_i\} \in \mathbb{R}^d$ and their associated labels $\{y_i\} \in \mathbb{R}$), we make the following shorthand notations

$$\begin{aligned}\mathbf{f}(\boldsymbol{\theta}_t) &= \mathrm{vec}(\{f(\mathbf{x}_i;\boldsymbol{\theta}_t)\}_{\mathbf{x}_i \in \mathcal{X}}) \in \mathbb{R}^{|\mathcal{X}|}, \\ \mathbf{e}(\boldsymbol{\theta}_t) &= \mathbf{f}(\boldsymbol{\theta}_t) - \mathcal{Y} \in \mathbb{R}^{|\mathcal{X}|}, \\ \mathbf{J}(\boldsymbol{\theta}_t) &= \frac{\partial \mathbf{f}(\boldsymbol{\theta}_t)}{\partial \boldsymbol{\theta}} \in \mathbb{R}^{|\mathcal{X}| \times |\boldsymbol{\theta}|}. \end{aligned} \tag{11}$$

The empirical $|\mathcal{X}| \times |\mathcal{X}|$ NTK Gram matrix is defined as

$$\hat{\boldsymbol{\Theta}}_t := \hat{\boldsymbol{\Theta}}_t(\mathcal{X},\mathcal{X}) = \mathbf{J}(\boldsymbol{\theta}_t)\mathbf{J}(\boldsymbol{\theta}_t)^\top. \tag{12}$$

From Theorem 2 we have that $\hat{\boldsymbol{\Theta}}_0 \xrightarrow{n\rightarrow\infty} \boldsymbol{\Theta} \in \mathbb{R}^{|\mathcal{X}|\times|\mathcal{X}|}$ with $\Theta_{ij} = \Theta^{(L+1)}(\mathbf{x}_i,\mathbf{x}_j)$.

In Theorem 5 below, we show that when training the ResNet using the loss function $\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2}\sum_{(\mathbf{x}_i,y_i)\in\mathcal{D}}(f(\mathbf{x}_i;\boldsymbol{\theta}) - y_i)^2 = \frac{1}{2}\|\mathbf{e}(\boldsymbol{\theta})\|_2^2$ and gradient descent with small enough learning rate η, we get $\sup_t \|\hat{\boldsymbol{\Theta}}_t - \hat{\boldsymbol{\Theta}}_0\|_F = \mathcal{O}(\frac{1}{\sqrt{n}})$, which implies $\hat{\boldsymbol{\Theta}}_t \xrightarrow{n\rightarrow\infty} \boldsymbol{\Theta}$. To obtain the result $\sup_t \|\hat{\boldsymbol{\Theta}}_t - \hat{\boldsymbol{\Theta}}_0\|_F = \mathcal{O}(\frac{1}{\sqrt{n}})$ we extend the strategy of (Lee et al., 2019) from MLP to the ResNet. The extension is based on the following two lemmas.

Lemma 3 *Let $\mathbf{W}$ be an $m \times n$ random matrix whose entries are independent standard normal variables. Then for every $t \geq 0$, with probability at least $1 - 2\exp(-t^2/2)$ we have*

$$\sqrt{m} - \sqrt{n} - t \leq \lambda_{min}(\mathbf{W}) \leq \lambda_{max}(\mathbf{W}) \leq \sqrt{m} + \sqrt{n} + t, \tag{13}$$

where $\lambda_{min}(\mathbf{W})$ and $\lambda_{max}(\mathbf{W})$ denote the smallest and largest singular values of $\mathbf{W}$, respectively.

Lemma 4 *Consider the ResNet model in (6) initialized with $\boldsymbol{\theta}_0$, and assume that the activation function ϕ satisfies $|\phi(z)| \leq C_\phi |z|$, $|\phi'(z)| \leq C_\phi$ and $|\phi(z) - \phi(\tilde{z})|, |\phi'(z) - \phi'(\tilde{z})| \leq C_\phi |z - \tilde{z}|$, for some $C_\phi > 0$. Then, there exists a $K > 0$ (that does not depend on n) such that for every $C > 0$ and $n \gg C^2$, with high probability over the random initialization, the following holds for all $\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}} \in B(\boldsymbol{\theta}_0, C) := \{\boldsymbol{\theta} : \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2 \leq C\}$*

$$\begin{aligned} &\|\mathbf{J}(\boldsymbol{\theta})\|_F \leq K, \\ &\|\mathbf{J}(\boldsymbol{\theta}) - \mathbf{J}(\tilde{\boldsymbol{\theta}})\|_F \leq K \|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2. \end{aligned} \tag{14}$$

Lemma 3 is adopted from (Vershynin, 2010) (Corollary 5.35 there). It is used to prove Lemma 4 and will be used again later in this paper. Lemma 4 extends the "MLP version" that appears in (Lee et al., 2019) to the considered ResNet model. Yet, due to the structure of the ResNet that is much more complex than for MLP, the proof of Lemma 4 is more complex and is deferred to Appendix B. Using Lemma 4, we have the following theorem.

Theorem 5 (Stability of the NTK during training) *Consider the ResNet model in (6) with activation function that satisfies the conditions from Lemma 4. Assume that $\lambda_{min}(\boldsymbol{\Theta}) > 0$, the training set $\mathcal{D} = (\mathcal{X}, \mathcal{Y})$ is contained in some compact set and $\mathbf{x} \neq \tilde{\mathbf{x}}$ for all $\mathbf{x}, \tilde{\mathbf{x}} \in \mathcal{X}$. Then, for $\delta_0 > 0$ there exist $R_0 > 0$, N and $K > 1$, such that for every $n > N$ when applying gradient descent on $\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2}\|\mathbf{e}(\boldsymbol{\theta})\|_2^2$ with learning rate $\eta_0 < 2(\lambda_{min}(\boldsymbol{\Theta}) + \lambda_{max}(\boldsymbol{\Theta}))^{-1}$ the following holds with probability at least $1 - \delta_0$ over the random initialization*

$$\begin{aligned} &\|\mathbf{e}(\boldsymbol{\theta}_t)\|_2 \leq \left(1 - \frac{\eta_0}{3}\lambda_{min}(\boldsymbol{\Theta})\right)^t R_0, \\ &\sum_{j=1}^{t} \|\boldsymbol{\theta}_j - \boldsymbol{\theta}_{j-1}\|_2 \leq \frac{3KR_0}{\lambda_{min}(\boldsymbol{\Theta})}, \\ &\sup_t \|\hat{\boldsymbol{\Theta}}_t - \hat{\boldsymbol{\Theta}}_0\|_F = \frac{6K^3R_0}{\lambda_{min}(\boldsymbol{\Theta})} n^{-0.5}. \end{aligned} \tag{15}$$

Proof The proof is based on induction and applying Lemma 4 with $C = \frac{3KR_0}{\lambda_{min}(\boldsymbol{\Theta})}$. Essentially, it is an extension of Theorems G.1 and G.4 in (Lee et al., 2019) from MLP to the considered ResNet model. Notice that while the proof of Lemma 4 (local boundness and Lipschitzness of the gradient) is very different for different network architectures (e.g., ResNet), the other steps that are required to prove these theorems do not depend on the network model. ■

Note that the first line in (15) implies convergence to zero training loss, the second line implies stability of the weights during training (the bound on their amount of change does not depend on the network width n), and the third line shows the stability of the NTK Gram matrix, which implies $\hat{\boldsymbol{\Theta}}_t \xrightarrow{n \to \infty} \boldsymbol{\Theta}$, as discussed above.

4. Comparing the Smoothness of ResNet and MLP NTKs

In this section, we compare the smoothness of the results of ResNet and MLP in the NTK regime (and beyond) using different evaluation methodologies. The smoothness property of a learned function (especially, of an interpolation) is of interest also because prior works have connected it with better generalization (Lu et al., 2019; Giryes, 2020; Xie et al., 2020). We start with comparing uniform (i.e., for any input) upper bounds on the norm of the models' Jacobians *after training*, which is possible due to the NTK regime. This analysis formally shows that decreasing α in the ResNet model limits the non-smoothness of the learned function and reduces its associated bound below the bound obtained for MLP. Therefore, we examine different values of α also in other evaluation methodologies, such as kernel visualization and measuring the smoothness of NTK regression outputs by an approximated $\mathcal{L}^2$-norm of the outputs' second derivatives.

Throughout this section, we focus on the distinction between MLPs and ResNets for ReLU activations, which are extremely popular in practice, and for which $T(\mathbf{K})$ and $\dot{T}(\mathbf{K})$ in the GP kernel and the NTK have closed-form expressions (see Appendix D). The smoothness distinction may not exist in other cases, such as when the activations are the identity function (then, the networks merely learn linear maps), or when the activations are the smooth erf function (see Appendix E).

4.1. Comparing Bounds on Models' Jacobians

In the NTK regime, i.e., when the conditions of Theorem 5 hold, we get from the second line in (15) that for any t we have $\boldsymbol{\theta}_t \in B(\boldsymbol{\theta}_0, \frac{3KR_0}{\lambda_{min}(\boldsymbol{\Theta})})$, which by Lemma 3 implies that for $\sqrt{n} \gg \frac{3KR_0}{\lambda_{min}(\boldsymbol{\Theta})}$ the parameters in the NTK regime are tightly connected to their Gaussian initialization. Formally, recall that $\boldsymbol{\theta}_0 = \mathrm{vec}(\mathbf{w}_0^{(L+1)}, \{\mathbf{W}_0^{(\ell)}\}, \{\mathbf{V}_0^{(\ell)}\}, \mathbf{U}_0)$, where all the elements in $\boldsymbol{\theta}_0$ are i.i.d. standard normal. Let $\boldsymbol{\theta} \in B(\boldsymbol{\theta}_0, C)$. Therefore, the spectral norm of $\mathbf{W}^{(\ell)}$ obeys

$$\begin{aligned} \|\mathbf{W}^{(\ell)}\| &\leq \|\mathbf{W}_0^{(\ell)}\| + \|\mathbf{W}^{(\ell)} - \mathbf{W}_0^{(\ell)}\| \\ &\leq 2\sqrt{n} + t + C \leq 3\sqrt{n}, \end{aligned} \tag{16}$$

where the first inequality uses the triangular inequality, the second inequality uses Lemma 3 and $\|\mathbf{W}^{(\ell)} - \mathbf{W}_0^{(\ell)}\| \leq \|\mathbf{W}^{(\ell)} - \mathbf{W}_0^{(\ell)}\|_F \leq \|\boldsymbol{\theta} - \boldsymbol{\theta}_0\|_2 \leq C$, and the last inequality holds with high probability for $\sqrt{n} \gg C$. Using the same arguments we have $\|\mathbf{V}^{(\ell)}\| \leq 3\sqrt{n}$, $\|\mathbf{U}\| \leq \sqrt{d} + 2\sqrt{n}$ and $\|\mathbf{w}^{(L+1)}\|_2 \leq 2\sqrt{n}$. This mean that, in the NTK regime (of both ResNet and MLP) the spectral norm of the weights can be easily bounded. This is in contrast with the general case where there is no convenient way to control the weights of DNNs *after* training.

Using these properties of the NTK regime for *finite*, yet large n (namely, $\sqrt{n} \gg \frac{3KR_0}{\lambda_{min}(\boldsymbol{\Theta})}$), we show the benefit of using small values for the hyperparameter α in ResNets. The advantage of this setting has been empirically demonstrated for classification by Zhang et al. (2019) (outside the NTK regime). An indicator of the smoothness of a (trained) network $f(\mathbf{x})$, which is also amenable to analysis, can be the maximal norm of the network's "input-output Jacobian" $\sup_{\mathbf{x}} \left\| \frac{\partial}{\partial \mathbf{x}} f(\mathbf{x}) \right\|_2$ (similarly to the way Lipschitz continuity of the gradient is used in the optimization literature). Smaller upper bound on this quantity can be interpreted as higher smoothness.

Considering the ResNet model in (6), we have

$$\begin{aligned}\frac{\partial}{\partial \mathbf{x}} f_{ResNet}(\mathbf{x}) &= \frac{\partial f}{\partial \mathbf{x}^{(L)}} \frac{\partial \mathbf{x}^{(L)}}{\partial \mathbf{x}^{(L-1)}} \cdots \frac{\partial \mathbf{x}^{(1)}}{\partial \mathbf{x}^{(0)}} \frac{\partial \mathbf{x}^{(0)}}{\partial \mathbf{x}} \\ &= \frac{\sigma_w}{\sqrt{n}} \mathbf{w}^{(L+1)\top} \left(\prod_{\ell=1}^{L} \left(\mathbf{I}_n + \alpha \frac{\sigma_v}{\sqrt{n}} \mathbf{V}^{(\ell)} \mathrm{diag}\left\{ \phi'(\mathbf{g}^{(\ell)}) \right\} \frac{\sigma_w}{\sqrt{n}} \mathbf{W}^{(\ell)} \right) \right) \frac{1}{\sqrt{d}} \mathbf{U}.\end{aligned} \tag{17}$$

Let us bound $\sup_{\mathbf{x}} \left\| \frac{\partial}{\partial \mathbf{x}} f_{ResNet}(\mathbf{x}) \right\|_2$

$$\begin{aligned}\sup_{\mathbf{x}} \left\| \frac{\partial}{\partial \mathbf{x}} f_{ResNet}(\mathbf{x}) \right\|_2 &\le \frac{\sigma_w}{\sqrt{n}} \|\mathbf{w}^{(L+1)}\|_2 \frac{1}{\sqrt{d}} \|\mathbf{U}\| \cdot \prod_{\ell=1}^{L} \left(1 + \alpha C_\phi \frac{\sigma_v}{\sqrt{n}} \|\mathbf{V}^{(\ell)}\| \frac{\sigma_w}{\sqrt{n}} \|\mathbf{W}^{(\ell)}\| \right) \\ &\le \frac{\sigma_w}{\sqrt{n}} 2\sqrt{n} \frac{1}{\sqrt{d}} (\sqrt{d} + 2\sqrt{n}) \cdot \prod_{\ell=1}^{L} \left(1 + \alpha C_\phi \frac{\sigma_v}{\sqrt{n}} 3\sqrt{n} \frac{\sigma_w}{\sqrt{n}} 3\sqrt{n} \right) \\ &\le 2\sigma_w (1 + 2\sqrt{\frac{n}{d}}) \left(1 + 9\alpha C_\phi \sigma_v \sigma_w\right)^L := B_{\mathrm{ResNet}}.\end{aligned} \tag{18}$$

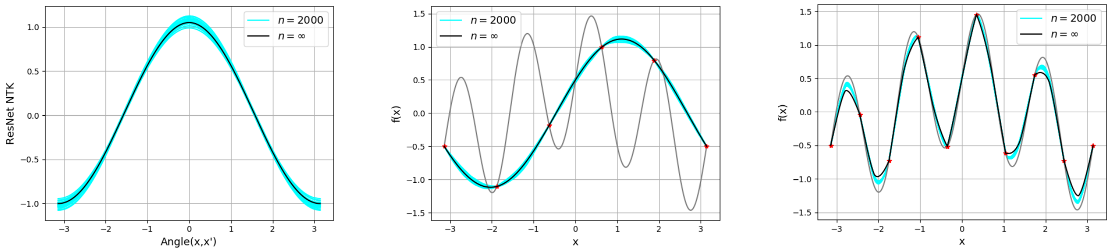
It can be seen that a smaller value of α decreases the bound, which hints that it encourages $f_{\mathrm{ResNet}}(\mathbf{x})$ to be smoother. Note also that $\left\| \frac{\partial}{\partial \mathbf{x}} f_{ResNet}(\mathbf{x}) \right\| = \mathcal{O}(\sqrt{n})$ (which we got due to the fact that, as done in all NTK models, the weights $\mathbf{U}$ that are applied on the input are normalized by the input dimension $\frac{1}{\sqrt{d}}$ rather than by $\frac{1}{\sqrt{n}}$). This factor is not surprising, since we proved that under rather mild conditions on $\mathcal{X}$, the ResNet can fit any training data in the NTK regime, and thus the slope of $f_{\mathrm{ResNet}}(\mathbf{x})$ is not bounded by a constant number. Therefore, to obtain a more formal result on the advantage of small α, let us relate the above result to the one that is obtained for MLP, which also has the same $\sqrt{\frac{n}{d}}$ factor because of the normalization of the first layer.

