Proceedings of the 7th Symposium on Advances in Approximate Bayesian Inference, 2025 1–26

Observation Noise and Initialization in Wide Neural Networks

Anonymous Authors
Anonymous Institution

Abstract

Performing gradient descent in a wide neural network is equivalent to computing the posterior mean of a Gaussian Process with the Neural Tangent Kernel (NTK-GP), for a specific choice of prior mean and with zero observation noise. However, existing formulations of this result have two limitations: i) the resultant NTK-GP assumes no noise in the observed target variables, which can result in suboptimal predictions with noisy data; ii) it is unclear how to extend the equivalence to an arbitrary prior mean, a crucial aspect of formulating a well-specified model. To address the first limitation, we introduce a regularizer into the neural network's training objective, formally showing its correspondence to incorporating observation noise into the NTK-GP model. To address the second, we introduce a *shifted network* that enables arbitrary prior mean functions. This approach allows us to perform gradient descent on a single neural network, without expensive ensembling or kernel matrix inversion. Our theoretical insights are validated empirically, with experiments exploring different values of observation noise and network architectures.

1. Introduction

The connection between wide neural networks and Gaussian Processes via the Neural Tangent Kernel (NTK) (Jacot et al., 2018) provides a powerful framework for understanding training dynamics in deep learning. Lee et al. (2019) showed that a wide neural network trained with gradient flow/descent on mean squared error (MSE) aligns with the posterior mean of an NTK-GP. However, this result has two key limitations: it assumes zero observation noise, leading to model misspecification, and it only holds for a specific prior mean—namely, a randomly initialized network. Observation noise, or aleatoric uncertainty, is crucial in probabilistic models like GPs (Williams and Rasmussen, 2006), as real-world data is inherently noisy due to measurement errors and annotation ambiguities (Kendall and Gal, 2017). In GPs, this uncertainty is captured through a variance term, ensuring robust and well-calibrated predictions.

Hu et al. (2020) introduced a regularizer penalizing deviations from initialization, demonstrating improved performance in noisy settings. However, their analysis assumed the network remains in a linear regime throughout training without proving this assumption. This is critical, as the regularizer alters training dynamics, making previous results from Lee et al. (2019) inapplicable. Other works have explored this regularizer for generalization and stability improvements (Nitanda and Suzuki (2020); Suh et al. (2021); He et al. (2020)), yet all rely on unverified linearity assumptions (see Appendix A for a more detailed related work section). In this work, we formally show that the regularizer preserves network linearity while introducing non-zero aleatoric noise into the NTK-GP mean posterior. This ensures the NTK-GP posterior mean properly accounts for observation noise, aligning it with real-world data and supporting the observed generalization benefits of regularization.

To address the problem of supporting inference with an arbitrary prior mean, we propose the use of a *shifted network* during training. This approach provides a principled strategy to eliminate initialization randomness, ensuring deterministic convergence of a single *shifted network* to the posterior mean of the defined NTK-GP prior. In essence, we investigate the following question:

Main Research Question. Is there a loss function such that performing gradient descent on that loss gives us a network with predictions

$$m(\mathbf{x}') + \hat{\mathbf{\Theta}}_{\mathbf{x}',\mathbf{x}} \left(\hat{\mathbf{\Theta}}_{\mathbf{x},\mathbf{x}} + \beta I\right)^{-1} (\mathbf{y} - m(\mathbf{x})),$$

for an arbitrary prior mean $m(\mathbf{x})$ and for arbitrary values of observation noise β?

We answer this question in the affirmative. Specifically, we build up the theory for supporting $\beta > 0$ in Section 3 (Theorems 1 and 2) and then add the theory for supporting arbitrary prior means in Section 4 (Theorem 3).

2. Preliminaries

Neural Network Parameterization. The choice of parameterization affects how signals propagate and how gradients scale with width. Let $\phi : \mathbb{R} \to \mathbb{R}$ be a Lipschitz activation function. In the standard parameterization, layer outputs are given by

$$h^{l+1} := W^{l+1} x^l + b^{l+1}, \quad x^{l+1} := \phi(h^{l+1}) \in \mathbb{R}^{n_{l+1}}, \tag{1}$$

where $W^{l+1} \in \mathbb{R}^{n_{l+1} \times n_l}$ and $b^{l+1} \in \mathbb{R}^{n_{l+1}}$. The parameter vector $\theta \in \mathbb{R}^p$ stacks all weights and biases. For simplicity, we consider scalar outputs ($k = 1$) and assume $n_1 = \cdots = n_L =: n$. Weights are initialized as $W^l_{0,ij} \sim \mathcal{N}(0, \frac{\sigma^2_{w,l}}{n_l})$, and biases as $b^l_{0,i} \sim \mathcal{N}(0, \sigma^2_{b,l})$. In this setting, the Jacobian norm diverges as $n_l \to \infty$.

The NTK parameterization rescales the weights:

$$h^{l+1} := \frac{1}{\sqrt{n_l}} W^{l+1} x^l + b^{l+1}, \quad x^{l+1} := \phi(h^{l+1}). \tag{2}$$

Here, $W^l_{0,ij} \sim \mathcal{N}(0, \sigma^2_{w,l})$ and $b^l_{0,i} \sim \mathcal{N}(0, \sigma^2_{b,l})$. This scaling ensures stable signal propagation in the infinite-width limit. Appendix C details the equivalence between parameterizations under proper learning rate selection.

Neural Tangent Kernel (NTK). The NTK describes the evolution of predictions in wide neural networks. Given $f(x, \theta)$ with parameters $\theta \in \mathbb{R}^p$, define the Jacobian $J(x, \theta) := \frac{\partial f(x,\theta)}{\partial \theta} \in \mathbb{R}^{N \times p}$. The empirical NTK is:

$$\hat{\mathbf{\Theta}}_{x',x} := J(x', \theta_0) J(x, \theta_0)^\top \in \mathbb{R}^{N' \times N}. \tag{3}$$

As $n \to \infty$, Jacot et al. (2018) showed that $\hat{\mathbf{\Theta}}_{x',x}$ converges to a deterministic kernel $\mathbf{\Theta}$, which remains constant under unregularized gradient flow. This defines an NTK-GP, where the trained network mean aligns with the GP posterior mean.

3. Observation Noise through Regularized Gradient Descent

Weight-space regularization not only affects generalization (Hu et al., 2020) but also alters training dynamics. Lee et al. (2019) studied unregularized gradient flow, showing its role in aligning network convergence with Bayesian inference. Here, we analyze regularized gradient flow, proving that the training trajectory remains arbitrarily close to its linearized counterpart, providing insights into how regularization modifies learning.

Define $f(\theta) := f(\mathbf{x}, \theta)$, $g(\theta) := f(\mathbf{x}, \theta) - \mathbf{y} \in \mathbb{R}^N$, $J(\theta) := J(\mathbf{x}, \theta) \in \mathbb{R}^{N \times p}$ in the training points.[1] We will consider (for NTK parametrization[2]) the regularized training loss

$$\mathcal{L}^\beta(\theta) := \frac{1}{2}\sum_{i=1}^{N}(f(\mathbf{x}_i, \theta) - \mathbf{y}_i)^2 + \frac{1}{2}\beta\|\theta - \theta_0\|_2^2 = \frac{1}{2}\|g(\theta)\|_2^2 + \frac{1}{2}\beta\|\theta - \theta_0\|_2^2. \tag{4}$$

The gradient of this loss is given by

$$\nabla_\theta \mathcal{L}^\beta(\theta) = J(\theta)^\top g(\theta) + \beta(\theta - \theta_0). \tag{5}$$

We study the training dynamics under the regularized gradient flow[3][4]

$$\frac{d\theta_t}{dt} = -\eta \nabla_\theta \mathcal{L}^\beta(\theta_t). \tag{6}$$

By the chain rule, $\frac{df(x,\theta_t)}{dt} = J(x, \theta_t)\frac{d\theta_t}{dt}$, and thus the dynamics of the network are

$$\frac{df(x, \theta_t)}{dt} = -\eta\left(J(x, \theta_t)J(\theta_t)^\top g(\theta_t) + \beta J(x, \theta_t)(\theta_t - \theta_0)\right). \tag{7}$$

For the sake of readability, we omit the dependence of θ_t on θ_0 and β. To gain insights into the role of regularization, we first analyze the regularized gradient flow for the linearized network

$$f_{\theta_0}^{\text{lin}}(x, \theta) := f(x, \theta_0) + J(x, \theta_0)(\theta - \theta_0). \tag{8}$$

This is a linear ODE and hence has a closed-form solution. We formalize this in the following theorem.

Theorem 1 *For training time $t \to \infty$, at any point $\mathbf{x}'$,*

$$f_{\theta_0}^{\text{lin}}(\mathbf{x}', \theta_\infty) = f(\mathbf{x}', \theta_0) + \hat{\mathbf{\Theta}}_{\mathbf{x}',\mathbf{x}}\left(\hat{\mathbf{\Theta}}_{\mathbf{x},\mathbf{x}} + \beta I\right)^{-1}(\mathbf{y} - f(\mathbf{x}, \theta_0)). \tag{9}$$

We derive this solution (for gradient flow and gradient descent) in Appendix B.

If the network initialization is treated as a random variable, the trained linearized network converges to a normal distribution as $n \to \infty$, with a mean matching the NTK-GP posterior

1. We assume that $\mathbf{x}_i \neq \mathbf{x}_j$ for $i \neq j$.
2. See Appendix C.2 for standard parametrization.
3. In Appendix F, we state equivalent results for gradient descent with small enough learning rates. These will involve geometric sums instead of the exponential.
4. We will prove that this ODE has a unique solution under our assumptions.

mean for a zero prior. However, as noted in (Lee et al. (2019); He et al. (2020)), its covariance does not correspond to an NTK-GP posterior covariance (see Appendix H).

This section aims to prove convergence of the training dynamics of neural networks to those of a linearized network for large enough n. For this, we leverage the Lipschitzness of the Jacobian (Appendix D) and derive new results to prove that the parameter deviation from initialization remains $O(1)$ (Appendix E.1). Lastly, using these results, we conclude that the regularized gradient flow and its linearized counterpart remain arbitrarily close.

3.1. Closeness to the Linearized Network along the Regularized Gradient Flow

We now show that the parameters θ_t under regularized gradient flow remain arbitrarily close to the parameters θ_t^{lin} obtained from the gradient flow applied to the linearized network for large enough layer width. Using this, we can further prove that the neural network trained from initial parameters θ_0 is arbitrarily close to its linearization around θ_0.

Theorem 2 *Let $\beta \geq 0$. Let $\delta_0 > 0$ be arbitrarily small. Then, there are C_1, C_2, such that for n large enough, with probability of at least $1 - \delta_0$ over random initialization,*

$$\sup_{t \geq 0} \|\theta_t - \theta_t^{\text{lin}}\|_2 \leq C_1 \frac{1}{\sqrt{n}}, \tag{10}$$

$$\forall \|x\|_2 \leq 1 : \sup_{t \geq 0} \|f(x, \theta_t) - f_{\theta_0}^{\text{lin}}(x, \theta_t^{\text{lin}})\|_2 \leq C_2 \frac{1}{\sqrt{n}}. \tag{11}$$

See Appendix E.2 for a proof. Unlike previous proofs for $\beta = 0$, which rely on the exponential decay of $\|g(\theta_t)\|_2$, we use a more general approach valid for $\beta \geq 0$. We decompose $\|f(x, \theta_t) - f_{\theta_0}^{\text{lin}}(x, \theta_t^{\text{lin}})\|_2$ into two terms and bound them using Theorem 8 (from Appendix E.1). These results establish that wide networks under regularized gradient flow remain in a linear regime, allowing us to apply Theorem 1 to analyze training with regularization.

3.2. NTK-GP Posterior Mean with Aleatoric Noise Interpretation

At convergence under regularized gradient flow, the output of the linearized network corresponds to the NTK-GP posterior mean with non-zero observation noise, providing a Bayesian interpretation of regularized training. While the trained parameters θ_∞ depend on the random initialization θ_0, this variability is not central to our analysis. Unlike deep ensembles (He et al., 2020), or Thompson sampling (Thompson, 1933), which explicitly leverage initialization randomness, our focus is on the behaviour of a single trained network. The initialization strategy we define next removes this randomness, ensuring deterministic convergence of an individual network to the posterior mean of the NTK-GP.

4. Neural Network Initialization as NTK-GPs with Arbitrary Prior Mean

Standard neural network initialization typically involves randomly setting weights and biases to break symmetry but lacks a principled way to encode specific inductive biases or desired

properties. In contrast, Gaussian processes provide a structured framework for defining prior distributions over functions through the prior mean and covariance. One approach to obtaining the NTK-GP posterior with zero prior mean is to train an ensemble of networks and average their outputs (He et al., 2020). However, if only the posterior mean is needed, this can be achieved more efficiently with a single network. Section 4.1 explores how, under the NTK-GP framework, neural networks can be initialized to reflect arbitrary prior means, allowing greater flexibility for different tasks.

4.1. Shifting the labels or predictions

Inspired by standard techniques in GP literature Williams and Rasmussen (2006), we provide a formal construction for modifying the network to introduce arbitrary prior mean. This can be accomplished either by shifting the predictions of the neural network by its predictions at initialization, or equivalently, defining a new shifted network. The following theorem formalizes this, resolving the main research question we posed in Section 1.

Theorem 3 *(Shifted Network.) Consider any function m. Given a random initialization θ_0, define shifted predictions $\tilde{f}_{\theta_0}(\mathbf{x}, \theta)$ as follows:*

$$\tilde{f}_{\theta_0}(\mathbf{x}, \theta) := f(\mathbf{x}, \theta) - f(\mathbf{x}, \theta_0) + m(\mathbf{x}). \tag{12}$$

Training this modified network (starting with θ_0) leads to the following output (in the infinite-width limit)

$$\tilde{f}_{\theta_0}(\mathbf{x}', \theta_\infty) = m(\mathbf{x}') + \mathbf{\Theta}_{\mathbf{x}',\mathbf{x}}(\mathbf{\Theta}_{\mathbf{x},\mathbf{x}} + \beta I)^{-1}(\mathbf{y} - m(\mathbf{x})). \tag{13}$$

This can be interpreted as the posterior mean of an NTK-GP with prior mean function m.

We prove this in Appendix G. Unlike the standard network, where $f_{\theta_0}(\mathbf{x}', \theta_\infty)$ is random but has an NTK-GP posterior mean with prior 0, the shifted network's output $\tilde{f}_{\theta_0}(\mathbf{x}', \theta_\infty)$ is deterministic. This follows from the NTK being independent of initialization in the infinite-width limit.

A drawback of shifting predictions is the need to store θ_0, doubling memory usage, which may be prohibitive for large networks. Computing the shift also adds a minor overhead due to the initial forward pass. However, no additional back-propagation is required, making this approach more efficient than training an ensemble.

5. Experiments

We empirically validate our results by studying the convergence of wide neural networks trained with regularized gradient descent to their linearized counterparts. We measure how trained parameters deviate from the optimal linearized network parameters and how this difference shrinks with width. Additionally, we compare the trained network's predictions to kernel ridge regression while varying depth and regularization strength β.

We train fully connected MLPs under the NTK parametrization using full-batch gradient descent on a synthetic regression task: $y = \sin(x) + \cos(2x) + \epsilon$, where $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Inputs x are sampled from $[-6, 6]$, and we use a two-layer network. We compute the l_2 norm between trained parameters and those from kernel ridge regression and evaluate prediction error on a test set.

5.1. Empirical Convergence of the Parameters

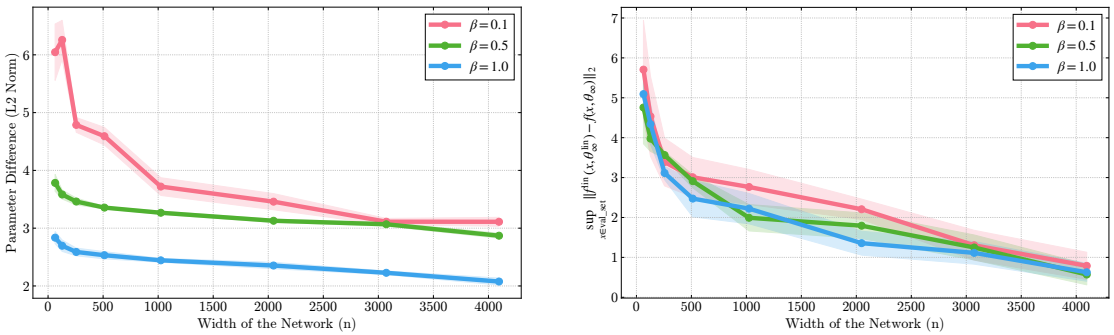

Figure 1: Comparison of parameter and function differences between trained neural networks and their linearized counterparts. Left: Deviation in parameter space. Right: Discrepancy in function space across validation points. Shaded regions represent standard deviation divided by the square root of the number of seeds.