Considering the MLP model in (1) with $n_1 = \ldots = n_L = n$ and scalar output, we get the following input-output Jacobian

$$\begin{aligned}\frac{\partial}{\partial \mathbf{x}} f_{MLP}(\mathbf{x}) &= \frac{\partial f}{\partial \mathbf{x}^{(L)}} \frac{\partial \mathbf{x}^{(L)}}{\partial \mathbf{x}^{(L-1)}} \cdots \frac{\partial \mathbf{x}^{(1)}}{\partial \mathbf{x}^{(0)}} \frac{\partial \mathbf{x}^{(0)}}{\partial \mathbf{x}} \\ &= \frac{\sigma_w}{\sqrt{n}} \mathbf{w}^{(L+1)\top} \left(\prod_{\ell=2}^{L} \mathrm{diag}\left\{ \phi'(\mathbf{g}^{(\ell)}) \right\} \frac{\sigma_w}{\sqrt{n}} \mathbf{W}^{(\ell)} \right) \mathrm{diag}\left\{ \phi'(\mathbf{g}^{(1)}) \right\} \frac{\sigma_w}{\sqrt{d}} \mathbf{W}^{(1)}.\end{aligned} \tag{19}$$

Let us bound $\sup_{\mathbf{x}} \left\| \frac{\partial}{\partial \mathbf{x}} f_{MLP}(\mathbf{x}) \right\|_2$ (recall that $\mathbf{W}^{(1)} \in \mathbb{R}^{n \times d}$ in the MLP)

$$\begin{aligned}\sup_{\mathbf{x}} \left\| \frac{\partial}{\partial \mathbf{x}} f_{MLP}(\mathbf{x}) \right\|_2 &\le \frac{\sigma_w}{\sqrt{n}} \|\mathbf{w}^{(L+1)}\|_2 C_\phi \frac{\sigma_w}{\sqrt{d}} \|\mathbf{W}^{(1)}\| \prod_{\ell=2}^{L} C_\phi \frac{\sigma_w}{\sqrt{n}} \|\mathbf{W}^{(\ell)}\| \\ &\le \frac{\sigma_w}{\sqrt{n}} 2\sqrt{n} C_\phi \frac{\sigma_w}{\sqrt{d}} (\sqrt{d} + 2\sqrt{n}) \prod_{\ell=2}^{L} C_\phi \frac{\sigma_w}{\sqrt{n}} 3\sqrt{n} \\ &\le 2C_\phi \sigma_w^2 (1 + 2\sqrt{\frac{n}{d}}) \left(3C_\phi \sigma_w\right)^{L-1} := B_{\mathrm{MLP}}.\end{aligned} \tag{20}$$



(a) Empirical and asymptotic NTK (b) Interpolation with 6 samples (c) Interpolation with 10 samples

Figure 1: Empirical (finite width of $n = 2000$ and 30 different Gaussian initializations) and asymptotic NTK for ResNet with $L = 5$ nonlinear layers, ReLU nonlinearities, $\alpha = 0.1$, $\sigma_v = \sigma_w = 1$, for inputs on the sphere (circle) in $\mathbb{R}^2$. (1(a)): The kernel shape. (1(b))-(1(c)): Interpolation using the closed-form NTK solution and using gradient descent training of the finite-width ResNet (5K iterations with lr 0.05 in (1(b)), and 10K iterations with lr 0.5 in (1(c))).

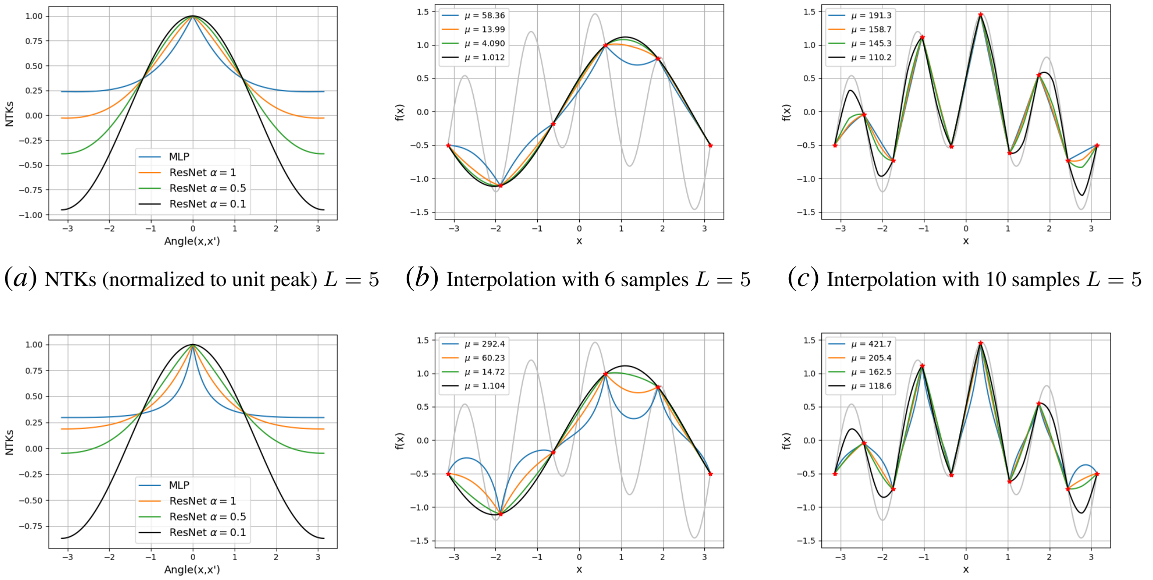
Comparing (18) and (20), we can compute the value of α for which $B_{\text{ResNet}} \leq B_{\text{MLP}}$. We assume that the value of the constants σ_v, σ_w and C_ϕ is 1, as common in practice. Specifically, C_ϕ bounds the expansiveness of the activation function $\phi(\cdot)$ and its derivative (see Lemma 4), and equals 1 for the widely used ReLU activation. The hyperparameters σ_v and σ_w scale the weight matrices, and setting $\sigma_v = \sigma_w = 1$ coincides for square matrices with the popular Xavier's Gaussian initialization (where the standard deviation of the entries is $1/\sqrt{n}$, as appears in our models). Thus, we get

$$\begin{aligned} \frac{B_{\text{ResNet}}}{B_{\text{MLP}}} = \frac{(1+9\alpha C_\phi \sigma_v \sigma_w)^L}{3^{L-1}(C_\phi \sigma_w)^L} = \frac{(1+9\alpha)^L}{3^{L-1}} \\ \leq 1 \iff \alpha \leq \frac{3^{1-1/L}-1}{9} \end{aligned} \tag{21}$$

We can see that a moderate value of α, such as 0.1, implies $B_{\text{ResNet}} \leq B_{\text{MLP}}$ for any $L \geq 3$. Interestingly, this is the fine-tuned value that has been used in the empirical classification paper by Zhang et al. (2019). Note that for $\alpha \to 0$ the residual blocks cannot be trained. Thus, using too small α is not recommended in practice. Even in the NTK regime we have observed no advantage in extremely low α. Finally, (21) hints that if α is small (e.g., smaller than 0.1) then increasing the number of nonlinear layers L will increase the smoothness advantage of the ResNet NTK. This behavior is also demonstrated empirically in the sequel.

4.2. Comparing the Kernels and their Interpolations

In this section, we compare the smoothness of the ResNet NTK and the MLP NTK by visualizing the kernel of each model (which is data-independent if the input norm is fixed), and by comparing the interpolations obtained by kernel regression with the different NTKs. For a given interpolation f with a scalar input $x \in [-\pi, \pi]$ and a scalar output, we scale it to unit $\mathcal{L}^2$-norm, $\bar{f} := f/\|f\|_{\mathcal{L}^2}$, and use an approximate $\mathcal{L}^2$-norm of the second derivative of $\bar{f}$ as a quantitative measure of smoothness (where a smaller value is interpreted as higher smoothness). We empirically find this measure to be richer than first-derivative quantities, like $\sup_x |f'(x)|$ that is more tractable for analysis (as we have



(a) NTKs (normalized to unit peak) $L = 5$ (b) Interpolation with 6 samples $L = 5$ (c) Interpolation with 10 samples $L = 5$

(d) NTKs (normalized to unit peak) $L = 15$ (e) Interpolation with 6 samples $L = 15$ (f) Interpolation with 10 samples $L = 15$

Figure 2: NTKs for MLP and ResNet (for different values of α) with $L = 5$ (top) and $L = 15$ (bottom) nonlinear layers, ReLU nonlinearities, $\sigma_v = \sigma_w = 1$, for inputs on the sphere (circle) in $\mathbb{R}^2$. (2(a)),(2(d)): The kernels shape. (2(b))-(2(c)), (2(e))-(2(f)): Interpolations by the closed-form solutions, measured by $\mu(\cdot)$ defined in (22). Note that the legend in (2(a)),(2(d)) applies to all the figures.

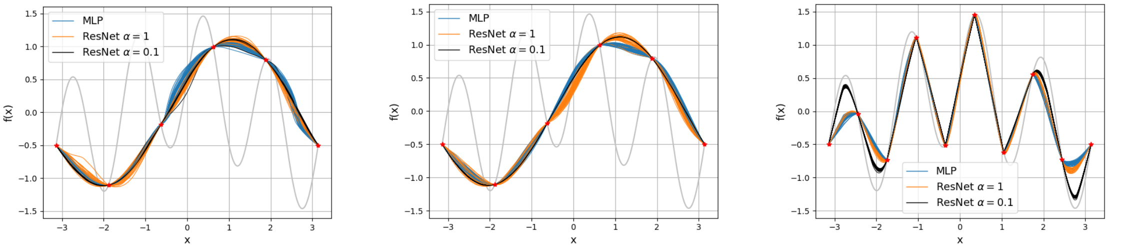
done above) but is affected by linear slopes of f and cannot sum the effects of multiple nonsmooth points.

We numerically approximate $\|\bar{f}''\|_{\mathcal{L}^2}$ as follows. We densely sample f at $N = 4096$ "test samples" with equal spacing $\Delta x = \frac{2\pi}{N}$, i.e., at $x_q = q\Delta x$ with $q = -N/2, ..., N/2-1$. We compute its normalized version $\bar{f} = f/\widehat{\|f\|_{\mathcal{L}^2}}$, where $\widehat{\|f\|_{\mathcal{L}^2}} = (\frac{1}{2\pi}\sum_q |f(x_q)|^2\Delta x)^{0.5}$. As the interpolations in our experiments are periodic ($f(-\pi) = f(\pi)$), we utilize the Fourier series representation of $\bar{f}$ to mitigate numerical computation issues of discrete derivatives. We approximate the kth Fourier coefficient $c_k = \frac{1}{2\pi}\int_{-\pi}^{\pi}\bar{f}(x)\mathrm{e}^{-jkx}dx$ by $\frac{1}{2\pi}\sum_{q=-N/2}^{N/2-1}\bar{f}(x_q)\mathrm{e}^{-jkq2\pi/N}\Delta x = \frac{1}{N}\mathrm{FFT}[\{\bar{f}(x_q)\}](k)$. Thus, denoting $\mathrm{FFT}[\{\bar{f}(x_q)\}](k)$ by $F(k)$, we approximate $\|\bar{f}''\|_{\mathcal{L}^2}$ by

$$\mu(f) := \left(\frac{1}{N^2}\sum_{k=-N/2}^{N/2-1}|k|^4|F(k)|^2\right)^{\frac{1}{2}}, \tag{22}$$

where we used Parseval's identity and the property that the kth Fourier coefficient of the rth derivative of $\bar{f}$, $\bar{f}^{(r)}$, is given by $(jk)^r c_k$ (where c_k is the kth Fourier coefficient of $\bar{f}$).

Note that scaling a kernel by a scalar factor does not change its kernel regression result f and thus also $\mu(f)$. Therefore, we found the measure $\mu(f)$ to be more informative than, e.g., comparing the FFTs of the ResNet and MLP NTKs (we observed that with large/small enough factor the magnitude of the FFT of each of them can be placed on-top/below the other).



(a) Interpolation with 6 samples (Adam) (b) Interpolation with 6 samples (SGD) (c) Interpolation with 10 samples (SGD)

Figure 3: Empirical interpolations of MLP and ResNet (for different values of α) with $L = 5$ nonlinear layers, ReLU nonlinearities, $\sigma_v = \sigma_w = 1$, for inputs on the sphere (circle) in $\mathbb{R}^2$. We use practical models with width of $n = 500$, 30 different Xavier's Gaussian initializations (instead of normalizations by $1/\sqrt{n}$) and 1K iterations of SGD/Adam optimizers.

Before comparing with the MLP, we visualize the theoretical NTK results for ResNet (similar visualization of the NTK theory has been shown only for MLPs, e.g., in (Jacot et al., 2018; Lee et al., 2019)). Figure 1(a) shows the concentration of the empirical NTK (for 30 different Gaussian initializations) around the asymptotic expression given in (8), for the ResNet model with $L = 5$, ReLU nonlinearities, $\alpha = 0.1$, $\sigma_v = 1$, $\sigma_w = 1$, and width of $n = 2000$, for inputs on the sphere (circle) in $\mathbb{R}^2$. Figures 1(b) and 1(c) show that the results of kernel regression (4) with the asymptotic NTK are very similar to the interpolations learned by the ResNet (for 30 different Gaussian initializations, when we use gradient descent with step-size 0.5 and 0.05 for 6 and 10 training samples, respectively).

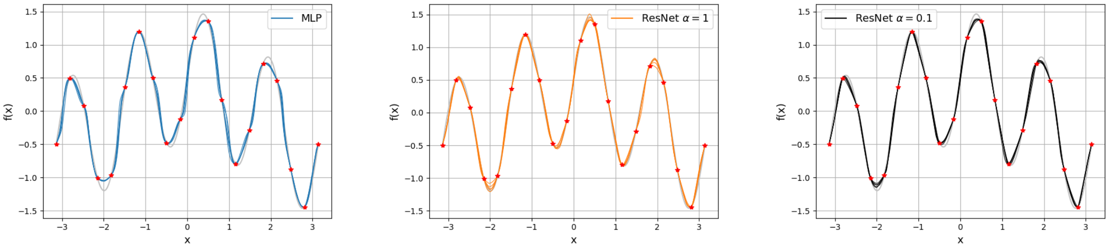
We turn to compare the NTK results for MLP and ResNet, which are given in (5) and (8), respectively. We use $\sigma_v = \sigma_w = 1$, ReLU nonlinearities, and inputs from the circle. We modify the number of nonlinear layers, L, as well as the hyperparameter α in the ResNet. In the interpolation results (where we use (4)) we also modify the amount of given samples and measure the smoothness of each resulted f with $\mu(f)$ defined in (22). The results are presented in Figure 2.

It can be seen that decreasing α yields a smoother ResNet NTK with smoother interpolation results. For $\alpha = 1$ the ResNet NTK is more similar to the MLP NTK, but even then it appears smoother and less "edgy" than the MLP. Moreover, for the MLP NTK, increasing L clearly reduces the smoothness (as observed both visually and by the increase in μ). For the ResNet NTK this effect is moderate for $\alpha = 1$, and almost unseen for $\alpha = 0.1$.

More results and details are presented in Appendix C, including showing that an extremely small value of α does not significantly affect the kernel shape and smoothness compared to the moderate $\alpha = 0.1$. In Appendix C.2 we also present several visual results for a two-dimensional input and accuracy results for binary classification of high-dimensional input (MNIST data).

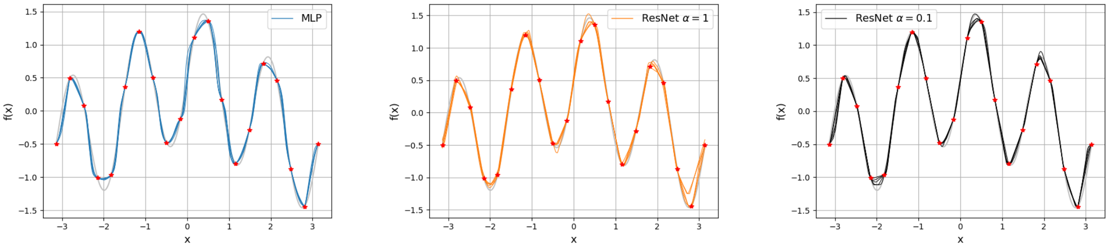
4.3. Results Outside the NTK Regime

Finally, we demonstrate that the observations that are made for the NTK regime carry also to other settings. We consider MLP and ResNet with $L = 5$ nonlinear layers and width of (only) $n = 500$ neurons. We replace the NTK initializations (including the normalizations by $1/\sqrt{n}$) with Xavier's Gaussian initialization (this mainly affects the first and last layers), and instead of gradient descent



(a) Interpolation by MLP (Adam) (b) Interpolation by ResNet $\alpha = 1$ (Adam) (c) Interpolation by ResNet $\alpha = 0.1$ (Adam)

Figure 4: Empirical interpolations of MLP and ResNet (for different values of α) with $L = 5$ nonlinear layers, ReLU nonlinearities, $\sigma_v = \sigma_w = 1$, for inputs on the sphere (circle) in $\mathbb{R}^2$. We use practical models with width of $n = 500$, 5 different Xavier's random Gaussian initializations (instead of normalizations by $1/\sqrt{n}$) and 1K iterations of Adam optimizer.



(a) Interpolation by MLP (Adam) (b) Interpolation by ResNet $\alpha = 1$ (Adam) (c) Interpolation by ResNet $\alpha = 0.1$ (Adam)

Figure 5: Empirical interpolations of MLP and ResNet (for different values of α) with $L = 5$ nonlinear layers, ReLU nonlinearities, $\sigma_v = \sigma_w = 1$, for scalar (1D) inputs. We use practical models with width of $n = 500$ and biases, 5 different PyTorch (default) random uniform initializations (instead of Gaussian initializations and normalizations by $1/\sqrt{n}$) and 1K iterations of Adam optimizer.

we perform either SGD (with lr 0.01) or Adam (with the default parameters stated in (Kingma and Ba, 2014)) on "mini-batches" of size 1. We emphasize that we do not use any explicit regularization (such as batch-normalization or weight decay). The results for 30 different realizations of the initialization are presented in Figure 3.

Despite the discrepancy between these settings and the conditions that are required for the NTK regime to hold, we see similarity in the results. Even in these settings, it is clear that moderately decreasing α yields smoother interpolation results for the ResNet. For $\alpha = 0.1$ the results of ResNet are much smoother than those of MLP. For $\alpha = 1$ the results of ResNet are more similar to the results of MLP, as observed the NTK regime. Yet, with Adam optimizer the results of ResNet are smoother also with $\alpha = 1$. As mentioned above, a detailed empirical study on the use of small values of α (outside the NTK Regime) for classification tasks has been done in (Zhang et al., 2019). Our work can be regarded as an NTK-based support for this approach.

Next, we repeat the numerical experiments with 20 samples. In Figure 4 we present the interpolation results for different Xavier's Gaussian initializations, where the input to the networks is 2D points on the circle, as done in the previous experiments. The optimization method is 1K iterations of Adam with lr 1e-4 and "mini-batches" of size 1, where we save the model with minimal training loss.

In Figure 5 we take a step farther. We add bias to all the layers and feed the networks with plain scalar input. We replace the NTK initializations (including the normalizations by $1/\sqrt{n}$) with PyTorch default uniform initialization (which is similar to Kaiming's initialization). Again, we optimize by the Adam method, as mentioned above (recall that the theory requires gradient descent).

Despite the discrepancy between these settings and the conditions for the NTK regime, we see similarity in the results: The interpolations of the ResNets are smoother than those of the MLP.

5. Conclusion

In this paper we developed the NTK for a ResNet model and proved its stability during training with gradient descent (under common NTK assumptions). As the smoothness of interpolators can indicate better generalization (Lu et al., 2019; Giryes, 2020; Xie et al., 2020), we compared the smoothness properties of ReLU-based ResNet and MLP in the regime where training them can be characterized by kernel regression with their associated NTKs. Our smoothness examination, which is based on different evaluation methodologies, shows that ResNet, especially with moderately attenuated residual blocks, yields smoother interpolations than MLP in the NTK regime. We also showed that this smoothness advantage of ResNet can be observed outside the NTK regime, i.e., when the settings differ from the NTK assumptions.

Our NTK analysis has captured the advantage of reducing the skip weighting factor α. One may inquire whether it is possible to use the NTK regime for finding other new improvements to ResNet.

Acknowledgments

TT and RG acknowledge support from the European research council (ERC StG 757497 PI Giryes) and Nvidia for donating a GPU. JB acknowledges partial support from the Alfred P. Sloan Foundation, NSF RI-1816753, NSF CAREER CIF 1845360, and Samsung Electronics.

References

Sina Alemohammad, Zichao Wang, Randall Balestriero, and Richard Baraniuk. The recurrent neural tangent kernel. *arXiv preprint arXiv:2006.10246*, 2020.

Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pages 322–332. PMLR, 2019a.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. In *Advances in Neural Information Processing Systems*, pages 8141–8150, 2019b.

Ronen Basri, David Jacobs, Yoni Kasten, and Shira Kritchman. The convergence rate of neural networks for learned functions of different frequencies. In *Advances in Neural Information Processing Systems*, pages 4761–4771, 2019.

Alberto Bietti and Francis Bach. Deep equals shallow for relu networks in kernel regimes. *arXiv preprint arXiv:2009.14397*, 2020.

Alberto Bietti and Julien Mairal. On the inductive bias of neural tangent kernels. In *Advances in Neural Information Processing Systems*, volume 32, pages 12873–12884, 2019.

Lin Chen and Sheng Xu. Deep neural tangent kernel and laplace kernel have the same rkhs. *arXiv preprint arXiv:2009.10683*, 2020.

Lenaic Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In *Advances in Neural Information Processing Systems*, pages 2937–2947, 2019.

Youngmin Cho and Lawrence K Saul. Kernel methods for deep learning. In *Advances in neural information processing systems*, pages 342–350, 2009.

Simon Du, Jason Lee, Haochuan Li, Liwei Wang, and Xiyu Zhai. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pages 1675–1685. PMLR, 2019.

Amnon Geifman, Abhay Yadav, Yoni Kasten, Meirav Galun, David Jacobs, and Ronen Basri. On the similarity between the laplace and neural tangent kernels. *arXiv preprint arXiv:2007.01580*, 2020.

Raja Giryes. A function space analysis of finite neural networks with insights from sampling theory. *CoRR, abs/2004.06989*, 2020.

Boris Hanin and Mihai Nica. Finite depth and width corrections to the neural tangent kernel. *arXiv preprint arXiv:1909.05989*, 2019.

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

Jiri Hron, Yasaman Bahri, Jascha Sohl-Dickstein, and Roman Novak. Infinite attention: NNGP and NTK for deep attention networks. In *International Conference on Machine Learning*, pages 4376–4386. PMLR, 2020.

G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger. Densely connected convolutional networks. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2261–2269, 2017.

Kaixuan Huang, Yuqing Wang, Molei Tao, and Tuo Zhao. Why do deep residual networks generalize better than deep feedforward networks?–a neural tangent kernel perspective. *arXiv preprint arXiv:2002.06262*, 2020.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pages 8571–8580, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Alec Koppel, Garrett Warnell, Ethan Stump, and Alejandro Ribeiro. Parsimonious online learning with kernels via sparse projections in function space. *The Journal of Machine Learning Research*, 20(1):83–126, 2019.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105, 2012.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*, 2017.

Jaehoon Lee, Lechao Xiao, Samuel Schoenholz, Yasaman Bahri, Roman Novak, Jascha Sohl-Dickstein, and Jeffrey Pennington. Wide neural networks of any depth evolve as linear models under gradient descent. In *Advances in neural information processing systems*, pages 8572–8583, 2019.

Hao Li, Zheng Xu, Gavin Taylor, Christoph Studer, and Tom Goldstein. Visualizing the loss landscape of neural nets. In *Advances in Neural Information Processing Systems*, pages 6389–6399, 2018.

Etai Littwin, Tomer Galanti, and Lior Wolf. On random kernels of residual architectures. *arXiv preprint arXiv:2001.10460*, 2020.

Lu Lu, Pengzhan Jin, and George Em Karniadakis. Deeponet: Learning nonlinear operators for identifying differential equations based on the universal approximation theorem of operators. *Neural Networks*, 130:85–99, 2019.

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

Bernhard Schölkopf, Alexander J Smola, Francis Bach, et al. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. 2002.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015.

Bharath K Sriperumbudur and Zoltán Szabó. Optimal rates for random fourier features. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 1*, pages 1144–1152, 2015.

Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, volume 97, pages 6105–6114, 2019.

Andreas Veit, Michael Wilber, and Serge Belongie. Residual networks behave like ensembles of relatively shallow networks. In *International Conference on Neural Information Processing Systems*, page 550–558, 2016.

Roman Vershynin. Introduction to the non-asymptotic analysis of random matrices. *arXiv preprint arXiv:1011.3027*, 2010.

Christopher KI Williams. Computing with infinite networks. In *Proceedings of the 9th International Conference on Neural Information Processing Systems*, pages 295–301, 1996.

Francis Williams, Matthew Trager, Claudio Silva, Daniele Panozzo, Denis Zorin, and Joan Bruna. Gradient dynamics of shallow univariate relu networks. *Advances in Neural Information Processing Systems*, 32, 2019.

Yuege Xie, Rachel Ward, Holger Rauhut, and Hung-Hsu Chou. Weighted optimization: better generalization by smoother interpolation. *arXiv preprint arXiv:2006.08495*, 2020.

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

Greg Yang. Wide feedforward or recurrent neural networks of any architecture are gaussian processes. In *Advances in Neural Information Processing Systems*, pages 9951–9960, 2019b.

Greg Yang. Tensor programs ii: Neural tangent kernel for any architecture. *arXiv preprint arXiv:2006.14548*, 2020.

Yibo Yang, Jianlong Wu, Hongyang Li, Xia Li, Tiancheng Shen, and Zhouchen Lin. Dynamical system inspired adaptive time stepping controller for residual network families. In *AAAI*, 2020.

Huishuai Zhang. Training over-parameterized deep resnet is almost as easy as training a two-layer network. *arXiv preprint arXiv:1903.07120*, 2019.

Jingfeng Zhang, Bo Han, Laura Wynter, Bryan Kian Hsiang Low, and Mohan Kankanhalli. Towards robust resnet: a small step but a giant leap. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, pages 4285–4291. AAAI Press, 2019.

Appendix A. Proofs for Theorem 1 and Theorem 2

Note that the structure of the ResNet model in (6) shares similarities with the plain MLP model in (1). For example, due to the central limit theorem when $n \to \infty$ each pre-activation $g_i^{(\ell)}(\mathbf{x})$ is a stochastic Gaussian Process (GP) with zero mean and deterministic GP kernel (covariance), just like in MLP. Indeed, extension of the "convergence at initialization" results of GP kernel and NTK for models beyond MLP has been shown in several works.

Specifically, the convergence of the GP kernel and NTK of the ResNet model that is considered in this paper follows from the general results of (Yang, 2019b,a). This can be done because our ResNet model follows the NETSOR approach (Yang, 2019b,a): It is built from *A-vars* (iid Gaussian weights distributed as $\mathcal{N}(0, \frac{\sigma_a^2}{n})$ for some σ_a), *g-vars* (Gussian vectors with iid entries given by multiplication of A-var with h-var or by sum of other g-vars) and *h-vars* (element-wise nonlinearity, bounded uniformly by $\mathrm{e}^{(cx^2-\epsilon)}$ for some $c, \epsilon > 0$, applied on g-vars). For example, in our model $\mathbf{x}^{(0)}$ is g-var as the input can be considered as h-var, and then recursively $\mathbf{x}^{(\ell)}$ is g-var as it equals g-var + A-var × h-var. Therefore, it remains to compute the limiting kernels.

A.1. Computing the GP kernel for ResNet

To simplify the notation, we add the tilde symbol above each term that depends on the input $\tilde{\mathbf{x}}$ (rather than on $\mathbf{x}$), e.g., $\tilde{\mathbf{x}}^{(\ell)}$ denotes $\mathbf{x}^{(\ell)}(\tilde{\mathbf{x}})$. We will repeatedly use "total expectation", both for eliminating the cross-terms in $\mathbb{E}\left\langle \mathbf{x}^{(\ell)}, \tilde{\mathbf{x}}^{(\ell)} \right\rangle$, i.e., for $\mathbb{E}\left\langle \mathbf{x}^{(\ell-1)}, \mathbf{V}^{(\ell)}\phi(\tilde{\mathbf{g}}^{(\ell)}) \right\rangle = 0$, as well as for exploiting $\mathbb{E}\left[\mathbf{V}^{(\ell)\top}\mathbf{V}^{(\ell)}\right] = \mathbb{E}\left[\mathbf{W}^{(\ell)\top}\mathbf{W}^{(\ell)}\right] = n\mathbf{I}_n$ and $\mathbb{E}\left[\mathbf{w}^{(L+1)}\mathbf{w}^{(L+1)\top}\right] = \mathbf{I}_n$.

First, note the identity

$$\mathbb{E}\left[g_i^{(\ell)}\tilde{g}_i^{(\ell)}\right] = \mathbb{E}\left[\frac{\sigma_w}{\sqrt{n}}\mathbf{x}^{(\ell-1)\top}\mathbf{w}_i^{(\ell)}\frac{\sigma_w}{\sqrt{n}}\mathbf{w}_i^{(\ell)\top}\tilde{\mathbf{x}}^{(\ell-1)}\right] = \frac{\sigma_w^2}{n}\mathbb{E}\left\langle \mathbf{x}^{(\ell-1)}, \tilde{\mathbf{x}}^{(\ell-1)} \right\rangle, \tag{A.1}$$

where $\mathbf{w}_i^{(\ell)\top}$ denotes the ith row of $\mathbf{W}^{(\ell)}$. Therefore, we have

$$K^{(L+1)}(\mathbf{x}, \tilde{\mathbf{x}}) = \mathbb{E}\left[g^{(L+1)}\tilde{g}^{(L+1)}\right] = \frac{\sigma_w^2}{n}\mathbb{E}\left\langle \mathbf{x}^{(L)}, \tilde{\mathbf{x}}^{(L)} \right\rangle. \tag{A.2}$$

Using $\mathbf{x}^{(\ell)} = \mathbf{x}^{(\ell-1)} + \alpha\frac{\sigma_v}{\sqrt{n}}\mathbf{V}^{(\ell)}\phi(\mathbf{g}^{(\ell)})$ and $\mathbb{E}\left\langle \mathbf{x}^{(\ell-1)}, \mathbf{V}^{(\ell)}\phi(\tilde{\mathbf{g}}^{(\ell)}) \right\rangle = 0$, we get

$$\begin{aligned} K^{(L+1)}(\mathbf{x}, \tilde{\mathbf{x}}) &= \frac{\sigma_w^2}{n}\mathbb{E}\left\langle \mathbf{x}^{(L-1)}, \tilde{\mathbf{x}}^{(L-1)} \right\rangle + \alpha^2\frac{\sigma_w^2}{n}\frac{\sigma_v^2}{n}\mathbb{E}\left\langle \mathbf{V}^{(L)}\phi(\mathbf{g}^{(L)}), \mathbf{V}^{(L)}\phi(\tilde{\mathbf{g}}^{(L)}) \right\rangle \\ &= \frac{\sigma_w^2}{n}\mathbb{E}\left\langle \mathbf{x}^{(L-1)}, \tilde{\mathbf{x}}^{(L-1)} \right\rangle + \alpha^2\frac{\sigma_w^2\sigma_v^2}{n}\mathbb{E}\left\langle \phi(\mathbf{g}^{(L)}), \phi(\tilde{\mathbf{g}}^{(L)}) \right\rangle \\ &= \mathbb{E}\left[g_i^{(L)}\tilde{g}_i^{(L)}\right] + \alpha^2\sigma_w^2\sigma_v^2\mathbb{E}\left[\phi(g_i^{(L)})\phi(\tilde{g}_i^{(L)})\right] \\ &= K^{(L)}(\mathbf{x}, \tilde{\mathbf{x}}) + \alpha^2\sigma_v^2\sigma_w^2 T\left(\begin{bmatrix} K^{(L)}(\mathbf{x}, \mathbf{x}) & K^{(L)}(\mathbf{x}, \tilde{\mathbf{x}}) \\ K^{(L)}(\mathbf{x}, \tilde{\mathbf{x}}) & K^{(L)}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}) \end{bmatrix}\right). \end{aligned} \tag{A.3}$$