We compare the trained network parameters to those obtained via kernel ridge regression with the NTK by computing their ℓ_2 norm difference. Using Theorem 3, we set the prior mean to zero. As shown in Figure 1 (left), this difference decreases with increasing width, supporting Theorem 2.

For smaller widths, deviations from the linearized solution highlight finite-width effects, but these diminish as width grows. Appendix I provides additional results for varying depths. Notably, inverting the NTK matrix posed a computational bottleneck, limiting scalability, whereas training the network remained efficient even with full gradient descent.

5.2. Empirical Convergence of the Trained Network to a Linear Model

We now examine function-space convergence by comparing the trained network's predictions to those of the corresponding linearized model on unseen data. The validation set, comprising 20% of the dataset, was sampled from the same distribution as the training data. Specifically, we compute $\sup_{x\in\mathcal{V}}\|f(x,\theta_\infty) - f^{\text{lin}}(x,\theta_\infty^{\text{lin}})\|_2$, which quantifies the deviation between the trained network and the kernel ridge regression solution.

Figure 1 (right) shows that this discrepancy decreases with increasing width, further supporting Theorem 2. Additional results for different depths are in Appendix I.

6. Conclusion and Future Work

We analyzed regularized training in wide neural networks under the NTK framework, proving that weight-space regularization is equivalent to adding aleatoric noise to the NTK-GP posterior mean in the infinite-width limit. We also introduced a shifted network approach that enables arbitrary prior functions and ensures deterministic convergence without ensembles or kernel inversion. Empirical results confirm our theoretical findings.

Future work could explore our regularizer in architectures with NTK convergence, such as convolutional and residual networks Arora et al. (2019); Belfer et al. (2024); Yang (2020).

References

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International conference on machine learning*, pages 242–252. PMLR, 2019.

Sanjeev Arora, Simon S Du, Wei Hu, Zhiyuan Li, Russ R Salakhutdinov, and Ruosong Wang. On exact computation with an infinitely wide neural net. *Advances in neural information processing systems*, 32, 2019.

Yuval Belfer, Amnon Geifman, Meirav Galun, and Ronen Basri. Spectral analysis of the neural tangent kernel for deep residual networks. *Journal of Machine Learning Research*, 25(184):1–49, 2024.

Zonghao Chen, Xupeng Shi, Tim GJ Rudner, Qixuan Feng, Weizhong Zhang, and Tong Zhang. A neural tangent kernel perspective on function-space regularization in neural networks. In *OPT 2022: Optimization for Machine Learning (NeurIPS 2022 Workshop)*, 2022.

Bobby He, Balaji Lakshminarayanan, and Yee Whye Teh. Bayesian deep ensembles via the neural tangent kernel. *Advances in neural information processing systems*, 33:1010–1022, 2020.

Wei Hu, Zhiyuan Li, and Dingli Yu. Simple and effective regularization methods for training on noisily labeled data with generalization guarantee, 2020. URL https://arxiv.org/abs/1905.11368.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30, 2017.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. Deep neural networks as gaussian processes. *International Conference on Learning Representations*, 37, 2018.

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. *Advances in neural information processing systems*, 32, 2019.

Atsushi Nitanda and Taiji Suzuki. Optimal rates for averaged stochastic gradient descent under neural tangent kernel regime. *arXiv preprint arXiv:2006.12297*, 2020.

Tim G. J. Rudner, Sanyam Kapoor, Shikai Qiu, and Andrew Gordon Wilson. Function-space regularization in neural networks: A probabilistic perspective, 2023. URL https://arxiv.org/abs/2312.17162.

Namjoon Suh, Hyunouk Ko, and Xiaoming Huo. A non-parametric regression viewpoint: Generalization of overparametrized deep relu network under noisy observations. In *International Conference on Learning Representations*, 2021.

William R. Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, 25(3/4):285–294, 1933. ISSN 00063444. URL http://www.jstor.org/stable/2332286.

Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

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

Difan Zou, Yuan Cao, Dongruo Zhou, and Quanquan Gu. Gradient descent optimizes over-parameterized deep relu networks. *Machine learning*, 109:467–492, 2020.

Appendix A. Related Work

Linearization The paper by Lee et al. (2019) serves as the starting point for our work. It demonstrates that as network width approaches infinity, training dynamics simplify and can be approximated by a linearized model using a first-order Taylor expansion. Lee et al. (2019) also study the links between the output of trained neural networks and GPs. Crucially, we extend this work by proving that this linearization still holds in the presence of observation noise.

Kernel Methods and Neural Networks The seminal paper by Jacot et al. (2018) made two contributions. First, it introduced the equivalence between kernel methods and wide neural networks, specifically for the case of kernel ridge regression. Second, it popularized the study of neural networks in function space, rather than parameter space. We leverage both of these insights: our Theorem 3 considers the function-space view of a Gaussian Process (a kernel method). Several later studies have explored the links between wide neural networks and GPs through the NTK, to investigate the functional behaviour of neural networks in noisy settings. For example, Rudner et al. (2023) introduced function-space regularization to encode desired properties into predictions, indirectly addressing observation noise, while Chen et al. (2022) linked NTK-based function norms to RKHS regularization, proving to be useful in low-data regimes.

Global Minima and Overparameterization In the context of overparametrization, Allen-Zhu et al. (2019) prove that stochastic gradient descent (SGD) can find global minima for neural networks in polynomial time. Similarly, Zou et al. (2020) show that ReLU networks trained with SGD converge to global minima for a wide range of loss functions by ensuring that weight updates remain within a small perturbation around the initialization. While we do not rely on these results directly, our result is spiritually related in that we guarantee convergence to the global optimum with high probability.

Regularization A line of work has explored the role of regularization in wide neural networks through the lens of the NTK. Hu et al. (2020) introduced the regularizer penalizing deviations from initialization, providing generalization bounds in noisy settings but assuming network linearity without proof. Nitanda and Suzuki (2020) and Suh et al. (2021) extended this approach to constrain network dynamics and stabilize deeper architectures.

Bayesian Ensembles He et al. (2020) described a way of training Bayesian ensembles of neural networks, allowing for inference in the NTK-GP with zero prior mean by averaging the ensembles using the law of large numbers. In cases where we are only interested in obtaining the posterior mean, our approach is more efficient since we only train one network[5].

5. They additionally provide a way of estimating the posterior covariance, which is not of interest in our paper.

Appendix B. Regularized gradient flow and gradient descent for the linearized network

Consider the linearized network $f^{\text{lin}}(x,\theta) = f(x,\theta_0) + J(x,\theta_0)(\theta-\theta_0)$. For notational convenience, we drop the dependence on θ_0 throughout the Appendix. In the following, we will consider training the parameters using the regularized training loss

$$\mathcal{L}^{\beta,\text{lin}}(\theta) := \frac{1}{2}\sum_{i=1}^{N}(f^{\text{lin}}(\mathbf{x}_i,\theta)-\mathbf{y}_i)^2 + \frac{1}{2}\beta\|\theta-\theta_0\|_2^2. \tag{14}$$

B.1. Regularized gradient flow for the linearized network

The evolution of the parameters through gradient flow with learning rate η_0 is given by

$$\begin{aligned}
\frac{d\theta_t^{\text{lin}}}{dt} &= -\eta_0\left(J(\theta_0)^\top g^{\text{lin}}(\theta_t^{\text{lin}}) + \beta(\theta_t^{\text{lin}}-\theta_0)\right) && (15)\\
&= -\eta_0\left(J(\theta_0)^\top\left(f(\theta_0)+J(\theta_0)(\theta_t^{\text{lin}}-\theta_0)-\mathbf{y}\right)+\beta(\theta_t^{\text{lin}}-\theta_0)\right) && (16)\\
&= -\eta_0\left(J(\theta_0)^\top J(\theta_0)+\beta I_p\right)(\theta_t^{\text{lin}}-\theta_0) - \eta_0 J(\theta_0)^\top(f(\theta_0)-\mathbf{y}). && (17)
\end{aligned}$$