We also have

$$K^{(1)}(\mathbf{x}, \tilde{\mathbf{x}}) = \frac{\sigma_w^2}{n}\mathbb{E}\left\langle \mathbf{x}^{(0)}, \tilde{\mathbf{x}}^{(0)} \right\rangle = \frac{\sigma_w^2}{n}\frac{1}{d}\mathbb{E}\left[\mathbf{x}^\top\mathbf{U}^\top\mathbf{U}\tilde{\mathbf{x}}\right] = \frac{\sigma_w^2}{d}\mathbf{x}^\top\tilde{\mathbf{x}}. \tag{A.4}$$

A.2. Computing the NTK for ResNet

To simplify the notation, we add the tilde symbol above each term that depends on the input $\tilde{\mathbf{x}}$ (rather than on $\mathbf{x}$), e.g., $\tilde{\mathbf{x}}^{(\ell)}$ denotes $\mathbf{x}^{(\ell)}(\tilde{\mathbf{x}})$. Recall the parameter vector $\boldsymbol{\theta} = \mathrm{vec}(\mathbf{w}^{(L+1)}, \{\mathbf{W}^{(\ell)}\}, \{\mathbf{V}^{(\ell)}\}, \mathbf{U})$. Therefore, we have

$$\Theta^{(L+1)}(\mathbf{x}, \tilde{\mathbf{x}}) = \mathbb{E}\left\langle \frac{\partial f(\mathbf{x};\boldsymbol{\theta})}{\partial \boldsymbol{\theta}}, \frac{\partial f(\tilde{\mathbf{x}};\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} \right\rangle \tag{A.5}$$

$$= \mathbb{E}\left\langle \frac{\partial f}{\partial \mathbf{w}^{(L+1)}}, \frac{\partial \tilde{f}}{\partial \mathbf{w}^{(L+1)}} \right\rangle + \mathbb{E}\left\langle \frac{\partial f}{\partial \mathbf{U}}, \frac{\partial \tilde{f}}{\partial \mathbf{U}} \right\rangle + \sum_{\ell=1}^{L} \mathbb{E}\left\langle \frac{\partial f}{\partial \mathbf{W}^{(\ell)}}, \frac{\partial \tilde{f}}{\partial \mathbf{W}^{(\ell)}} \right\rangle + \mathbb{E}\left\langle \frac{\partial f}{\partial \mathbf{V}^{(\ell)}}, \frac{\partial \tilde{f}}{\partial \mathbf{V}^{(\ell)}} \right\rangle.$$

Clearly, $\frac{\partial f}{\partial \mathbf{w}^{(L+1)}} = \frac{\partial g^{(L+1)}}{\partial \mathbf{w}^{(L+1)}} = \frac{\sigma_w}{\sqrt{n}} \mathbf{x}^{(L)\top}$. To express the other derivatives let us define

$$\boldsymbol{\delta}^{(\ell)} := \nabla_{\mathbf{x}^{(\ell)}} f = \left(\frac{\partial f}{\partial \mathbf{x}^{(\ell)}} \right)^{\top} = \left(\frac{\sigma_w}{\sqrt{n}} \mathbf{w}^{(L+1)\top} \frac{\partial \mathbf{x}^{(L)}}{\partial \mathbf{x}^{(L-1)}} \cdots \frac{\partial \mathbf{x}^{(\ell+1)}}{\partial \mathbf{x}^{(\ell)}} \right)^{\top}, \tag{A.6}$$

and note that from $\mathbf{x}^{(\ell)} = \mathbf{x}^{(\ell-1)} + \alpha \frac{\sigma_v}{\sqrt{n}} \mathbf{V}^{(\ell)} \phi(\mathbf{g}^{(\ell)}) = \mathbf{x}^{(\ell-1)} + \alpha \frac{\sigma_v}{\sqrt{n}} \mathbf{V}^{(\ell)} \phi(\frac{\sigma_w}{\sqrt{n}} \mathbf{W}^{(\ell)} \mathbf{x}^{(\ell-1)})$ we have

$$\frac{\partial \mathbf{x}^{(\ell)}}{\partial \mathbf{x}^{(\ell-1)}} = \mathbf{I}_n + \alpha \frac{\sigma_v}{\sqrt{n}} \mathbf{V}^{(\ell)} \mathrm{diag}\left\{ \phi'(\mathbf{g}^{(\ell)}) \right\} \frac{\sigma_w}{\sqrt{n}} \mathbf{W}^{(\ell)}. \tag{A.7}$$

Other necessary derivatives are given by

$$\frac{\partial x_i^{(\ell)}}{\partial \mathbf{W}^{(\ell)}} = \sum_{j=1}^{n} \frac{\partial x_i^{(\ell)}}{\partial \phi(g_j^{(\ell)})} \frac{\partial \phi(g_j^{(\ell)})}{\partial \mathbf{W}^{(\ell)}} = \alpha \frac{\sigma_v}{\sqrt{n}} \sum_{j=1}^{n} V_{ij}^{(\ell)} \phi'(g_j^{(\ell)}) \frac{\sigma_w}{\sqrt{n}} \begin{bmatrix} \mathbf{0} \\ \mathbf{x}^{(\ell-1)\top} \text{ at row } j \\ \mathbf{0} \end{bmatrix} \tag{A.8}$$

$$= \alpha \frac{\sigma_v}{\sqrt{n}} \frac{\sigma_w}{\sqrt{n}} \begin{bmatrix} V_{i1}^{(\ell)} \phi'(g_1^{(\ell)}) \mathbf{x}^{(\ell-1)\top} \\ \vdots \\ V_{in}^{(\ell)} \phi'(g_n^{(\ell)}) \mathbf{x}^{(\ell-1)\top} \end{bmatrix} = \alpha \frac{\sigma_v}{\sqrt{n}} \frac{\sigma_w}{\sqrt{n}} \mathrm{diag}\left\{ \phi'(\mathbf{g}^{(\ell)}) \right\} \mathbf{v}_i^{(\ell)} \mathbf{x}^{(\ell-1)\top},$$

$$\frac{\partial x_i^{(\ell)}}{\partial \mathbf{V}^{(\ell)}} = \alpha \frac{\sigma_v}{\sqrt{n}} \begin{bmatrix} \mathbf{0} \\ \phi(\mathbf{g}^{(\ell)})^{\top} \text{ at row } i \\ \mathbf{0} \end{bmatrix},$$

where $\mathbf{v}_i^{(\ell)\top}$ denotes the ith row of $\mathbf{V}^{(\ell)}$. This yields

$$\frac{\partial f}{\partial \mathbf{W}^{(\ell)}} = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i^{(\ell)}} \frac{\partial x_i^{(\ell)}}{\partial \mathbf{W}^{(\ell)}} = \alpha \frac{\sigma_v}{\sqrt{n}} \frac{\sigma_w}{\sqrt{n}} \mathrm{diag}\left\{ \phi'(\mathbf{g}^{(\ell)}) \right\} \mathbf{V}^{(\ell)\top} \boldsymbol{\delta}^{(\ell)} \mathbf{x}^{(\ell-1)\top}, \tag{A.9}$$

$$\frac{\partial f}{\partial \mathbf{V}^{(\ell)}} = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i^{(\ell)}} \frac{\partial x_i^{(\ell)}}{\partial \mathbf{V}^{(\ell)}} = \alpha \frac{\sigma_v}{\sqrt{n}} \boldsymbol{\delta}^{(\ell)} \phi(\mathbf{g}^{(\ell)})^{\top},$$

$$\frac{\partial f}{\partial \mathbf{U}} = \sum_{i=1}^{n} \frac{\partial f}{\partial x_i^{(0)}} \frac{\partial x_i^{(0)}}{\partial \mathbf{U}} = \frac{1}{\sqrt{d}} \boldsymbol{\delta}^{(0)} \mathbf{x}^{\top}.$$

Now we can compute the required expectations by repeatedly using "total expectation", conditioning (mainly) on random variables after the ℓth layer, and noting that since we consider $n \to \infty$ the covariance of $g_i^{(\ell)}, \tilde{g}_i^{(\ell)}$ (or $\frac{\sigma_w^2}{n}\mathbf{x}^{(\ell-1)\top}\tilde{\mathbf{x}}^{(\ell-1)}$) converges to the deterministic $K^{(\ell)}(\mathbf{x}, \tilde{\mathbf{x}})$, so values of $T(\cdot)$ and $\dot{T}(\cdot)$ are deterministic and can be taken out of the expectations.

$$\mathbb{E}\left\langle \frac{\partial f}{\partial \mathbf{w}^{(L+1)}}, \frac{\partial \tilde{f}}{\partial \mathbf{w}^{(L+1)}} \right\rangle = \frac{\sigma_w^2}{n}\mathbb{E}\left\langle \mathbf{x}^{(L)}, \tilde{\mathbf{x}}^{(L)} \right\rangle = K^{(L+1)}(\mathbf{x}, \tilde{\mathbf{x}}). \tag{A.10}$$

$$\mathbb{E}\left\langle \frac{\partial f}{\partial \mathbf{U}}, \frac{\partial \tilde{f}}{\partial \mathbf{U}} \right\rangle = \mathbb{E}\left[\boldsymbol{\delta}^{(0)\top}\tilde{\boldsymbol{\delta}}^{(0)}\right] \cdot \frac{1}{d}\mathbf{x}^\top\tilde{\mathbf{x}} = \frac{1}{\sigma_w^2}\mathbb{E}\left[\boldsymbol{\delta}^{(0)\top}\tilde{\boldsymbol{\delta}}^{(0)}\right] \cdot K^{(1)}(\mathbf{x}, \tilde{\mathbf{x}}) \tag{A.11}$$

$$\begin{aligned}\mathbb{E}\left\langle \frac{\partial f}{\partial \mathbf{V}^{(\ell)}}, \frac{\partial \tilde{f}}{\partial \mathbf{V}^{(\ell)}} \right\rangle &= \alpha^2\mathbb{E}\left[\boldsymbol{\delta}^{(\ell)\top}\tilde{\boldsymbol{\delta}}^{(\ell)} \cdot \frac{\sigma_v^2}{n}\phi(\mathbf{g}^{(\ell)})^\top\phi(\tilde{\mathbf{g}}^{(\ell)})\right] \qquad\qquad (\text{A.12})\\ &= \alpha^2\mathbb{E}\left[\boldsymbol{\delta}^{(\ell)\top}\tilde{\boldsymbol{\delta}}^{(\ell)} \cdot \frac{\sigma_v^2}{n}\sum_{i=1}^n \phi(g_i^{(\ell)})\phi(\tilde{g}_i^{(\ell)})\right]\\ &= \alpha^2\mathbb{E}\left[\boldsymbol{\delta}^{(\ell)\top}\tilde{\boldsymbol{\delta}}^{(\ell)}\right] \cdot \sigma_v^2 T\left(\begin{bmatrix} K^{(\ell)}(\mathbf{x}, \mathbf{x}) & K^{(\ell)}(\mathbf{x}, \tilde{\mathbf{x}}) \\ K^{(\ell)}(\mathbf{x}, \tilde{\mathbf{x}}) & K^{(\ell)}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}) \end{bmatrix}\right).\end{aligned}$$

$$\begin{aligned}\mathbb{E}\left\langle \frac{\partial f}{\partial \mathbf{W}^{(\ell)}}, \frac{\partial \tilde{f}}{\partial \mathbf{W}^{(\ell)}} \right\rangle &= \alpha^2\mathbb{E}\left[\frac{\sigma_w^2}{n}\mathbf{x}^{(\ell-1)\top}\tilde{\mathbf{x}}^{(\ell-1)} \cdot \boldsymbol{\delta}^{(\ell)\top}\mathbf{V}^{(\ell)}\frac{\sigma_v^2}{n}\mathrm{diag}\left\{\phi'(\mathbf{g}^{(\ell)})\right\}\mathrm{diag}\left\{\phi'(\tilde{\mathbf{g}}^{(\ell)})\right\}\mathbf{V}^{(\ell)\top}\tilde{\boldsymbol{\delta}}^{(\ell)}\right]\\ &= \alpha^2\mathbb{E}\left[\frac{\sigma_w^2}{n}\mathbf{x}^{(\ell-1)\top}\tilde{\mathbf{x}}^{(\ell-1)} \cdot \boldsymbol{\delta}^{(\ell)\top}\left(\frac{\sigma_v^2}{n}\sum_{i=1}^n \phi'(g_i^{(\ell)})\phi'(\tilde{g}_i^{(\ell)})\mathbf{v}_i^{(\ell)}\mathbf{v}_i^{(\ell)\top}\right)\tilde{\boldsymbol{\delta}}^{(\ell)}\right]\\ &= \alpha^2\mathbb{E}\left[\frac{\sigma_w^2}{n}\mathbf{x}^{(\ell-1)\top}\tilde{\mathbf{x}}^{(\ell-1)} \cdot \boldsymbol{\delta}^{(\ell)\top}\tilde{\boldsymbol{\delta}}^{(\ell)} \cdot \frac{\sigma_v^2}{n}\sum_{i=1}^n \phi'(g_i^{(\ell)})\phi'(\tilde{g}_i^{(\ell)})\right]\\ &= \alpha^2 K^{(\ell)}(\mathbf{x}, \tilde{\mathbf{x}}) \cdot \mathbb{E}\left[\boldsymbol{\delta}^{(\ell)\top}\tilde{\boldsymbol{\delta}}^{(\ell)}\right] \cdot \sigma_v^2\dot{T}\left(\begin{bmatrix} K^{(\ell)}(\mathbf{x}, \mathbf{x}) & K^{(\ell)}(\mathbf{x}, \tilde{\mathbf{x}}) \\ K^{(\ell)}(\mathbf{x}, \tilde{\mathbf{x}}) & K^{(\ell)}(\tilde{\mathbf{x}}, \tilde{\mathbf{x}}) \end{bmatrix}\right).\end{aligned} \tag{A.13}$$

Let us now derive a recursive expression for $\Pi^{(\ell)}(\mathbf{x}, \tilde{\mathbf{x}}) := \frac{1}{\sigma_w^2}\mathbb{E}\left[\boldsymbol{\delta}^{(\ell)\top}\tilde{\boldsymbol{\delta}}^{(\ell)}\right]$, using the relation $\boldsymbol{\delta}^{(\ell)} = \left(\frac{\partial \mathbf{x}^{(\ell+1)}}{\partial \mathbf{x}^{(\ell)}}\right)^\top \boldsymbol{\delta}^{(\ell+1)} = \left(\mathbf{I}_n + \alpha\frac{\sigma_w}{\sqrt{n}}\mathbf{W}^{(\ell+1)\top}\mathrm{diag}\left\{\phi'(\mathbf{g}^{(\ell+1)})\right\}\frac{\sigma_v}{\sqrt{n}}\mathbf{V}^{(\ell+1)\top}\right)\boldsymbol{\delta}^{(\ell+1)}$

$$
\begin{aligned}
&\Pi^{(\ell)}(\mathbf{x},\tilde{\mathbf{x}}) = \frac{1}{\sigma_w^2}\mathbb{E}\left[\boldsymbol{\delta}^{(\ell)\top}\tilde{\boldsymbol{\delta}}^{(\ell)}\right] \qquad\qquad \text{(A.14)}\\
&= \frac{1}{\sigma_w^2}\mathbb{E}\left[\boldsymbol{\delta}^{(\ell+1)\top}\left(\mathbf{I}_n + \alpha^2\frac{\sigma_v^2}{n}\mathbf{V}^{(\ell+1)}\mathrm{diag}\left\{\phi'(\mathbf{g}^{(\ell+1)})\right\}\frac{\sigma_w^2}{n}\mathbf{W}^{(\ell+1)}\mathbf{W}^{(\ell+1)\top}\mathrm{diag}\left\{\phi'(\tilde{\mathbf{g}}^{(\ell+1)})\right\}\mathbf{V}^{(\ell+1)\top}\right)\tilde{\boldsymbol{\delta}}^{(\ell+1)}\right]\\
&= \frac{1}{\sigma_w^2}\mathbb{E}\left[\boldsymbol{\delta}^{(\ell+1)\top}\left(\mathbf{I}_n + \alpha^2\frac{\sigma_v^2\sigma_w^2}{n}\mathbf{V}^{(\ell+1)}\mathrm{diag}\left\{\phi'(\mathbf{g}^{(\ell+1)})\right\}\mathrm{diag}\left\{\phi'(\tilde{\mathbf{g}}^{(\ell+1)})\right\}\mathbf{V}^{(\ell+1)\top}\right)\tilde{\boldsymbol{\delta}}^{(\ell+1)}\right]\\
&= \frac{1}{\sigma_w^2}\mathbb{E}\left[\boldsymbol{\delta}^{(\ell+1)\top}\left(\mathbf{I}_n + \alpha^2\left(\frac{\sigma_v^2\sigma_w^2}{n}\sum_{i=1}^{n}\phi'(g_i^{(\ell+1)})\phi'(\tilde{g}_i^{(\ell+1)})\mathbf{v}_i^{(\ell+1)}\mathbf{v}_i^{(\ell+1)\top}\right)\right)\tilde{\boldsymbol{\delta}}^{(\ell+1)}\right]\\
&= \frac{1}{\sigma_w^2}\mathbb{E}\left[\boldsymbol{\delta}^{(\ell+1)\top}\tilde{\boldsymbol{\delta}}^{(\ell+1)}\cdot\left(1 + \alpha^2\frac{\sigma_v^2\sigma_w^2}{n}\sum_{i=1}^{n}\phi'(g_i^{(\ell+1)})\phi'(\tilde{g}_i^{(\ell+1)})\right)\right]\\
&= \Pi^{(\ell+1)}(\mathbf{x},\tilde{\mathbf{x}})\cdot\left(1 + \alpha^2\sigma_v^2\sigma_w^2\dot{T}\left(\begin{bmatrix}K^{(\ell+1)}(\mathbf{x},\mathbf{x}) & K^{(\ell+1)}(\mathbf{x},\tilde{\mathbf{x}})\\ K^{(\ell+1)}(\mathbf{x},\tilde{\mathbf{x}}) & K^{(\ell+1)}(\tilde{\mathbf{x}},\tilde{\mathbf{x}})\end{bmatrix}\right)\right).
\end{aligned}
$$

Note that the reasoning for the third equality (where total expectation is used to handle $\mathbf{W}^{(\ell+1)}\mathbf{W}^{(\ell+1)\top}$) is delicate, since $\mathbf{W}^{(\ell+1)}$ appears also in $\mathbf{g}^{(\ell+1)}$. This obstacle, which occurs in all NTK works, is handled by assuming that $\mathbf{W}^{(\ell+1)\top}$ used in backprop is independent from $\mathbf{W}^{(\ell+1)}$ in $\mathbf{g}^{(\ell+1)}$ that is used in the forward pass. This assumption has been justified in the limit $n \to \infty$, as long as the last layer weight ($\mathbf{w}^{(L+1)}$) is sampled independently from other parameters and has zero mean (Arora et al., 2019b; Yang, 2019a).

Finally, we compute the base case $\ell = L$

$$
\Pi^{(L)}(\mathbf{x},\tilde{\mathbf{x}}) = \frac{1}{\sigma_w^2}\mathbb{E}\left[\boldsymbol{\delta}^{(L)\top}\tilde{\boldsymbol{\delta}}^{(L)}\right] = \frac{1}{\sigma_w^2}\mathbb{E}\left[\frac{\sigma_w}{\sqrt{n}}\mathbf{w}^{(L+1)\top}\frac{\sigma_w}{\sqrt{n}}\mathbf{w}^{(L+1)}\right] = 1. \qquad \text{(A.15)}
$$

Substituting equations A.10–A.13 in (A.5) and using the definitions of $\Pi^{(\ell)}(\mathbf{x},\tilde{\mathbf{x}})$, $\Sigma^{(\ell+1)}(\mathbf{x},\tilde{\mathbf{x}})$ and $\dot{\Sigma}^{(\ell+1)}(\mathbf{x},\tilde{\mathbf{x}})$, we get the expression for the ResNet NTK that appears in (8).

Appendix B. Proof for Lemma 4

Recall that $\boldsymbol{\theta}_0 := \mathrm{vec}(\mathbf{w}_0^{(L+1)}, \{\mathbf{W}_0^{(\ell)}\}, \{\mathbf{V}_0^{(\ell)}\}, \mathbf{U}_0)$, where all the elements in $\boldsymbol{\theta}_0$ are i.i.d. standard normal. Let $\boldsymbol{\theta} \in B(\theta_0, C)$. Therefore, with high probability

$$\|\mathbf{W}^{(\ell)}\| \leq \|\mathbf{W}_0^{(\ell)}\| + \|\mathbf{W}^{(\ell)} - \mathbf{W}_0^{(\ell)}\| \leq 2\sqrt{n} + t + C \leq 3\sqrt{n}, \tag{B.1}$$

where the first inequality uses the triangular inequality, the second inequality uses Lemma 3 and $\|\mathbf{W}^{(\ell)} - \mathbf{W}_0^{(\ell)}\| \leq \|\mathbf{W}^{(\ell)} - \mathbf{W}_0^{(\ell)}\|_F \leq C$ and the last inequality uses $n \gg C^2$ and holds with high probability. Using the same arguments we have $\|\mathbf{V}^{(\ell)}\| \leq 3\sqrt{n}$, $\|\mathbf{U}\| \leq \sqrt{d} + 2\sqrt{n}$ and $\|\mathbf{w}^{(L+1)}\|_2 \leq 2\sqrt{n}$.

Observe that

$$\|\mathbf{J}(\boldsymbol{\theta})\|_F^2 = \sum_{\mathbf{x}\in\mathcal{X}} \left(\left\| \frac{\partial f(\mathbf{x};\theta)}{\partial \mathbf{w}^{(L+1)}} \right\|_2^2 + \left\| \frac{\partial f(\mathbf{x};\theta)}{\partial \mathbf{U}} \right\|_F^2 + \sum_{\ell=1}^{L} \left\| \frac{\partial f(\mathbf{x};\theta)}{\partial \mathbf{W}^{(\ell)}} \right\|_F^2 + \left\| \frac{\partial f(\mathbf{x};\theta)}{\partial \mathbf{V}^{(\ell)}} \right\|_F^2 \right). \tag{B.2}$$

Let us bound the terms in the sum. We will use the equality $\|\mathbf{a}\mathbf{b}^\top\|_F = \|\mathbf{a}\|_2\|\mathbf{b}\|_2$ and the derivatives that are obtained in Appendix A.2.

$$\left\| \frac{\partial f(\mathbf{x};\theta)}{\partial \mathbf{w}^{(L+1)}} \right\|_2 = \frac{\sigma_w}{\sqrt{n}} \left\| \mathbf{x}^{(L)} \right\|_2. \tag{B.3}$$

$$\left\| \frac{\partial f(\mathbf{x};\theta)}{\partial \mathbf{U}} \right\|_F = \frac{1}{\sqrt{d}} \left\| \boldsymbol{\delta}^{(0)} \mathbf{x}^\top \right\|_F = \frac{1}{\sqrt{d}} \left\| \boldsymbol{\delta}^{(0)} \right\|_2 \|\mathbf{x}\|_2 \leq \frac{B}{\sqrt{d}} \left\| \boldsymbol{\delta}^{(0)} \right\|_2. \tag{B.4}$$

$$\begin{aligned} \left\| \frac{\partial f(\mathbf{x};\theta)}{\partial \mathbf{W}^{(\ell)}} \right\|_F &= \left\| \alpha \frac{\sigma_v}{\sqrt{n}} \frac{\sigma_w}{\sqrt{n}} \mathrm{diag}\left\{ \phi'(\mathbf{g}^{(\ell)}) \right\} \mathbf{V}^{(\ell)\top} \boldsymbol{\delta}^{(\ell)} \mathbf{x}^{(\ell-1)\top} \right\|_F \\ &\leq \alpha C_\phi \sigma_v \sigma_w \frac{1}{\sqrt{n}} \|\mathbf{V}^{(\ell)\top} \boldsymbol{\delta}^{(\ell)}\|_2 \frac{1}{\sqrt{n}} \|\mathbf{x}^{(\ell-1)}\|_2 \\ &\leq \alpha C_\phi \sigma_v \sigma_w \frac{1}{\sqrt{n}} \|\mathbf{V}^{(\ell)}\| \|\boldsymbol{\delta}^{(\ell)}\|_2 \frac{1}{\sqrt{n}} \|\mathbf{x}^{(\ell-1)}\|_2 \\ &\leq 3\alpha C_\phi \sigma_v \sigma_w \|\boldsymbol{\delta}^{(\ell)}\|_2 \frac{1}{\sqrt{n}} \|\mathbf{x}^{(\ell-1)}\|_2. \end{aligned} \tag{B.5}$$

$$\begin{aligned} \left\| \frac{\partial f(\mathbf{x};\theta)}{\partial \mathbf{V}^{(\ell)}} \right\|_F &= \left\| \alpha \frac{\sigma_v}{\sqrt{n}} \boldsymbol{\delta}^{(\ell)} \phi(\mathbf{g}^{(\ell)})^\top \right\|_F = \left\| \alpha \frac{\sigma_v}{\sqrt{n}} \boldsymbol{\delta}^{(\ell)} \phi(\frac{\sigma_w}{\sqrt{n}} \mathbf{W}^{(\ell)} \mathbf{x}^{(\ell-1)})^\top \right\|_F \\ &\leq \alpha C_\phi \sigma_v \sigma_w \|\boldsymbol{\delta}^{(\ell)}\|_2 \frac{1}{\sqrt{n}} \|\mathbf{W}^{(\ell)}\| \frac{1}{\sqrt{n}} \|\mathbf{x}^{(\ell-1)}\|_2 \\ &\leq 3\alpha C_\phi \sigma_v \sigma_w \|\boldsymbol{\delta}^{(\ell)}\|_2 \frac{1}{\sqrt{n}} \|\mathbf{x}^{(\ell-1)}\|_2. \end{aligned} \tag{B.6}$$

In Section B.1 we prove that $\frac{1}{\sqrt{n}}\|\mathbf{x}^{(\ell)}\|_2 \leq K_1$ and $\|\boldsymbol{\delta}^{(\ell)}\|_2 \leq K_2$. Therefore,

$$\|\mathbf{J}(\boldsymbol{\theta})\|_F \leq \sqrt{|\mathcal{X}| \left((c_0 K_1)^2 + (c_1 K_2)^2 + L(c_2 K_1 K_2)^2 + L(c_3 K_1 K_2)^2 \right)} = \tilde{K}. \tag{B.7}$$

We turn to show that $\|\mathbf{J}(\boldsymbol{\theta}) - \mathbf{J}(\tilde{\boldsymbol{\theta}})\|_F \leq K\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2$ for $\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}} \in B(\boldsymbol{\theta}_0, C)$. To simplify the notation, we add the tilde symbol above each term that depends on the $\tilde{\boldsymbol{\theta}}$ (rather than on $\boldsymbol{\theta}$), e.g., $\tilde{\mathbf{x}}^{(\ell)}$ denotes $\mathbf{x}^{(\ell)}(\mathbf{x}; \tilde{\boldsymbol{\theta}})$.

$$\begin{aligned}\|\mathbf{J}(\boldsymbol{\theta}) - \mathbf{J}(\tilde{\boldsymbol{\theta}})\|_F^2 = \sum_{\mathbf{x}\in\mathcal{X}} \Bigg(&\left\| \frac{\partial f(\mathbf{x};\theta)}{\partial \mathbf{w}^{(L+1)}} - \frac{\partial f(\mathbf{x};\tilde{\boldsymbol{\theta}})}{\partial \tilde{\mathbf{w}}^{(L+1)}} \right\|_2^2 \left\| \frac{\partial f(\mathbf{x};\theta)}{\partial \mathbf{U}} - \frac{\partial f(\mathbf{x};\tilde{\boldsymbol{\theta}})}{\partial \tilde{\mathbf{U}}} \right\|_F^2 \\ &+ \sum_{\ell=1}^{L} \left\| \frac{\partial f(\mathbf{x};\theta)}{\partial \mathbf{W}^{(\ell)}} - \frac{\partial f(\mathbf{x};\tilde{\theta})}{\partial \tilde{\mathbf{W}}^{(\ell)}} \right\|_F^2 + \left\| \frac{\partial f(\mathbf{x};\theta)}{\partial \mathbf{V}^{(\ell)}} - \frac{\partial f(\mathbf{x};\tilde{\theta})}{\partial \tilde{\mathbf{V}}^{(\ell)}} \right\|_F^2 \Bigg).\end{aligned} \tag{B.8}$$

Let us bound the terms in the sum.

$$\left\| \frac{\partial f(\mathbf{x};\boldsymbol{\theta})}{\partial \mathbf{w}^{(L+1)}} - \frac{\partial f(\mathbf{x};\tilde{\boldsymbol{\theta}})}{\partial \tilde{\mathbf{w}}^{(L+1)}} \right\|_2 = \frac{\sigma_w}{\sqrt{n}} \left\| \mathbf{x}^{(L)} - \tilde{\mathbf{x}}^{(L)} \right\|_2. \tag{B.9}$$

$$\left\| \frac{\partial f(\mathbf{x};\theta)}{\partial \mathbf{U}} - \frac{\partial f(\mathbf{x};\tilde{\boldsymbol{\theta}})}{\partial \tilde{\mathbf{U}}} \right\|_F = \frac{1}{\sqrt{d}} \left\| \boldsymbol{\delta}^{(0)}\mathbf{x}^\top - \tilde{\boldsymbol{\delta}}^{(0)}\mathbf{x}^\top \right\|_F \leq \frac{B}{\sqrt{d}} \left\| \boldsymbol{\delta}^{(0)} - \tilde{\boldsymbol{\delta}}^{(0)} \right\|_2. \tag{B.10}$$

$$\begin{aligned}&\left\| \frac{\partial f(\mathbf{x};\theta)}{\partial \mathbf{W}^{(\ell)}} - \frac{\partial f(\mathbf{x};\tilde{\theta})}{\partial \tilde{\mathbf{W}}^{(\ell)}} \right\|_F \qquad\qquad (\text{B.11}) \\ &= \left\| \underbrace{\alpha \frac{\sigma_v \sigma_w}{\sqrt{n}} \mathrm{diag}\left\{ \phi'(\mathbf{g}^{(\ell)}) \right\} \mathbf{V}^{(\ell)\top} \boldsymbol{\delta}^{(\ell)}}_{:=\boldsymbol{\gamma}^{(\ell)}} \frac{1}{\sqrt{n}} \mathbf{x}^{(\ell-1)\top} - \underbrace{\alpha \frac{\sigma_v \sigma_w}{\sqrt{n}} \mathrm{diag}\left\{ \phi'(\tilde{\mathbf{g}}^{(\ell)}) \right\} \tilde{\mathbf{V}}^{(\ell)\top} \tilde{\boldsymbol{\delta}}^{(\ell)}}_{:=\tilde{\boldsymbol{\gamma}}^{(\ell)}} \frac{1}{\sqrt{n}} \tilde{\mathbf{x}}^{(\ell-1)\top} \right\|_F \\ &\leq \left\| (\boldsymbol{\gamma}^{(\ell)} - \tilde{\boldsymbol{\gamma}}^{(\ell)}) \frac{1}{\sqrt{n}} \mathbf{x}^{(\ell-1)\top} \right\|_F + \left\| \tilde{\boldsymbol{\gamma}}^{(\ell)} \frac{1}{\sqrt{n}} (\mathbf{x}^{(\ell-1)\top} - \tilde{\mathbf{x}}^{(\ell-1)\top}) \right\|_F \\ &\leq \frac{1}{\sqrt{n}} \left\| \mathbf{x}^{(\ell-1)} \right\|_2 \left\| \boldsymbol{\gamma}^{(\ell)} - \tilde{\boldsymbol{\gamma}}^{(\ell)} \right\|_2 + \left\| \tilde{\boldsymbol{\gamma}}^{(\ell)} \right\|_2 \frac{1}{\sqrt{n}} \left\| \mathbf{x}^{(\ell-1)} - \tilde{\mathbf{x}}^{(\ell-1)} \right\|_2 \\ &\leq K_1 \left\| \boldsymbol{\gamma}^{(\ell)} - \tilde{\boldsymbol{\gamma}}^{(\ell)} \right\|_2 + 3\alpha C_\phi \sigma_v \sigma_w K_2 \frac{1}{\sqrt{n}} \left\| \mathbf{x}^{(\ell-1)} - \tilde{\mathbf{x}}^{(\ell-1)} \right\|_2.\end{aligned}$$

$$
\begin{aligned}
&\left\| \frac{\partial f(\mathbf{x};\theta)}{\partial \mathbf{V}^{(\ell)}} - \frac{\partial f(\mathbf{x};\tilde{\theta})}{\partial \tilde{\mathbf{V}}^{(\ell)}} \right\|_F \qquad\qquad \text{(B.12)}\\
&= \left\| \boldsymbol{\delta}^{(\ell)} \frac{1}{\sqrt{n}} \underbrace{\alpha\sigma_v \phi(\frac{\sigma_w}{\sqrt{n}}\mathbf{W}^{(\ell)}\mathbf{x}^{(\ell-1)})^\top}_{:=\mathbf{z}^{(\ell)\top}} - \tilde{\boldsymbol{\delta}}^{(\ell)} \frac{1}{\sqrt{n}} \underbrace{\alpha\sigma_v \phi(\frac{\sigma_w}{\sqrt{n}}\tilde{\mathbf{W}}^{(\ell)}\tilde{\mathbf{x}}^{(\ell-1)})^\top}_{:=\tilde{\mathbf{z}}^{(\ell)\top}} \right\|_F \\
&\le \left\| (\boldsymbol{\delta}^{(\ell)} - \tilde{\boldsymbol{\delta}}^{(\ell)}) \frac{1}{\sqrt{n}} \mathbf{z}^{(\ell)\top} \right\|_F + \left\| \tilde{\boldsymbol{\delta}}^{(\ell)} \frac{1}{\sqrt{n}} (\mathbf{z}^{(\ell)\top} - \tilde{\mathbf{z}}^{(\ell)\top}) \right\|_F \\
&\le \frac{1}{\sqrt{n}} \left\| \mathbf{z}^{(\ell)} \right\|_2 \left\| \boldsymbol{\delta}^{(\ell)} - \tilde{\boldsymbol{\delta}}^{(\ell)} \right\|_2 + \left\| \tilde{\boldsymbol{\delta}}^{(\ell)} \right\|_2 \frac{1}{\sqrt{n}} \left\| \mathbf{z}^{(\ell)} - \tilde{\mathbf{z}}^{(\ell)} \right\|_2 \\
&\le 3\alpha C_\phi \sigma_v \sigma_w K_1 \left\| \boldsymbol{\delta}^{(\ell)} - \tilde{\boldsymbol{\delta}}^{(\ell)} \right\|_2 + K_2 \frac{1}{\sqrt{n}} \left\| \mathbf{z}^{(\ell)} - \tilde{\mathbf{z}}^{(\ell)} \right\|_2 .
\end{aligned}
$$