This is a multidimensional linear ODE in $\theta_t^{\text{lin}}-\theta_0$. Its unique solution is given by

$$\theta_t^{\text{lin}} = \theta_0 + \left(e^{-\eta_0\left(J(\theta_0)^\top J(\theta_0)+\beta I_p\right)t} - I_p\right)\left(-\eta_0\left(J(\theta_0)^\top J(\theta_0)+\beta I_p\right)\right)^{-1}\left(-\eta_0 J(\theta_0)^\top(f(\theta_0)-\mathbf{y})\right) \tag{18}$$

$$= \theta_0 + \left(I_p - e^{-\eta_0\left(J(\theta_0)^\top J(\theta_0)+\beta I_p\right)t}\right)\left(J(\theta_0)^\top J(\theta_0)+\beta I_p\right)^{-1} J(\theta_0)^\top(\mathbf{y}-f(\theta_0)) \tag{19}$$

$$= \theta_0 + \left(I_p - e^{-\eta_0\left(J(\theta_0)^\top J(\theta_0)+\beta I_p\right)t}\right) J(\theta_0)^\top\left(J(\theta_0)J(\theta_0)^\top+\beta I_N\right)^{-1}(\mathbf{y}-f(\theta_0)) \tag{20}$$

$$= \theta_0 + \left(J(\theta_0)^\top - e^{-\eta_0\left(J(\theta_0)^\top J(\theta_0)+\beta I_p\right)t} J(\theta_0)^\top\right)\left(J(\theta_0)J(\theta_0)^\top+\beta I_N\right)^{-1}(\mathbf{y}-f(\theta_0)) \tag{21}$$

$$= \theta_0 + J(\theta_0)^\top\left(I_N - e^{-\eta_0\left(J(\theta_0)J(\theta_0)^\top+\beta I_N\right)t}\right)\left(J(\theta_0)J(\theta_0)^\top+\beta I_N\right)^{-1}(\mathbf{y}-f(\theta_0)). \tag{22}$$

In the third and fourth equality, we used that for $k\in\mathbb{Z}$,

$$\left(J(\theta_0)^\top J(\theta_0)+\beta I_p\right)^k J(\theta_0)^\top = J(\theta_0)^\top\left(J(\theta_0)J(\theta_0)^\top+\beta I_N\right)^k. \tag{23}$$

Substituting θ_t^{lin} into the formula for the linearized network, we get for any point $\mathbf{x}'$,

$$f^{\text{lin}}(\mathbf{x}',\theta_t^{\text{lin}}) \tag{24}$$

$$= f(\mathbf{x}',\theta_0) + J(\mathbf{x}',\theta_0)J(\theta_0)^\top\left(I_N - e^{-\eta_0\left(J(\theta_0)J(\theta_0)^\top+\beta I_N\right)t}\right)\left(J(\theta_0)J(\theta_0)^\top+\beta I_N\right)^{-1}(\mathbf{y}-f(\theta_0)) \tag{25}$$

$$= f(\mathbf{x}',\theta_0) + \hat{\mathbf{\Theta}}_{\mathbf{x}',\mathbf{x}}\left(I_N - e^{-\eta_0(\hat{\mathbf{\Theta}}_{\mathbf{x},\mathbf{x}}+\beta I_N)t}\right)\left(\hat{\mathbf{\Theta}}_{\mathbf{x},\mathbf{x}}+\beta I_N\right)^{-1}(\mathbf{y}-f(\theta_0)). \tag{26}$$

For training time $t\to\infty$, this gives

$$f^{\text{lin}}(\mathbf{x}',\theta_\infty^{\text{lin}}) = f(\mathbf{x}',\theta_0) + \hat{\mathbf{\Theta}}_{\mathbf{x}',\mathbf{x}}\left(\hat{\mathbf{\Theta}}_{\mathbf{x},\mathbf{x}}+\beta I_N\right)^{-1}(\mathbf{y}-f(\theta_0)). \tag{27}$$

B.2. Regularized gradient descent for the linearized network

Similarly to gradient flow, the evolution of the parameters through gradient descent (when training the regularized loss given by the linearized network) with learning rate η_0 is given by

$$\theta_t^{\text{lin}} = \theta_{t-1}^{\text{lin}} - \eta_0\left(J(\theta_0)^\top J(\theta_0)+\beta I_p\right)(\theta_{t-1}^{\text{lin}}-\theta_0) - \eta_0 J(\theta_0)^\top(f(\theta_0)-\mathbf{y}). \tag{28}$$

One may write this as

$$\theta_t^{\text{lin}} - \theta_0 = \left(I_p - \eta_0 \left(J(\theta_0)^\top J(\theta_0) + \beta I_p\right)\right)\left(\theta_{t-1}^{\text{lin}} - \theta_0\right) - \eta_0 J(\theta_0)^\top (f(\theta_0) - \mathbf{y}). \tag{29}$$

Applying the formula for $\theta_t^{\text{lin}} - \theta_0$ iteratively, leads to the geometric sum

$$\theta_t^{\text{lin}} - \theta_0 = -\eta_0 \sum_{u=0}^{t-1} \left(I_p - \eta_0 \left(J(\theta_0)^\top J(\theta_0) + \beta I_p\right)\right)^u J(\theta_0)^\top (f(\theta_0) - \mathbf{y}) \tag{30}$$

$$= \eta_0 J(\theta_0)^\top \sum_{u=0}^{t-1} \left(I_N - \eta_0 \left(J(\theta_0) J(\theta_0)^\top + \beta I_N\right)\right)^u (\mathbf{y} - f(\theta_0)) \tag{31}$$

$$= J(\theta_0)^\top \left(I_N - \left(I_N - \eta_0 \left(J(\theta_0) J(\theta_0)^\top + \beta I_N\right)\right)^t\right) \left(J(\theta_0) J(\theta_0)^\top + \beta I_N\right)^{-1} (\mathbf{y} - f(\theta_0)). \tag{32}$$

This converges for $t \to \infty$ if and only if $0 < \eta_0 < \frac{2}{\lambda_{\max}\left(J(\theta_0)J(\theta_0)^\top\right) + \beta}$. In that case, it converges (as expected) to the same $\theta_\infty^{\text{lin}}$ as that of the regularized gradient flow. Substituting θ_t^{lin} into the formula for the linearized network, we get for any point $\mathbf{x}'$,

$$f^{\text{lin}}(\mathbf{x}', \theta_t^{\text{lin}}) \tag{33}$$

$$= f(\mathbf{x}', \theta_0) + J(\mathbf{x}', \theta_0) J(\theta_0)^\top \left(I_N - \left(I_N - \eta_0 \left(J(\theta_0) J(\theta_0)^\top + \beta I_N\right)\right)^t\right) \left(J(\theta_0) J(\theta_0)^\top + \beta I_N\right)^{-1} (\mathbf{y} - f(\theta_0)) \tag{34}$$

$$= f(\mathbf{x}', \theta_0) + \hat{\mathbf{\Theta}}_{\mathbf{x}', \mathbf{x}} \left(I_N - \left(I_N - \eta_0 \left(\hat{\mathbf{\Theta}}_{\mathbf{x}, \mathbf{x}} + \beta I_N\right)\right)^t\right) \left(\hat{\mathbf{\Theta}}_{\mathbf{x}, \mathbf{x}} + \beta I_N\right)^{-1} (\mathbf{y} - f(\theta_0)). \tag{35}$$

Appendix C. Revisiting standard and NTK parametrizations, and convergence at initialization

In the following, we revisit the standard and the NTK parametrization. First, we repeat the result about the convergence of the NTK for the NTK parametrization at initialization. Then, we formally state how the NTK and standard parametrization are related, which makes it possible to prove the results for standard parametrization by using the results for NTK parametrization. Finally, we argue that using the same learning rate for every parameter under standard parametrization leads to redundancies in the NTK for the first layer and the biases.

C.1. Convergence of NTK under NTK parametrization at initialization

Here, we restate the following theorem from Yang (2020) about the convergence of the NTK at initialization. This was first shown in Jacot et al. (2018) when taking the limit of layer widths sequentially.

Theorem 4 *Consider a standard feedforward neural network in NTK parametrization. Then, the empirical NTK $\hat{\mathbf{\Theta}}_{\mathbf{x}', \mathbf{x}}$ converges to a deterministic matrix $\mathbf{\Theta}_{\mathbf{x}', \mathbf{x}}$, which we call the analytical NTK:*

$$\hat{\mathbf{\Theta}}_{x', x} = J(x', \theta_0) J(x, \theta_0)^\top = \sum_{l=1}^{L+1} \left(J(x', W^l) J(x, W^l)^\top + J(x', b^l) J(x, b^l)^\top\right) \xrightarrow{p} \mathbf{\Theta}_{x', x}, \tag{36}$$

for layer width $n \to \infty$.

Define $\mathbf{\Theta} := \mathbf{\Theta}_{\mathbf{x},\mathbf{x}} \in \mathbb{R}^{N\times N}$ as the analytical NTK on the training points. We will assume $\lambda_{\min}(\mathbf{\Theta}) > 0$. A sufficient condition for this is that $\|\mathbf{x}_i\|_2 = 1\ \forall i$, and that ϕ grows non-polynomially for large x, see Jacot et al. (2018). This directly implies that for n large enough, the minimum eigenvalue of the analytical NTK is lower bounded by a positive number with high probability:

Lemma 5 *For any $\delta_0 > 0$, there is n large enough, such that with probability of at least $1-\delta_0$, for the minimum eigenvalue of the empirical NTK,*

$$\lambda_{\min}\left(J(\theta_0)J(\theta_0)^\top\right) \geq \frac{1}{2}\lambda_{\min}(\mathbf{\Theta}),\ \textit{and}\ \lambda_{\max}\left(J(\theta_0)J(\theta_0)^\top\right) \leq 2\lambda_{\max}(\mathbf{\Theta}). \tag{37}$$

C.2. Equivalence of NTK parametrization to standard parametrization with layer-dependent learning rates

In this section, we will formally show how the NTK parametrization relates to the standard parametrization of neural networks. This makes it possible to prove results for standard parametrization by using the results for NTK parametrization.

Recall that the number of parameters is $p = \sum_{l=1}^{L+1}(n_{l-1}+1)n_l$. Define the diagonal matrix $H \in \mathbb{R}^{p\times p}$ through

$$H := \operatorname{diag}(H_{w,1}, H_{b,1}, \ldots, H_{w,L+1}, H_{b,L+1}), \tag{38}$$

where $H_{w,l} := \frac{1}{n_{l-1}} I_{n_{l-1}n_l}$, and $H_{b,l} := I_{n_l}$. The diagonal of $H^{\frac{1}{2}}$ contains the scalars by which each parameter is multiplied when going from NTK parametrization to standard parametrization. For θ_0^{std} initialized in standard parametrization, define

$$\theta_0^{\text{ntk}} := H^{-\frac{1}{2}}\theta_0^{\text{std}}. \tag{39}$$

Then, θ_0^{ntk} is initialized as in NTK parametrization. Further, let f^{std} denote a neural network in standard parametrization. Then,

$$f^{\text{ntk}}(x, \theta^{\text{ntk}}) := f^{\text{std}}(x, H^{\frac{1}{2}}\theta^{\text{ntk}}), \tag{40}$$

defines a neural network in NTK parametrization. Differentiating with respect to θ^{ntk} gives

$$J^{\text{ntk}}(x, \theta^{\text{ntk}}) = J^{\text{std}}(x, H^{\frac{1}{2}}\theta^{\text{ntk}})H^{\frac{1}{2}}, \quad \hat{\mathbf{\Theta}}^{\text{ntk}}_{x',x} = J^{\text{std}}(x', H^{\frac{1}{2}}\theta^{\text{ntk}})HJ^{\text{std}}(x, H^{\frac{1}{2}}\theta^{\text{ntk}}). \tag{41}$$

Motivated by this, define the regularized loss

$$\mathcal{L}^{\beta,\text{std}}(\theta) := \frac{1}{2}\sum_{i=1}^{N}(f^{\text{std}}(\mathbf{x}_i, \theta) - \mathbf{y}_i)^2 + \frac{1}{2}\beta(\theta-\theta_0)^\top H^{-1}(\theta-\theta_0), \tag{42}$$

with the gradient

$$\nabla_\theta \mathcal{L}^{\beta,\text{std}}(\theta) = J^{\text{std}}(\theta)^\top g^{\text{std}}(\theta) + \beta H^{-1}(\theta-\theta_0). \tag{43}$$

Define θ_t^{std} as parameters evolving by gradient flow of the regularized objective in standard parametrization[6], with layer-dependent learning rate $\eta_0 H$:

$$\frac{d\theta_t^{\text{std}}}{dt} = -\eta_0 H \nabla_\theta \mathcal{L}^{\beta,\text{std}}(\theta_t^{\text{std}}) = -\eta_0 H J^{\text{std}}(\theta_t^{\text{std}})^\top g^{\text{std}}(\theta_t^{\text{std}}) - \eta_0 \beta(\theta_t^{\text{std}} - \theta_0^{\text{std}}). \tag{44}$$

We define $\theta_t^{\text{ntk}} := H^{-\frac{1}{2}} \theta_t^{\text{std}}$. Then, as $\frac{d\theta_t^{\text{ntk}}}{dt} = H^{-\frac{1}{2}} \frac{d\theta_t^{\text{std}}}{dt}$,

$$\begin{aligned}
\frac{d\theta_t^{\text{ntk}}}{dt} &= -\eta_0 H^{\frac{1}{2}} J^{\text{std}}(\theta_t^{\text{std}})^\top g^{\text{std}}(\theta_t^{\text{std}}) - \eta_0 H^{-\frac{1}{2}} \beta(\theta_t^{\text{std}} - \theta_0^{\text{std}}) && (45)\\
&= -\eta_0 \left(J^{\text{std}}(H^{\frac{1}{2}} \theta_t^{\text{ntk}}) H^{\frac{1}{2}} \right)^\top g^{\text{std}}(H^{\frac{1}{2}} \theta_t^{\text{ntk}}) - \eta_0 \beta(\theta_t^{\text{ntk}} - \theta_0^{\text{ntk}}) && (46)\\
&= -\eta_0 J^{\text{ntk}}(\theta_t^{\text{ntk}})^\top g^{\text{ntk}}(\theta_t^{\text{ntk}}) - \eta_0 \beta(\theta_t^{\text{ntk}} - \theta_0^{\text{ntk}}). && (47)
\end{aligned}$$

Thus, θ_t^{ntk} follows the regularized gradient flow of the objective under NTK parametrization with learning rate η_0. Now, we can apply our results for NTK parametrization from above, and transfer them to standard parametrization by using $\theta_t^{\text{std}} = H^{\frac{1}{2}} \theta_t^{\text{ntk}}$, $f^{\text{std}}(x, \theta_t^{\text{std}}) = f^{\text{ntk}}(x, H^{-\frac{1}{2}} \theta_t^{\text{std}})$.

C.3. Redundancies when using the same learning rate for standard parametrization

In the previous section, we established the equivalence between training a neural network in NTK parametrization with learning rate η_0, and a neural network in standard parametrization with layer-dependent learning rate $\eta_0 H$. By definition, the learning rate for the first layer is $\frac{1}{n_0}\eta_0 = \frac{1}{d}\eta_0$, and the one for the biases is η_0. The learning rate for the other weight matrices is $\frac{1}{n_{l-1}}\eta_0 = \frac{1}{n}\eta_0$, for $l = 2, \ldots, L+1$. Note that the convergence of the learning rates to 0 for $n \to \infty$ is necessary to stabilize the gradient.

The learning rate that was used in the proof of Lee et al. (2019) is $\frac{1}{n}\eta_0$ for any layer. In the following, we argue that this effectively leads to the first layer, and the biases not being trained in the infinite-width limit. For simplicity, let $\beta = 0$. Lee et al. (2019) shows that using the learning rate $\frac{1}{n}\eta_0$ for each layer in standard parametrization, leads to the trained network for large width being driven by the standard parametrization NTK

$$\frac{1}{n} J^{\text{std}}(x', \theta_0) J^{\text{std}}(x, \theta_0)^\top = \frac{1}{n} \sum_{l=1}^{L+1} \left(J^{\text{std}}(x', W_0^l) J^{\text{std}}(x, W_0^l)^\top + J^{\text{std}}(x', b_0^l) J^{\text{std}}(x, b_0^l)^\top \right). \tag{48}$$

By using the equivalences from the previous section, we may write for $l = 2, \ldots, L+1$ (using $H_{w,l} = \frac{1}{n} I_{n_{l-1} n_l}$):

$$\begin{aligned}
\frac{1}{n} J^{\text{std}}(x', W_0^l) J^{\text{std}}(x, W_0^l)^\top &= \left(J^{\text{std}}(H_{w,l}^{\frac{1}{2}} \sqrt{n} W_0^l) H_{w,l}^{\frac{1}{2}} \right) \left(J^{\text{std}}(H_{w,l}^{\frac{1}{2}} \sqrt{n} W_0^l) H_{w,l}^{\frac{1}{2}} \right)^\top && (49)\\
&= J^{\text{ntk}}(\sqrt{n} W_0^l) J^{\text{ntk}}(\sqrt{n} W_0^l)^\top. && (50)
\end{aligned}$$

6. The existence of a unique solution of this ODE will follow from the relation to the gradient flow under NTK parametrization.