Showing that $\left\| \boldsymbol{\gamma}^{(\ell)} - \tilde{\boldsymbol{\gamma}}^{(\ell)} \right\|_2, \frac{1}{\sqrt{n}} \|\mathbf{x}^{(\ell)} - \tilde{\mathbf{x}}^{(\ell)}\|_2, \left\| \boldsymbol{\delta}^{(\ell)} - \tilde{\boldsymbol{\delta}}^{(\ell)} \right\|_2, \frac{1}{\sqrt{n}} \|\mathbf{z}^{(\ell)} - \tilde{\mathbf{z}}^{(\ell)}\|_2 \le \overline{K}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2$ allows to obtain the required local Lipschitzness result for $\mathbf{J}(\boldsymbol{\theta})$. However, as shown in Section B.2, proving $\left\| \boldsymbol{\delta}^{(\ell)} - \tilde{\boldsymbol{\delta}}^{(\ell)} \right\|_2 \le \overline{K}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2$ requires that $\left\|\mathbf{x}^{(\ell)} - \tilde{\mathbf{x}}^{(\ell)}\right\|_2 \le \overline{K}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2$ (without the $\frac{1}{\sqrt{n}}$ factor).

In Section B.2 we show that all the distances $\left\| \boldsymbol{\gamma}^{(\ell)} - \tilde{\boldsymbol{\gamma}}^{(\ell)} \right\|_2$, $\|\mathbf{x}^{(\ell)} - \tilde{\mathbf{x}}^{(\ell)}\|_2$, $\left\| \boldsymbol{\delta}^{(\ell)} - \tilde{\boldsymbol{\delta}}^{(\ell)} \right\|_2$, $\left\|\mathbf{z}^{(\ell)} - \tilde{\mathbf{z}}^{(\ell)}\right\|_2$ are indeed upper bounded by $\overline{K}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2$. Therefore,

$$
\|\mathbf{J}(\boldsymbol{\theta}) - \mathbf{J}(\tilde{\boldsymbol{\theta}})\|_F \le \sqrt{|\mathcal{X}| \left((\overline{c_0}\overline{K})^2 + (\overline{c_1}\overline{K})^2 + L(\overline{c_2}\overline{K})^2 + L(\overline{c_3}\overline{K})^2 \right)} \|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2 = \tilde{\tilde{K}} \|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2, \tag{B.13}
$$

and the proof of Lemma 4 is finished with $K = \max(\tilde{K}, \tilde{\tilde{K}})$.

B.1. Auxiliary local boundness proofs

We prove by induction that $\frac{1}{\sqrt{n}}\|\mathbf{x}^{(\ell)}\|_2 \le K_1$.

Base case: since $\|\mathbf{x}\|_2 \le B$, we have with high probability over the random initialization of $\mathbf{U} \in \mathbb{R}^{n \times d}$ that $\frac{1}{\sqrt{n}}\|\mathbf{x}^{(0)}\|_2 = \frac{1}{\sqrt{n}}\|\frac{1}{\sqrt{d}}\mathbf{U}\mathbf{x}\|_2 \le \frac{1}{\sqrt{d}}\frac{1}{\sqrt{n}}\|\mathbf{U}\|B \le \frac{3}{\sqrt{d}}B$.

Assuming that $\frac{1}{\sqrt{n}}\|\mathbf{x}^{(\ell-1)}\|_2 \le \tilde{K}_1$, we get

$$
\begin{aligned}
\frac{1}{\sqrt{n}}\|\mathbf{x}^{(\ell)}\|_2 &= \frac{1}{\sqrt{n}}\|\mathbf{x}^{(\ell-1)} + \alpha\frac{\sigma_v}{\sqrt{n}}\mathbf{V}^{(\ell)}\phi(\frac{\sigma_w}{\sqrt{n}}\mathbf{W}^{(\ell)}\mathbf{x}^{(\ell-1)})\|_2 \qquad \text{(B.14)}\\
&\le \left(1 + \alpha C_\phi \sigma_v \sigma_w \frac{1}{\sqrt{n}}\|\mathbf{V}^{(\ell)}\|\frac{1}{\sqrt{n}}\|\mathbf{W}^{(\ell)}\|\right)\frac{1}{\sqrt{n}}\|\mathbf{x}^{(\ell-1)}\|_2 \\
&\le (1 + 9\alpha C_\phi \sigma_v \sigma_w)\,\tilde{K}_1.
\end{aligned}
$$

Therefore, we have that for all $\ell \in [L] : \frac{1}{\sqrt{n}}\|\mathbf{x}^{(\ell)}\|_2 \le K_1 = (1 + 9\alpha C_\phi \sigma_v \sigma_w)^L \frac{3}{\sqrt{d}}B$.

We prove by induction that $\|\boldsymbol{\delta}^{(\ell)}\|_2 \le K_2$. Recall that $\boldsymbol{\delta}^{(\ell)} = \left(\frac{\sigma_w}{\sqrt{n}}\mathbf{w}^{(L+1)\top}\frac{\partial \mathbf{x}^{(L)}}{\partial \mathbf{x}^{(L-1)}}\cdots\frac{\partial \mathbf{x}^{(\ell+1)}}{\partial \mathbf{x}^{(\ell)}}\right)^\top = \left(\frac{\partial \mathbf{x}^{(\ell+1)}}{\partial \mathbf{x}^{(\ell)}}\right)^\top \boldsymbol{\delta}^{(\ell+1)}$.

Base case: $\|\boldsymbol{\delta}^{(L)}\|_2 = \frac{\sigma_w}{\sqrt{n}}\|\mathbf{w}^{(L+1)}\|_2 \le \frac{\sigma_w}{\sqrt{n}}2\sqrt{n} = 2\sigma_w$.

Assuming that $\|\boldsymbol{\delta}^{(\ell+1)}\|_2 \le \tilde{K}_2$, we get

$$\begin{aligned}\|\boldsymbol{\delta}^{(\ell)}\|_2 &= \left\|\left(\mathbf{I}_n + \alpha\frac{\sigma_w}{\sqrt{n}}\mathbf{W}^{(\ell+1)\top}\mathrm{diag}\left\{\phi'(\mathbf{g}^{(\ell+1)})\right\}\frac{\sigma_v}{\sqrt{n}}\mathbf{V}^{(\ell+1)\top}\right)\boldsymbol{\delta}^{(\ell+1)}\right\|_2 \\ &\le \left(1 + \alpha C_\phi \sigma_v \sigma_w \frac{1}{\sqrt{n}}\|\mathbf{V}^{(\ell+1)}\|\frac{1}{\sqrt{n}}\|\mathbf{W}^{(\ell+1)}\|\right)\|\boldsymbol{\delta}^{(\ell+1)}\|_2 \\ &\le \left(1 + 9\alpha C_\phi \sigma_v \sigma_w\right)\tilde{K}_2.\end{aligned} \tag{B.15}$$

Therefore, we have that for all $\ell \in [L] : \|\boldsymbol{\delta}^{(\ell)}\|_2 \le K_2 = (1 + 9\alpha C_\phi \sigma_v\sigma_w)^L 2\sigma_w$.

B.2. Auxiliary local Lipschitzness proofs

Recall that $\boldsymbol{\theta} = \mathrm{vec}(\mathbf{w}^{(L+1)}, \{\mathbf{W}^{(\ell)}\}, \{\mathbf{V}^{(\ell)}\}, \mathbf{U})$. Therefore, we will repeatedly use $\|\mathbf{W}^{(\ell)} - \tilde{\mathbf{W}}^{(\ell)}\| \le \|\mathbf{W}^{(\ell)} - \tilde{\mathbf{W}}^{(\ell)}\|_F \le \|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2$, and similarly for the other parameters. Also, for simplification we will use $\{c_i\}$ to denote constants that do not depend on $\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}, n$.

We prove by induction (together) that $\left\|\mathbf{x}^{(\ell)} - \tilde{\mathbf{x}}^{(\ell)}\right\|_2 \le \overline{K}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2$ and also $\left\|\mathbf{g}^{(\ell)} - \tilde{\mathbf{g}}^{(\ell)}\right\|_2 \le \overline{K}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2$.

Base case:

$$\left\|\mathbf{x}^{(0)}(\mathbf{x},\theta) - \mathbf{x}^{(0)}(\mathbf{x},\tilde{\theta})\right\|_2 = \left\|\frac{1}{\sqrt{d}}\mathbf{U}\mathbf{x} - \frac{1}{\sqrt{d}}\tilde{\mathbf{U}}\mathbf{x}\right\|_2 \le \frac{1}{\sqrt{d}}\|\mathbf{U} - \tilde{\mathbf{U}}\|\|\mathbf{x}\|_2 \le \frac{B}{\sqrt{d}}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2,$$

and

$$\begin{aligned}\left\|\mathbf{g}^{(1)}(\mathbf{x},\theta) - \mathbf{g}^{(1)}(\mathbf{x},\tilde{\theta})\right\|_2 &= \left\|\frac{\sigma_w}{\sqrt{n}}\mathbf{W}^{(0)}\mathbf{x}^{(0)} - \frac{\sigma_w}{\sqrt{n}}\tilde{\mathbf{W}}^{(1)}\tilde{\mathbf{x}}^{(0)}\right\|_2 \\ &\le \left\|(\mathbf{W}^{(0)} - \tilde{\mathbf{W}}^{(1)})\frac{\sigma_w}{\sqrt{n}}\mathbf{x}^{(0)}\right\|_2 + \left\|\frac{\sigma_w}{\sqrt{n}}\tilde{\mathbf{W}}^{(1)}(\mathbf{x}^{(0)} - \tilde{\mathbf{x}}^{(0)})\right\|_2 \\ &\le \|\mathbf{W}^{(0)} - \tilde{\mathbf{W}}^{(1)}\|\frac{\sigma_w}{\sqrt{n}}\|\mathbf{x}^{(0)}\|_2 + \frac{\sigma_w}{\sqrt{n}}\|\tilde{\mathbf{W}}^{(1)}\|\|\mathbf{x}^{(0)} - \tilde{\mathbf{x}}^{(0)}\|_2 \\ &\le \sigma_w K_1\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2 + 3\sigma_w\frac{B}{\sqrt{d}}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2.\end{aligned} \tag{B.16}$$

Thus, $\left\|\mathbf{x}^{(0)}(\mathbf{x},\theta) - \mathbf{x}^{(0)}(\mathbf{x},\tilde{\theta})\right\|_2, \left\|\mathbf{g}^{(1)}(\mathbf{x},\theta) - \mathbf{g}^{(1)}(\mathbf{x},\tilde{\theta})\right\|_2 \le c_1\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2$.

Assuming that $\|\mathbf{x}^{(\ell-1)} - \tilde{\mathbf{x}}^{(\ell-1)}\|_2, \|\mathbf{g}^{(\ell)} - \tilde{\mathbf{g}}^{(\ell)}\|_2 \leq \tilde{K}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2$, we get

$$\begin{aligned}
\|\mathbf{x}^{(\ell)} - \tilde{\mathbf{x}}^{(\ell)}\|_2 &= \left\|\mathbf{x}^{(\ell-1)} + \alpha\frac{\sigma_v}{\sqrt{n}}\mathbf{V}^{(\ell)}\phi(\mathbf{g}^{(\ell)}) - \tilde{\mathbf{x}}^{(\ell-1)} - \alpha\frac{\sigma_v}{\sqrt{n}}\tilde{\mathbf{V}}^{(\ell)}\phi(\tilde{\mathbf{g}}^{(\ell)})\right\|_2 \qquad \text{(B.17)}\\
&\leq \|\mathbf{x}^{(\ell-1)} - \tilde{\mathbf{x}}^{(\ell-1)}\|_2 + \alpha\left\|\frac{\sigma_v}{\sqrt{n}}\mathbf{V}^{(\ell)}\phi(\mathbf{g}^{(\ell)}) - \frac{\sigma_v}{\sqrt{n}}\tilde{\mathbf{V}}^{(\ell)}\phi(\tilde{\mathbf{g}}^{(\ell)})\right\|_2\\
&\leq \tilde{K}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2 + \alpha\|\mathbf{V}^{(\ell)} - \tilde{\mathbf{V}}^{(\ell)}\|\frac{\sigma_v}{\sqrt{n}}\|\phi(\mathbf{g}^{(\ell)})\|_2 + \frac{\sigma_v}{\sqrt{n}}\|\tilde{\mathbf{V}}^{(\ell)}\|\|\phi(\mathbf{g}^{(\ell)}) - \phi(\tilde{\mathbf{g}}^{(\ell)})\|_2\\
&\leq \tilde{K}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2 + \alpha C_\phi\frac{\sigma_v}{\sqrt{n}}\|\frac{\sigma_w}{\sqrt{n}}\mathbf{W}^{(\ell)}\mathbf{x}^{(\ell-1)}\|_2\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2 + 3\sigma_v C_\phi\|\mathbf{g}^{(\ell)} - \tilde{\mathbf{g}}^{(\ell)}\|_2\\
&\leq (\tilde{K} + 3\sigma_v C_\phi)\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2 + 3\alpha C_\phi\sigma_v\sigma_w\frac{1}{\sqrt{n}}\|\mathbf{x}^{(\ell-1)}\|_2\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2\\
&\leq (\tilde{K} + 3\sigma_v C_\phi + 3\alpha C_\phi\sigma_v\sigma_w K_1)\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2 \leq c_2\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2.
\end{aligned}$$

$$\begin{aligned}
\left\|\mathbf{g}^{(\ell+1)} - \tilde{\mathbf{g}}^{(\ell+1)}\right\|_2 &= \left\|\frac{\sigma_w}{\sqrt{n}}\mathbf{W}^{(\ell+1)}\mathbf{x}^{(\ell)} - \frac{\sigma_w}{\sqrt{n}}\tilde{\mathbf{W}}^{(\ell+1)}\tilde{\mathbf{x}}^{(\ell)}\right\|_2 \qquad \text{(B.18)}\\
&\leq \|\mathbf{W}^{(\ell+1)} - \tilde{\mathbf{W}}^{(\ell+1)}\|\frac{\sigma_w}{\sqrt{n}}\|\mathbf{x}^{(\ell)}\|_2 + \frac{\sigma_w}{\sqrt{n}}\|\tilde{\mathbf{W}}^{(\ell+1)}\|\|\mathbf{x}^{(\ell)} - \tilde{\mathbf{x}}^{(\ell)}\|_2\\
&\leq \sigma_w K_1\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2 + 3\sigma_w c_2\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2 \leq c_3\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2.
\end{aligned}$$

Therefore, we have that for all $\ell \in [L] : \|\mathbf{x}^{(\ell)} - \tilde{\mathbf{x}}^{(\ell)}\|_2, \|\mathbf{g}^{(\ell)} - \tilde{\mathbf{g}}^{(\ell)}\|_2 \leq c_3^L\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2 \leq \overline{K}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2$.