This is equal to the empirical NTK under NTK parametrization for weights $\sqrt{n}W^l_{0,i,j} \sim \mathcal{N}(0, \sigma_{w,l})$ in the l-th layer. However, for the first layer, we get (using $H_{w,1} = \frac{1}{d} I_{dn}$)

$$\frac{1}{n} J^{\text{std}}(x', W^1_0) J^{\text{std}}(x, W^1_0)^\top = \frac{d}{n} \left(J^{\text{std}}(H^{\frac{1}{2}}_{w,1} \sqrt{d} W^1_0) H^{\frac{1}{2}}_{w,1} \right) \left(J^{\text{std}}(H^{\frac{1}{2}}_{w,1} \sqrt{d} W^1_0) H^{\frac{1}{2}}_{w,1} \right)^\top \tag{51}$$

$$= \frac{d}{n} J^{\text{ntk}}(\sqrt{d} W^1_0) J^{\text{ntk}}(\sqrt{d} W^1_0)^\top \tag{52}$$

$$\to 0, \text{ for } n \to \infty, \tag{53}$$

as $J^{\text{ntk}}(\sqrt{d} W^1_0) J^{\text{ntk}}(\sqrt{d} W^1_0)^\top$ converges by Theorem 4. Similarly for the biases, for $l = 1, \ldots, L+1$:

$$\frac{1}{n} J^{\text{std}}(x', b^l_0) J^{\text{std}}(x, b^l_0)^\top = \frac{1}{n} J^{\text{ntk}}(x', b^l_0) J^{\text{ntk}}(x, b^l_0)^\top \to 0, \text{ for } n \to \infty. \tag{54}$$

Thus, the analytical standard parametrization NTK of Lee et al. (2019) does not depend on the contribution of the gradient with respect to the first layer and the biases. In other words, using the learning rate $\frac{1}{n}\eta_0$ for the first layer and the biases leads to them not being trained for large widths.

Instead, one may scale the learning rates "correctly", as motivated by the NTK parametrization in the previous section. For large widths n, the trained network is then governed by the following modified NTK for standard parametrization:

$$J^{\text{std}}(\theta) H J^{\text{std}}(\theta) \tag{55}$$

$$= J^{\text{std}}(W^1) J^{\text{std}}(W^1)^\top + J^{\text{std}}(b^1) J^{\text{std}}(b^1)^\top + \sum_{l=2}^{L+1} \left(\frac{1}{n} J^{\text{std}}(W^l) J^{\text{std}}(W^l)^\top + J^{\text{std}}(b^l) J^{\text{std}}(b^l)^\top \right). \tag{56}$$

For simplicity, we do not consider training of the first layer and the biases.

Appendix D. Local Lipschitzness and Boundedness of the Jacobian

The goal of this section is to prove the following lemma:

Lemma 6 *For any $\delta_0 > 0$, there is $K' > 0$ (independent of C), such that: For every $C > 0$, there is n large enough, such that with probability of at least $1 - \delta_0$ (over random initialization): For any point x with $\|x\|_2 \leq 1$:*

$$\forall \theta \in B(\theta_0, C) : \|J(x, \theta)\|_2 \leq K', \tag{57}$$

$$\forall \theta, \tilde{\theta} \in B(\theta_0, C) : \|J(x, \theta) - J(x, \tilde{\theta})\|_2 \leq \frac{1}{\sqrt{n}} K' \|\theta - \tilde{\theta}\|_2. \tag{58}$$

In particular, with $K = \sqrt{N} K'$, for the Jacobian over the training points:

$$\forall \theta \in B(\theta_0, C) : \|J(\theta)\|_F \leq K, \tag{59}$$

$$\forall \theta, \tilde{\theta} \in B(\theta_0, C) : \|J(\theta) - J(\tilde{\theta})\|_F \leq \frac{1}{\sqrt{n}} K \|\theta - \tilde{\theta}\|_2. \tag{60}$$

As a direct consequence, for the Hessian $\nabla_\theta^2 f(x,\theta) \in \mathbb{R}^{p\times p}$ *of the network,*

$$\forall \theta \in B(\theta_0, C) : \left\|\nabla_\theta^2 f(x,\theta)\right\|_2 \leq \frac{1}{\sqrt{n}} K'. \tag{61}$$

Here, the Frobenius-norm is used to aggregate over different training points, i.e. $\|J(\theta)\|_F^2 = \sum_{i=1}^N \|J(\mathbf{x}_i,\theta)\|_2^2$.

Lee et al. (2019) proved this Lemma for standard parametrization. In their version for NTK parametrization (Lemma 2), there is a typo stating that the Lipschitz-constant of the Jacobian is $O(1)$ instead of $O(\frac{1}{\sqrt{n}})$, which is why we quickly go over how to obtain this Lemma. As mentioned in the previous section, we don't train the first layer and the biases for simplicity. Let f^{ntk} be the network in NTK parametrization, and let θ_0 be randomly initialized according to the NTK parametrization. Further, let $\theta, \tilde{\theta} \in B(\theta_0, C)$. Then, $f^{\text{std}}(\theta) := f^{\text{ntk}}(\sqrt{n}\theta)$ is a network in standard parametrization, and $\frac{1}{\sqrt{n}}\theta_0$ is randomly initialized in standard parametrization. Further, $\frac{1}{\sqrt{n}}\theta, \frac{1}{\sqrt{n}}\tilde{\theta} \in B(\frac{1}{\sqrt{n}}\theta_0, \frac{1}{\sqrt{n}}C)$. Note, that $J^{\text{ntk}}(\theta) = \frac{1}{\sqrt{n}} J^{\text{std}}(\frac{1}{\sqrt{n}}\theta)$. Now, by applying Lemma 1 from Lee et al. (2019) to the network in standard parametrization and the parameters $\frac{1}{\sqrt{n}}\theta_0, \frac{1}{\sqrt{n}}\theta, \frac{1}{\sqrt{n}}\tilde{\theta}$, we get with high probability over random initialization:

$$\|J^{\text{ntk}}(x,\theta)\|_2 = \frac{1}{\sqrt{n}}\|J^{\text{std}}(x, \frac{1}{\sqrt{n}}\theta)\|_2 \leq \frac{1}{\sqrt{n}}\sqrt{n}K' = K', \tag{62}$$

and

$$\|J^{\text{ntk}}(x,\theta) - J^{\text{ntk}}(x,\theta)\|_2 = \frac{1}{\sqrt{n}}\|J^{\text{std}}(x, \frac{1}{\sqrt{n}}\theta) - J^{\text{std}}(x, \frac{1}{\sqrt{n}}\tilde{\theta})\|_2 \tag{63}$$

$$\leq \frac{1}{\sqrt{n}}\sqrt{n}K'\|\frac{1}{\sqrt{n}}\theta - \frac{1}{\sqrt{n}}\tilde{\theta}\|_2 = \frac{1}{\sqrt{n}}K'\|\theta - \tilde{\theta}\|_2. \tag{64}$$

Appendix E. Proof for regularized gradient flow

The goal of this section is to proof Theorem 2. In section E.1 we show that the norm of the gradient of the regularized loss decays exponentially over time. This result directly implies that the distance between the parameters and their initialization is bounded by a constant, i.e. $\|\theta_t - \theta_0\|_2 = O(1)$. Further, it follows that the Jacobian remains close to its initial value, with deviations scaling as $O(\frac{1}{\sqrt{n}})$. Using this, we can prove the closeness of the network to it's linearization during training in section E.2.

E.1. Exponential decay of the regularized gradient and closeness of parameters to their initial value

We start with the following technical lemma.

Lemma 7 *Let* $\beta \geq 0$. *We have for any* $t \geq 0$*:* $\|g(\theta_t)\|_2 \leq \|g(\theta_0)\|_2$. *Further, for any* $\delta_0 > 0$*, there is* $R_0 > 0$*, such that for* n *large enough, with probability of at least* $1 - \delta_0$ *over random initialization,* $\|g(\theta_0)\|_2 \leq R_0$.

Proof Using the chain rule and the definition of the gradient flow, we obtain

$$\frac{d}{dt}\mathcal{L}^{\beta}(\theta_t) = \nabla_\theta \mathcal{L}^{\beta}(\theta_t)^\top \frac{d\theta_t}{dt} = -\eta_0 \nabla_\theta \mathcal{L}^{\beta}(\theta_t)^\top \nabla_\theta \mathcal{L}^{\beta}(\theta_t) = -\eta_0 \|\nabla_\theta \mathcal{L}^{\beta}(\theta_t)\|_2^2 \leq 0. \tag{65}$$

Thus,

$$\frac{1}{2}\|g(\theta_t)\|_2^2 \leq \frac{1}{2}\|g(\theta_t)\|_2^2 + \frac{1}{2}\|\theta_t - \theta_0\|_2^2 = \mathcal{L}^{\beta}(\theta_t) \leq \mathcal{L}^{\beta}(\theta_0) = \frac{1}{2}\|g(\theta_0)\|_2^2. \tag{66}$$

Hence, $\|g(\theta_t)\|_2 \leq \|g(\theta_0)\|_2$. Further, note that $f(\theta_0)$ converges in distribution to a Gaussian with mean zero and covariance given by the NNGP kernel Lee et al. (2018). Thus, for n large enough, one can bound $\|g(\theta_0)\|_2$ with high probability. ■

In the following, we show that the norm of the gradient of the regularized loss decays exponentially over time.

Theorem 8 *Let $\beta \geq 0$. Let $\delta_0 > 0$ be arbitrarily small. There are $K', K, R_0, c_\beta > 0$, such that for n large enough, the following holds with probability of at least $1 - \delta_0$ over random initialization, when applying regularized gradient flow with learning rate $\eta = \eta_0$:*

$$\left\| \frac{d\theta_t}{dt} \right\|_2 = \eta_0 \|\nabla_\theta \mathcal{L}^{\beta}(\theta_t)\|_2 \leq \eta_0 K R_0 e^{-\eta_0 c_\beta t}, \tag{67}$$

$$\|\theta_t - \theta_0\|_2 \leq \frac{KR_0}{c_\beta}\left(1 - e^{-\eta_0 c_\beta t}\right) < \frac{KR_0}{c_\beta} =: C, \tag{68}$$

$$\forall \|x\|_2 \leq 1 : \|J(x, \theta_t) - J(x, \theta_0)\|_2 \leq \frac{1}{\sqrt{n}} K' C, \tag{69}$$

$$\|J(\theta_t) - J(\theta_0)\|_2 \leq \frac{1}{\sqrt{n}} KC. \tag{70}$$

Proof Using Lemma 7, there is $R_0 > 0$, such that for n large enough, with probability of at least $1 - \frac{1}{3}\delta_0$ over random initialization, $\|g(\theta_0)\|_2 \leq R_0$. Further, using Lemma 6, let K be the constant for local Lipschitzness/boundedness of the Jacobian with probability $1 - \frac{1}{3}\delta_0$ for n large enough. Finally, by Lemma 5, for n large enough, with probability of at least $1 - \frac{1}{3}\delta_0$ over random initialization, the minimum eigenvalue of the empirical NTK is lower bounded: $\lambda_{\min}(J(\theta_0)J(\theta_0)^\top) \geq \frac{1}{2}\lambda_{\min}(\mathbf{\Theta})$.[7] For n large enough, these three events hold with probability of at least $1 - \delta_0$ over random initialization. In the following, we consider such initializations θ_0.

Define $c_\beta := \frac{1}{2}\beta$ for $\beta > 0$, and $c_\beta := \frac{1}{3}\lambda_{\min}(\mathbf{\Theta})$ for $\beta = 0$.[8] Let $C := \frac{KR_0}{c_\beta}$. By Lemma 6, the gradient flow ODE has a unique solution as long as $\theta_t \in B(\theta_0, C)$. Consider

$$t_1 := \inf\{t \geq 0 : \|\theta_t - \theta_0\|_2 \geq C\}. \tag{71}$$

In the following, let $t \leq t_1$. Recall that

$$\frac{d\theta_t}{dt} = -\eta_0 \nabla_\theta \mathcal{L}^{\beta}(\theta_t) = -\eta_0 \left(J(\theta_t)^\top g(\theta_t) + \beta(\theta_T - \theta_0) \right). \tag{72}$$

7. We will only need this for $\beta = 0$.
8. One can choose any constant smaller than β for $\beta > 0$, and similarly any constant smaller than $\lambda_{\min}(\mathbf{\Theta})$ for $\beta = 0$.

We want to show that the gradient $\nabla_\theta \mathcal{L}^\beta(\theta_t)$ of the regularized loss converges to 0 quickly, and hence θ_t doesn't move much. For $\beta = 0$, its norm is $\|J(\theta_t)^\top g(\theta_t)\|_2$ and hence Lee et al. (2019) and related proofs showed that $\|g(\theta_t)\|_2$ converges to 0 quickly. However, for $\beta > 0$, this is not the case, as the training error does not converge to 0. Instead, we directly look at the dynamics of the norm of the gradient:

$$\frac{d}{dt}\|\nabla_\theta \mathcal{L}^\beta(\theta_t)\|_2^2 = 2\left(\nabla_\theta \mathcal{L}^\beta(\theta_t)\right)^\top \nabla_\theta^2 \mathcal{L}^\beta(\theta_t)\frac{d\theta_t}{dt} \tag{73}$$

$$= -2\eta_0 \left(\nabla_\theta \mathcal{L}^\beta(\theta_t)\right)^\top \nabla_\theta^2 \mathcal{L}^\beta(\theta_t)\left(\nabla_\theta \mathcal{L}^\beta(\theta_t)\right). \tag{74}$$

The Hessian $\nabla_\theta^2 \mathcal{L}^\beta(\theta_t) \in \mathbb{R}^{p\times p}$ of the regularized loss is given by

$$\nabla_\theta^2 \mathcal{L}^\beta(\theta_t) = g(\theta_t)^\top \nabla_\theta^2 f(\theta_t) + J(\theta)^\top J(\theta_t) + \beta I_p. \tag{75}$$

Here, $\nabla_\theta^2 f(\theta) \in \mathbb{R}^{N\times p\times p}$, and $g(\theta_t)^\top \nabla_\theta^2 f(\theta_t) = \sum_{i=1}^N g(\mathbf{x}_i, \theta_t)\nabla_\theta^2 f(\mathbf{x}_i, \theta_t)$. Next, use the triangle inequality and Cauchy-Schwarz to write

$$\|g(\theta_t)^\top \nabla_\theta^2 f(\theta)\|_2 \le \sum_{i=1}^N |g(\mathbf{x}_i, \theta_t)| \|\nabla_\theta^2 f(\mathbf{x}_i, \theta_t)\|_2 \le \|g(\theta_t)\|_2 \sqrt{\sum_{i=1}^N \|\nabla_\theta^2 f(\mathbf{x}_i, \theta_t)\|_2^2}. \tag{76}$$

By Lemma 6, we have $\|\nabla_\theta^2 f(\mathbf{x}_i, \theta_t)\|_2 \le \frac{1}{\sqrt{n}} K'$. Further, by Lemma 7, we have $\|g(\theta_t)\|_2 \le \|g(\theta_0)\|_2 \le R_0$. Thus (with $K = \sqrt{N}K'$),

$$\|g(\theta_t)^\top \nabla_\theta^2 f(\theta)\|_2 \le \frac{1}{\sqrt{n}} K R_0. \tag{77}$$

As $g(\theta_t)^\top \nabla_\theta^2 f(\theta)$ is symmetric, its minimum eigenvalue is lower bounded:

$$\lambda_{\min}\left(g(\theta_t)^\top \nabla_\theta^2 f(\theta)\right) \ge -\|g(\theta_t)^\top \nabla_\theta^2 f(\theta)\|_2 \ge -\frac{1}{\sqrt{n}} K R_0. \tag{78}$$

Now consider $\beta > 0$. Then, for n large enough, the smallest eigenvalue of the Hessian of the regularized loss is positive:

$$\lambda_{\min}\left(\nabla_\theta \mathcal{L}^\beta(\theta_t)\right) \ge -\frac{1}{\sqrt{n}} K R_0 + 0 + \beta \ge \frac{\beta}{2} =: c_\beta. \tag{79}$$

Thus,

$$\frac{d}{dt}\|\nabla_\theta \mathcal{L}^\beta(\theta_t)\|_2^2 \le -2\eta_0 c_\beta \|\nabla_\theta \mathcal{L}^\beta(\theta_t)\|_2^2. \tag{80}$$

In Remark 9 we show how to modify the proof so that this step is valid for $\beta = 0$.