The proof for $\left\|\mathbf{z}^{(\ell)} - \tilde{\mathbf{z}}^{(\ell)}\right\|_2 \leq \overline{K}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2$ follows from directly from $\|\mathbf{g}^{(\ell)} - \tilde{\mathbf{g}}^{(\ell)}\|_2 \leq c_3^L\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2$:

$$\begin{aligned}
\left\|\mathbf{z}^{(\ell)} - \tilde{\mathbf{z}}^{(\ell)}\right\|_2 &= \left\|\alpha\sigma_v\phi(\mathbf{g}^{(\ell)}) - \alpha\sigma_v\phi(\tilde{\mathbf{g}}^{(\ell)})\right\|_2 \qquad \text{(B.19)}\\
&\leq \alpha C_\phi\sigma_v\left\|\mathbf{g}^{(\ell)} - \tilde{\mathbf{g}}^{(\ell)}\right\|_2 \leq \alpha C_\phi\sigma_v c_3^L\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2 \leq \overline{K}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2.
\end{aligned}$$

We turn to prove by induction that $\left\|\boldsymbol{\delta}^{(\ell)} - \tilde{\boldsymbol{\delta}}^{(\ell)}\right\|_2 \leq \overline{K}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2$.

Base case: $\left\|\boldsymbol{\delta}^{(L)} - \tilde{\boldsymbol{\delta}}^{(L)}\right\|_2 = \left\|\frac{\sigma_w}{\sqrt{n}}\mathbf{w}^{(L+1)} - \frac{\sigma_w}{\sqrt{n}}\tilde{\mathbf{w}}^{(L+1)}\right\|_2 \leq \frac{\sigma_w}{\sqrt{n}}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2 \leq \tilde{K}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2$.

Assuming that $\|\boldsymbol{\delta}^{(\ell+1)} - \tilde{\boldsymbol{\delta}}^{(\ell+1)}\|_2 \leq \tilde{K}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2$, we get

$$\begin{aligned}
\|\boldsymbol{\delta}^{(\ell)} - \tilde{\boldsymbol{\delta}}^{(\ell)}\|_2 = \Bigg\| & \left(\mathbf{I}_n + \alpha \frac{\sigma_v}{\sqrt{n}} \mathbf{V}^{(\ell+1)} \mathrm{diag}\left\{\phi'(\mathbf{g}^{(\ell+1)})\right\} \frac{\sigma_w}{\sqrt{n}} \mathbf{W}^{(\ell+1)}\right)^\top \boldsymbol{\delta}^{(\ell+1)} \\
& - \left(\mathbf{I}_n + \alpha \frac{\sigma_v}{\sqrt{n}} \tilde{\mathbf{V}}^{(\ell+1)} \mathrm{diag}\left\{\phi'(\tilde{\mathbf{g}}^{(\ell+1)})\right\} \frac{\sigma_w}{\sqrt{n}} \tilde{\mathbf{W}}^{(\ell+1)}\right)^\top \tilde{\boldsymbol{\delta}}^{(\ell+1)} \Bigg\|_2 \\
\leq \|\boldsymbol{\delta}^{(\ell+1)} - \tilde{\boldsymbol{\delta}}^{(\ell+1)}\|_2 & + \left\| \left(\alpha \frac{\sigma_v}{\sqrt{n}} \mathbf{V}^{(\ell+1)} \mathrm{diag}\left\{\phi'(\mathbf{g}^{(\ell+1)})\right\} \frac{\sigma_w}{\sqrt{n}} \mathbf{W}^{(\ell+1)}\right)^\top (\boldsymbol{\delta}^{(\ell+1)} - \tilde{\boldsymbol{\delta}}^{(\ell+1)}) \right\|_2 \\
& + \left\| \left(\alpha \frac{\sigma_v}{\sqrt{n}} \mathbf{V}^{(\ell+1)} \mathrm{diag}\left\{\phi'(\mathbf{g}^{(\ell+1)})\right\} \frac{\sigma_w}{\sqrt{n}} \mathbf{W}^{(\ell+1)} - \alpha \frac{\sigma_v}{\sqrt{n}} \tilde{\mathbf{V}}^{(\ell+1)} \mathrm{diag}\left\{\phi'(\tilde{\mathbf{g}}^{(\ell+1)})\right\} \frac{\sigma_w}{\sqrt{n}} \tilde{\mathbf{W}}^{(\ell+1)}\right)^\top \tilde{\boldsymbol{\delta}}^{(\ell+1)}) \right\|_2 \\
\leq (\tilde{K} + 9\alpha\sigma_v\sigma_w C_\phi \tilde{K})\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2 & \\
& + K_2 \left\| \alpha \frac{\sigma_v}{\sqrt{n}} \mathbf{V}^{(\ell+1)} \mathrm{diag}\left\{\phi'(\mathbf{g}^{(\ell+1)})\right\} \frac{\sigma_w}{\sqrt{n}} \mathbf{W}^{(\ell+1)} - \alpha \frac{\sigma_v}{\sqrt{n}} \tilde{\mathbf{V}}^{(\ell+1)} \mathrm{diag}\left\{\phi'(\tilde{\mathbf{g}}^{(\ell+1)})\right\} \frac{\sigma_w}{\sqrt{n}} \tilde{\mathbf{W}}^{(\ell+1)} \right\| \\
\leq c_4\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2 & + \alpha K_2 \left\| \frac{\sigma_v}{\sqrt{n}} \mathbf{V}^{(\ell+1)} \mathrm{diag}\left\{\phi'(\mathbf{g}^{(\ell+1)})\right\} \frac{\sigma_w}{\sqrt{n}} (\mathbf{W}^{(\ell+1)} - \tilde{\mathbf{W}}^{(\ell+1)}) \right\| \\
& + \alpha K_2 \left\| \left(\frac{\sigma_v}{\sqrt{n}} \mathbf{V}^{(\ell+1)} \mathrm{diag}\left\{\phi'(\mathbf{g}^{(\ell+1)})\right\} - \frac{\sigma_v}{\sqrt{n}} \tilde{\mathbf{V}}^{(\ell+1)} \mathrm{diag}\left\{\phi'(\tilde{\mathbf{g}}^{(\ell+1)})\right\} \right) \frac{\sigma_w}{\sqrt{n}} \tilde{\mathbf{W}}^{(\ell+1)} \right\|
\end{aligned} \tag{B.20}$$

$$\begin{aligned}
& \leq (c_4 + \alpha K_2 3\sigma_v \frac{\sigma_w}{\sqrt{n}} C_\phi)\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2 \\
& \quad + c_5 \left(\left\| \frac{\sigma_v}{\sqrt{n}} (\mathbf{V}^{(\ell+1)} - \tilde{\mathbf{V}}^{(\ell+1)}) \mathrm{diag}\left\{\phi'(\tilde{\mathbf{g}}^{(\ell+1)})\right\} \right\| + \left\| \frac{\sigma_v}{\sqrt{n}} \mathbf{V}^{(\ell+1)} (\mathrm{diag}\left\{\phi'(\mathbf{g}^{(\ell+1)})\right\} - \mathrm{diag}\left\{\phi'(\tilde{\mathbf{g}}^{(\ell+1)})\right\}) \right\| \right) \\
& \leq (c_6 + c_5 \frac{\sigma_v}{\sqrt{n}} C_\phi)\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2 + 3c_5\sigma_v C_\phi \left\| \mathbf{g}^{(\ell+1)} - \tilde{\mathbf{g}}^{(\ell+1)} \right\|_2 \\
& \leq (c_6 + c_5 \frac{\sigma_v}{\sqrt{n}} C_\phi)\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2 + 3c_5\sigma_v C_\phi c_3^L \|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2 \leq c_7\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2.
\end{aligned} \tag{B.21}$$

Therefore, we have that for all $\ell \in [L] : \|\boldsymbol{\delta}^{(\ell)} - \tilde{\boldsymbol{\delta}}^{(\ell)}\|_2 \leq c_7^L\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2 \leq \overline{K}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2$.

It is left to prove that $\left\|\boldsymbol{\gamma}^{(\ell)} - \tilde{\boldsymbol{\gamma}}^{(\ell)}\right\|_2 \leq \overline{K}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2$. This is achieved by the previous results for $\|\boldsymbol{\delta}^{(\ell)} - \tilde{\boldsymbol{\delta}}^{(\ell)}\|_2$ and $\|\mathbf{g}^{(\ell)} - \tilde{\mathbf{g}}^{(\ell)}\|_2$:

$$\begin{aligned}
&\left\|\boldsymbol{\gamma}^{(\ell)} - \tilde{\boldsymbol{\gamma}}^{(\ell)}\right\|_2 = \left\|\alpha\frac{\sigma_v\sigma_w}{\sqrt{n}}\mathrm{diag}\left\{\phi'(\mathbf{g}^{(\ell)})\right\}\mathbf{V}^{(\ell)\top}\boldsymbol{\delta}^{(\ell)} - \alpha\frac{\sigma_v\sigma_w}{\sqrt{n}}\mathrm{diag}\left\{\phi'(\tilde{\mathbf{g}}^{(\ell)})\right\}\tilde{\mathbf{V}}^{(\ell)\top}\tilde{\boldsymbol{\delta}}^{(\ell)}\right\|_2 \qquad \text{(B.22)} \\
&\leq \alpha\sigma_w\left\|\frac{\sigma_v}{\sqrt{n}}\mathrm{diag}\left\{\phi'(\mathbf{g}^{(\ell)})\right\}\mathbf{V}^{(\ell)\top}(\boldsymbol{\delta}^{(\ell)} - \tilde{\boldsymbol{\delta}}^{(\ell)})\right\|_2 \\
&\quad + \alpha\sigma_w\left\|\frac{\sigma_v}{\sqrt{n}}\left(\mathrm{diag}\left\{\phi'(\mathbf{g}^{(\ell)})\right\}\mathbf{V}^{(\ell)\top} - \mathrm{diag}\left\{\phi'(\tilde{\mathbf{g}}^{(\ell)})\right\}\tilde{\mathbf{V}}^{(\ell)\top}\right)\tilde{\boldsymbol{\delta}}^{(\ell)}\right\|_2 \\
&\leq 3\alpha\sigma_w\sigma_v C_\phi\|\boldsymbol{\delta}^{(\ell)} - \tilde{\boldsymbol{\delta}}^{(\ell)}\|_2 \\
&\quad + \alpha\sigma_w K_2\left(\left\|\frac{\sigma_v}{\sqrt{n}}(\mathbf{V}^{(\ell)} - \tilde{\mathbf{V}}^{(\ell)})\mathrm{diag}\left\{\phi'(\mathbf{g}^{(\ell)})\right\}\right\| + \left\|\frac{\sigma_v}{\sqrt{n}}\tilde{\mathbf{V}}^{(\ell)}(\mathrm{diag}\left\{\phi'(\mathbf{g}^{(\ell)})\right\} - \mathrm{diag}\left\{\phi'(\tilde{\mathbf{g}}^{(\ell)})\right\})\right\|\right) \\
&\leq (c_8c_7^L + c_9\frac{\sigma_v}{\sqrt{n}}C_\phi)\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2 + 3c_9\sigma_v C_\phi\left\|\mathbf{g}^{(\ell+1)} - \tilde{\mathbf{g}}^{(\ell+1)}\right\|_2 \\
&\leq c_{10}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2 \leq \overline{K}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2.
\end{aligned}$$

To conclude, we showed that for $\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}} \in B(\boldsymbol{\theta}_0, C)$ there exists $\overline{K} > 0$ (that does not depend on $\boldsymbol{\theta}, \tilde{\boldsymbol{\theta}}, n$) such that all the distances $\left\|\boldsymbol{\gamma}^{(\ell)} - \tilde{\boldsymbol{\gamma}}^{(\ell)}\right\|_2$, $\|\mathbf{x}^{(\ell)} - \tilde{\mathbf{x}}^{(\ell)}\|_2$, $\left\|\boldsymbol{\delta}^{(\ell)} - \tilde{\boldsymbol{\delta}}^{(\ell)}\right\|_2$, $\|\mathbf{z}^{(\ell)} - \tilde{\mathbf{z}}^{(\ell)}\|_2$ are upper bounded by $\overline{K}\|\boldsymbol{\theta} - \tilde{\boldsymbol{\theta}}\|_2$.

Appendix C. Additional Empirical NTK Results

C.1. Functions with Scalar Inputs

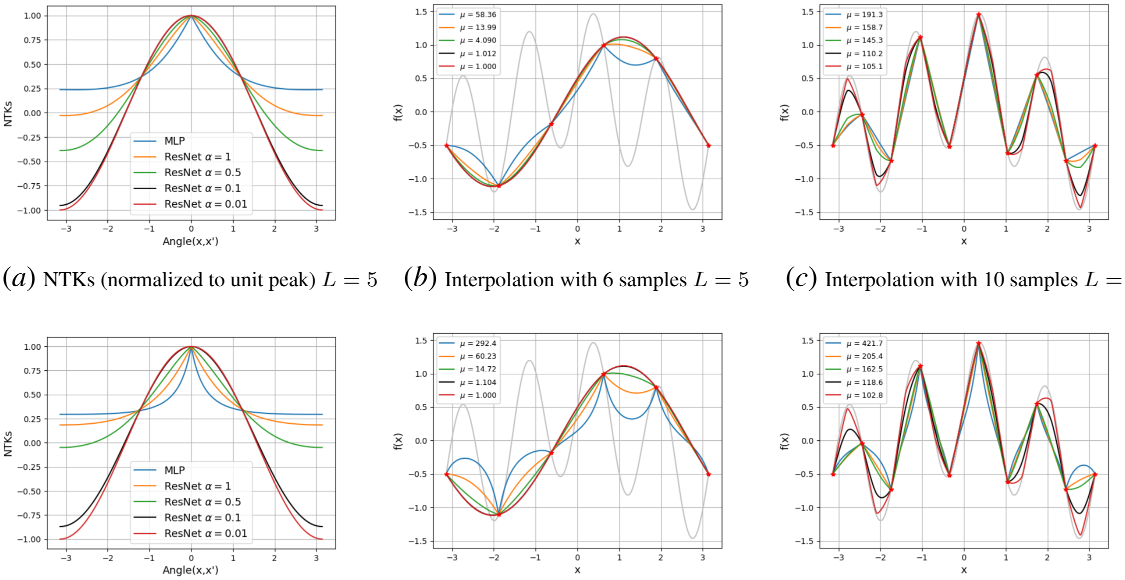
In this section we provide more experiments and details on the experimental setting that are missing in the main body of the paper, due to space limitation.

First, let us state the underlying ground truth function whose samples are used in the interpolation experiments

$$f(\beta) = \frac{1}{2}\cos(\beta) + \sin(4\beta), \qquad -\pi \leq \beta \leq \pi. \tag{C.1}$$

We treat the samples $\{\beta_i\}$ as points on the sphere (circle) in $\mathbb{R}^2$, i.e., samples of $\{\mathbf{x} \in \mathbb{R}^2 : x_1^2 + x_2^2 = 1\}$, since any point on the sphere has 1-to-1 mapping to an angle β, and vice versa ($\beta \rightarrow (\cos\beta, \sin\beta)$). This is motivated by the proof in (Jacot et al., 2018) that restricting the NTK to the unit sphere yields $\lambda_{min}(\mathbf{\Theta}) > 0$, which is required for the NTK theory. Note that we do not examine or compare reconstruction errors in this paper, and $f(\beta)$ is given here for reproducibility reasons.

Next, we present in Figure 6 an extended version of Figure 2 that includes also the results of ResNet NTK for an extremely small value of α, namely $\alpha = 0.01$. It can be seen that this small α does not significantly affect the kernel shape and interpolations' smoothness compared to the moderate $\alpha = 0.1$.



(a) NTKs (normalized to unit peak) $L = 5$

(b) Interpolation with 6 samples $L = 5$

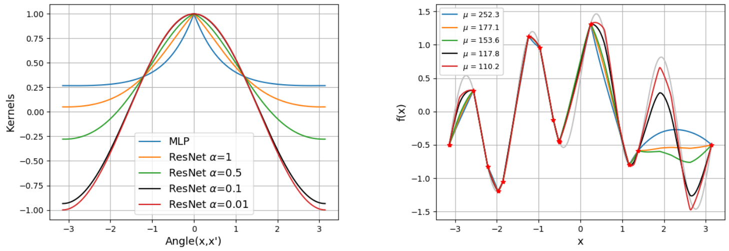
(c) Interpolation with 10 samples $L = 5$

(d) NTKs (normalized to unit peak) $L = 15$

(e) Interpolation with 6 samples $L = 15$

(f) Interpolation with 10 samples $L = 15$

Figure 6: NTKs for MLP and ResNet (for different values of α) with $L = 5$ (top) and $L = 15$ (bottom) nonlinear layers, ReLU nonlinearities, $\sigma_v = \sigma_w = 1$, for inputs on the sphere (circle) in $\mathbb{R}^2$. (6(a)),(6(d)): The kernels shape. (6(b))-(6(c)), (6(e))-(6(f)): Interpolations by the closed-form solutions, measured by $\mu(\cdot)$ defined in (22). Note that the legend in (6(a)),(6(d)) applies to all the figures.



(a) Kernels (normalized to unit peak) (b) Interpolation with 15 random samples

Figure 7: NTKs for MLP and ResNet (for different values of α) with $L = 7$ nonlinear layers, ReLU nonlinearities, $\sigma_v = \sigma_w = 1$, for inputs on the sphere (circle) in $\mathbb{R}^2$. (7(a)): The kernels shape. (7(b)): Interpolations by the closed-form solutions, measured by $\mu(\cdot)$ defined in (22). Note that the legend in (7(a)) applies to all the figures.

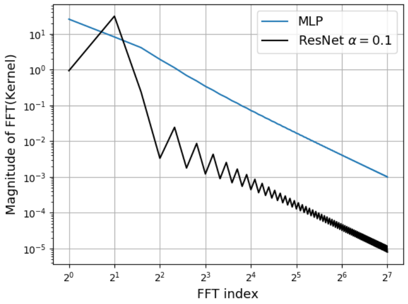


Figure 8: Magnitude of the first 128 FFT elements (out of 4096) of the NTKs for MLP and ResNet (with $\alpha = 0.1$) with $L = 5$ nonlinear layers, ReLU nonlinearities, $\sigma_v = \sigma_w = 1$, for inputs on the sphere (circle) in $\mathbb{R}^2$.

More NTK results, this time for $L = 7$ nonlinear layers and 15 random samples, are presented in Figure 7.

Finally, in Figure 8 we present, in logarithmic scale, the FFT spectrums of the NTKs of ResNet with $\alpha = 0.1$ and MLP, both with $L = 5$ nonlinear layers. This figure demonstrates our claim below (22): Since the decay rates of the FFT coefficients of the different kernels are approximately different only by a factor, we find the measure $\mu(f)$, *which depends also on the resulted interpolation and not only on the kernel*, to be more informative than comparing the FFT of the kernels. For example, multiplying the ResNet NTK by a large constant factor will make the magnitude of its FFT larger than the magnitude of the FFT of MLP NTK. Yet, it will not change the results of the kernel regression.

C.2. Functions with Multidimensional Inputs

In this section we briefly demonstrate the smoothness distinction between ResNet and MLP NTKs for multidimensional samples.

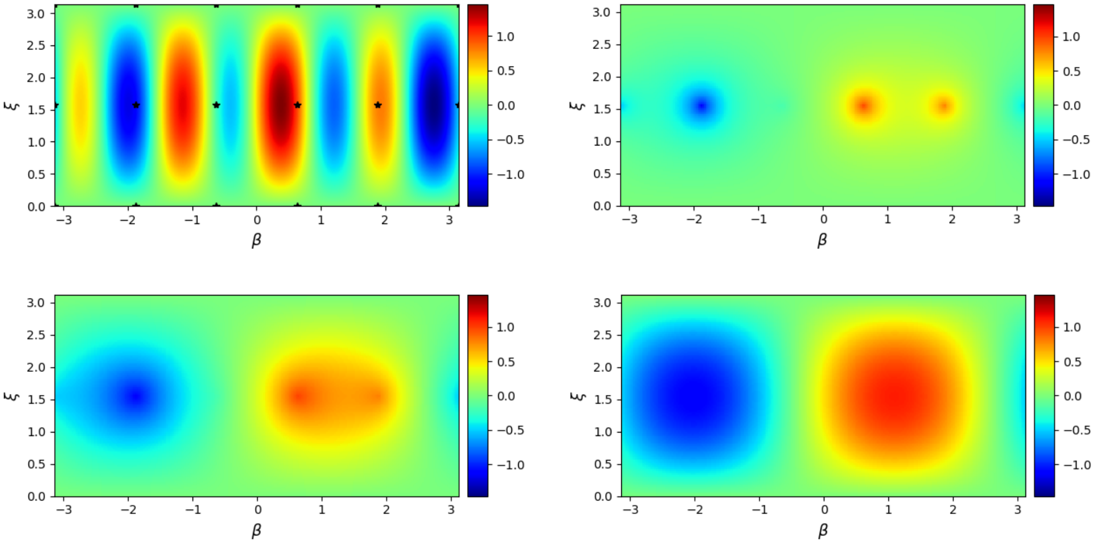


Figure 9: Two-dimensional interpolations from 6 × 3 samples using MLP and ResNet NTKs with $L = 15$ nonlinear layers, ReLU nonlinearities, $\sigma_v = \sigma_w = 1$, for inputs on the sphere in $\mathbb{R}^3$. From left to right and top to bottom: the underlying function with 6×3 samples, interpolation by the MLP NTK ($\mu = 1.47\text{e}4$), interpolation by the ResNet NTK with $\alpha = 1$ ($\mu = 2.33\text{e}3$), and interpolation by the ResNet NTK with $\alpha = 0.1$ ($\mu = 1.55\text{e}1$).

First, we extend the previous one-dimensional experiments to two dimensions. We modify the underlying ground truth function in (C.1) to

$$f(\beta, \xi) = \left(\frac{1}{2}\cos(\beta) + \sin(4\beta)\right)\sin(\xi), \qquad -\pi \leq \beta \leq \pi, \;\; 0 \leq \xi \leq \pi. \tag{C.2}$$

Here, the samples $\{(\beta_i, \xi_i)\}$ can be treated as points on the sphere in $\mathbb{R}^3$, i.e., samples of $\{\mathbf{x} \in \mathbb{R}^3 : \|\mathbf{x}\|_2 = 1\}$, since any point on the sphere has 1-to-1 mapping to a pair of angles (β, ξ), and vice versa $((\beta, \xi) \to (\cos\beta\sin\xi, \sin\beta\sin\xi, \cos\xi))$. Recall that for unique samples (i.e., $\mathbf{x}_i \neq \mathbf{x}_j$ for $i \neq j$) from the unit sphere we get $\lambda_{min}(\mathbf{\Theta}) > 0$, which is required for the NTK theory (Jacot et al., 2018).

Figures 9 and 10 show the interpolation results of the ResNet NTK with different values of α and the MLP NTK, for 6×3 uniform samples and for 10×5 uniform samples. We also report there the measure $\mu(\cdot)$ defined in (22) (straightforwardly extended to the 2D case). Similar to the one-dimensional case, it can be seen that the interpolation results of ResNet NTK are smoother, especially with small α.

We turn to examine the NTKs for high-dimensional data; specifically, the MNIST dataset, which includes images of size 28×28 of handwritten digits. As visualizing and measuring smoothness is difficult for high-dimensional data, we consider binary classification tasks, in which we interpolate ("overfit") the training set and report accuracy results on the test set. This links the different smoothness of the different NTKs with their generalization.

We consider the digits "0" and "8" and label them with $y \in \{+1, -1\}$. The training set includes $N/2$ samples from each class, where $N \in \{50, 100\}$, and the test set includes 1000 images from

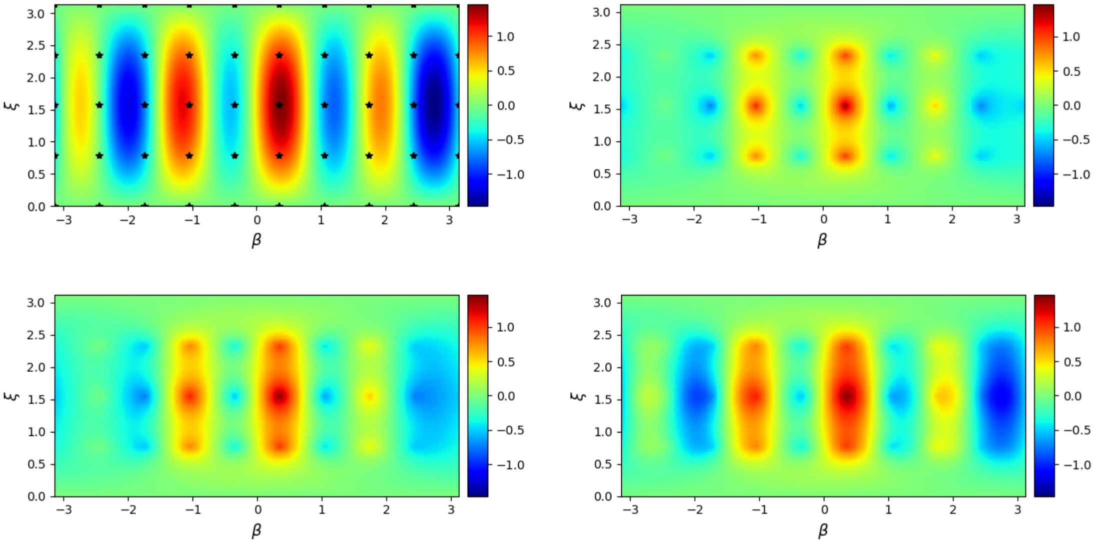
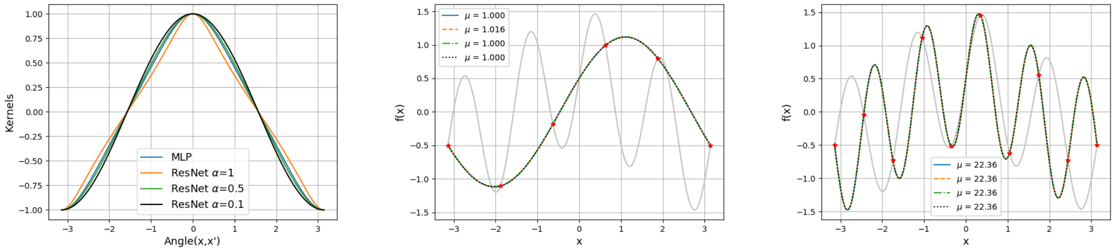


Figure 10: Two-dimensional interpolations from 10×5 samples using MLP and ResNet NTKs with $L = 15$ nonlinear layers, ReLU nonlinearities, $\sigma_v = \sigma_w = 1$, for inputs on the sphere in $\mathbb{R}^3$. From left to right and top to bottom: the underlying function with 10×5 samples, interpolation by the MLP NTK ($\mu = 2.59\text{e}4$), interpolation by the ResNet NTK with $\alpha = 1$ ($\mu = 1.24\text{e}4$), and interpolation by the ResNet NTK with $\alpha = 0.1$ ($\mu = 7.18\text{e}3$).

each class. The samples are presented as vectors in $\mathbb{R}^{28^2}$. The mean vector of the training set is subtracted from all samples and each sample is normalized to have unit Euclidean norm. We examine the MLP NTK and the ResNet NTK with $\alpha \in \{1, 0.1\}$ and $L \in \{5, 15, 30\}$. The classification of a test sample is made according to the sign of the ℓ_2 kernel regression (using the closed-form expression in (4)). The results are presented in Table 1. They demonstrate the advantage of using a smoother ResNet NTK.

Table 1: Accuracy results of different NTKs for binary MNIST classification tasks.

50 training samples	MLP NTK	ResNet NTK $\alpha = 1$	ResNet NTK $\alpha = 0.1$
$L = 5$	0.9625	0.967	0.972
$L = 15$	0.9615	0.9625	0.972
$L = 30$	0.958	0.9615	0.9715
100 training samples	MLP NTK	ResNet NTK $\alpha = 1$	ResNet NTK $\alpha = 0.1$
$L = 5$	0.981	0.9885	0.9905
$L = 15$	0.973	0.982	0.9925
$L = 30$	0.9645	0.9795	0.991



(a) NTKs (normalized to unit peak) $L = 5$ (b) Interpolation with 6 samples $L = 5$ (c) Interpolation with 10 samples $L = 5$

Figure 11: NTKs for MLP and ResNet (for different values of α) with $L = 5$ nonlinear layers, erf nonlinearities, $\sigma_v = \sigma_w = 1$, for inputs on the sphere (circle) in $\mathbb{R}^2$. (11(a)): The kernels shape. (11(b))-(11(c)): Interpolations by the closed-form solutions, measured by $\mu(\cdot)$ defined in (22). Note that the legend in (11(a)) applies to all the figures.

Appendix D. Closed-Form T and $\dot{T}$ Expressions for ReLU Nonlinearities

For completeness, we present here the closed-form expression of $T(\mathbf{K})$ and $\dot{T}(\mathbf{K})$ for $\phi(\cdot)$ which is the ReLU activation function. These results are due to (Cho and Saul, 2009).

Let $\mathbf{K} := \begin{bmatrix} K_{11} & K_{12} \\ K_{12} & K_{22} \end{bmatrix}$ be a 2×2 positive semidefinite matrix, $\rho := \frac{K_{12}}{\sqrt{K_{11}K_{22}}}$, and recall the definitions $T(\mathbf{K}) := \mathbb{E}_{(u,v)\sim\mathcal{N}(\mathbf{0},\mathbf{K})}\left[\phi(u)\phi(v)\right]$ and $\dot{T}(\mathbf{K}) := \mathbb{E}_{(u,v)\sim\mathcal{N}(\mathbf{0},\mathbf{K})}\left[\phi'(u)\phi'(v)\right]$. For the special case where $\phi(\cdot) = \max\{0, \cdot\}$, we have

$$\begin{aligned} T(\mathbf{K}) &= \frac{1}{2\pi}\sqrt{K_{11}K_{22}}\left(\rho\left(\pi - \arccos\left(\rho\right)\right) + \sqrt{1-\rho^2}\right), \\ \dot{T}(\mathbf{K}) &= \frac{1}{2\pi}\left(\pi - \arccos\left(\rho\right)\right). \end{aligned} \tag{D.1}$$

Appendix E. NTK Experiments with ERF Nonlinearities

In this section, we present NTK experiments for the erf activation function. First, we present the closed-form expression of $T(\mathbf{K})$ and $\dot{T}(\mathbf{K})$ for $\phi(\cdot)$ which is the erf function.

Let $\mathbf{K} := \begin{bmatrix} K_{11} & K_{12} \\ K_{12} & K_{22} \end{bmatrix}$ be a 2×2 positive semidefinite matrix and recall the definitions $T(\mathbf{K}) := \mathbb{E}_{(u,v)\sim\mathcal{N}(\mathbf{0},\mathbf{K})}\left[\phi(u)\phi(v)\right]$ and $\dot{T}(\mathbf{K}) := \mathbb{E}_{(u,v)\sim\mathcal{N}(\mathbf{0},\mathbf{K})}\left[\phi'(u)\phi'(v)\right]$. For the special case where $\phi(u) = \frac{2}{\sqrt{\pi}}\int_0^u \mathrm{e}^{-z^2}dz$, we have from (Williams, 1996) that

$$\begin{aligned} T(\mathbf{K}) &= \frac{2}{\pi}\arcsin\left(\frac{2\mathrm{K}_{12}}{\sqrt{(1+2\mathrm{K}_{11})(1+2\mathrm{K}_{22})}}\right), \\ \dot{T}(\mathbf{K}) &= \frac{4}{\pi}\det\left(\mathbf{I}_2 + 2\mathbf{K}\right)^{-1/2}. \end{aligned} \tag{E.1}$$

Next, we repeat several NTK experiments from Section 4.2, but with the erf activation instead of the ReLU activation. The other configurations are not changed. The results are presented in

Figure 11. It can be seen that with erf activations both MLP and ResNet NTKs have rather similar shapes and similar interpolation results (contrary to the ReLU case). This may imply that the smoothness distinction between the models requires using nonsmooth activations.