By Gronwalls inequality, it follows (for $\beta \ge 0$) that

$$\|\nabla_\theta \mathcal{L}^\beta(\theta_t)\|_2^2 \le e^{-2\eta_0 c_\beta t} \|\nabla_\theta \mathcal{L}^\beta(\theta_0)\|_2^2. \tag{81}$$

Thus,

$$\|\nabla_\theta \mathcal{L}^\beta(\theta_t)\|_2 \leq e^{-\eta_0 c_\beta t}\|\nabla_\theta \mathcal{L}^\beta(\theta_0)\|_2 \tag{82}$$
$$\leq e^{-\eta_0 c_\beta t}\|J(\theta_0)^\top g(\theta_0)\|_2 \tag{83}$$
$$\leq e^{-\eta_0 c_\beta t}\|J(\theta_0)\|_2\|g(\theta_0)\|_2 \tag{84}$$
$$\leq KR_0 e^{-\eta_0 c_\beta t}. \tag{85}$$

Hence, for the distance of parameters from initialization

$$\|\theta_t - \theta_0\|_2 = \left\|\int_0^t \frac{d\theta_u}{du} du\right\|_2 \tag{86}$$
$$\leq \int_0^t \left\|\frac{d\theta_u}{du}\right\|_2 du \tag{87}$$
$$\leq \eta_0 KR_0 \int_0^t e^{-\eta_0 c_\beta u} du \tag{88}$$
$$= \frac{KR_0}{c_\beta}(1 - e^{-\eta_0 c_\beta t}). \tag{89}$$

Thus, for $t \leq t_1$, $\|\theta_t - \theta_0\|_2 < C$. By continuity, $t_1 = \infty$. Using local Lipschitzness, one can further bound the distance of the Jacobian at any $\|x\|_2 \leq 1$

$$\|J(x, \theta_t) - J(x, \theta_0)\|_2 \leq \frac{1}{\sqrt{n}} K' \|\theta_t - \theta_0\|_2 \leq \frac{1}{\sqrt{n}} K' C. \tag{90}$$

This finishes the proof of Theorem 8.

■

Remark 9 *For $\beta = 0$, the Hessian is*

$$\nabla^2_\theta \mathcal{L}^0(\theta_t) = g(\theta_t)^\top \nabla^2_\theta f(\theta_t) + J(\theta_t)^\top J(\theta_t). \tag{91}$$

For $\beta > 0$, we just used that $J(\theta_t)^\top J(\theta_t)$ is positive semi-definite, as βI dominates the negative eigenvalues of the first term of the Hessian. For $\beta = 0$, this is not enough. $J(\theta_t)^\top J(\theta_t) \in \mathbb{R}^{p\times p}$ shouldn't be confused with the NTK $J(\theta_t)J(\theta_t)^\top \in \mathbb{R}^{N\times N}$. However, they share the same nonzero eigenvalues. For $p > N$ (which is the case for n large enough), $J(\theta_t)^\top J(\theta_t)$ will additionally have the eigenvalue 0 with multiplicity of at least $p - N$. Thus, we can't naively lower bound the minimum eigenvalue of the Hessian with the minimum eigenvalue of $J(\theta_t)^\top J(\theta_t)$.

Luckily, $\nabla_\theta \mathcal{L}^0(\theta_t) = J(\theta_t)^\top g(\theta_t)$ is in the row-span of $J(\theta_t)$. This is orthogonal to the nullspace of $J(\theta_t)$, i.e. the eigenspace corresponding to the eigenvalue 0 of $J(\theta_t)$. Thus, $\nabla_\theta \mathcal{L}^0(\theta_t)$ only "uses" the positive eigenvalues of $J(\theta_t)^\top J(\theta_t)$. The smallest positive eigenvalue of $J(\theta_t)^\top J(\theta_t)$ is equal to the smallest positive eigenvalue of the empirical NTK $J(\theta_t)J(\theta_t)^\top$, which is lower bounded by $\frac{1}{2}\lambda_{\min}(\mathbf{\Theta})$ on the high probability event we consider.

Hence, for n large enough,

$$\frac{d}{dt}\|\nabla_\theta \mathcal{L}^0(\theta_t)\|_2^2 = -3\eta_0 \left(\nabla_\theta \mathcal{L}^0(\theta_t)\right)^\top \nabla_\theta^2 \mathcal{L}^0(\theta_t) \left(\nabla_\theta \mathcal{L}^0(\theta_t)\right) \tag{92}$$

$$\leq -2\eta_0 \lambda_{\min}\left(\nabla_\theta \mathcal{L}^0(\theta_t)\right) \|\nabla_\theta \mathcal{L}^0(\theta_t)\|_2^2 - \eta_0 \frac{1}{2}\lambda_{\min}(\mathbf{\Theta})\|\nabla_\theta \mathcal{L}^0(\theta_t)\|_2^2 \tag{93}$$

$$\leq -2\eta_0 \left(-\frac{\log n}{\sqrt{n}} K R_0 + \frac{1}{2}\lambda_{\min}(\mathbf{\Theta})\right) \|\nabla_\theta \mathcal{L}^0(\theta_t)\|_2^2 \tag{94}$$

$$\leq -2\eta_0 \frac{1}{3}\lambda_{\min}(\mathbf{\Theta})\|\nabla_\theta \mathcal{L}^0(\theta_t)\|_2^2. \tag{95}$$

Define $c_0 := \frac{1}{3}\lambda_{\min}(\mathbf{\Theta})$ to continue with the proof above for $\beta \geq 0$. This is an alternative proof to Lee et al. (2019). It shows that in the unregularized case, it is important that the gradient flow lies in the row-space of the Jacobian.

E.2. Closeness to the Linearized Network along the Regularized Gradient Flow

Now, we can prove that the neural network along the regularized gradient flow stays close to the linearized network along the linear regularized gradient flow. We restate the theorem for convenience.

Theorem 2 *Let $\beta \geq 0$. Let $\delta_0 > 0$ be arbitrarily small. Then, there are C_1, C_2, such that for n large enough, with probability of at least $1 - \delta_0$ over random initialization,*

$$\sup_{t\geq 0}\|\theta_t - \theta_t^{\text{lin}}\|_2 \leq C_1 \frac{1}{\sqrt{n}}, \tag{10}$$

$$\forall \|x\|_2 \leq 1 : \sup_{t\geq 0}\|f(x,\theta_t) - f_{\theta_0}^{\text{lin}}(x,\theta_t^{\text{lin}})\|_2 \leq C_2 \frac{1}{\sqrt{n}}. \tag{11}$$

Proof The proof of Lee et al. (2019) use that the training error converges to 0. Thus, we need a different approach, which also provides a more straightforward and intuitive proof for $\beta = 0$. Recall that

$$f^{\text{lin}}(x,\theta) = f(x,\theta_0) + J(x,\theta_0)(\theta - \theta_0), \tag{96}$$

and

$$\frac{d\theta^{\text{lin}}}{dt} = -\eta_0 \left(J(\theta_0)^\top g^{\text{lin}}(\theta_t^{\text{lin}}) + \beta(\theta_t^{\text{lin}} - \theta_0)\right). \tag{97}$$

To prove the second part of the theorem, we will use

$$\|f(x,\theta_t) - f^{\text{lin}}(x,\theta_t^{\text{lin}})\|_2 \leq \|f(x,\theta_t) - f^{\text{lin}}(x,\theta_t)\|_2 + \|f^{\text{lin}}(x,\theta_t) - f^{\text{lin}}(x,\theta_t^{\text{lin}})\|_2. \tag{98}$$

We start by bounding the first term. Next, we bound $\|\theta_t - \theta_t^{\text{lin}}\|_2$, and use this to bound the second term.

First step: To bound $\|f(x,\theta_t) - f^{\text{lin}}(x,\theta_t)\|_2$, we compute

$$\left\| \frac{d}{dt}\left(f(x,\theta_t) - f^{\text{lin}}(x,\theta_t)\right)\right\|_2 = \left\|(J(x,\theta_t) - J(x,\theta_0))\,\frac{d\theta_t}{dt}\right\|_2 \tag{99}$$

$$\leq \|J(x,\theta_t) - J(x,\theta_0)\|_2 \left\|\frac{d\theta_t}{dt}\right\|_2 \tag{100}$$

$$\leq \frac{1}{\sqrt{n}} K' C \eta_0 K R_0 e^{-\eta_0 c_\beta t}, \tag{101}$$

where we used Theorem 8 in the last step. Now, we can bound

$$\|f(x,\theta_t) - f^{\text{lin}}(x,\theta_t)\|_2 \leq \frac{1}{\sqrt{n}} K' C \eta_0 K R_0 \int_0^t e^{-\eta_0 c_\beta u} du \leq \frac{1}{\sqrt{n}} K' C \frac{K R_0}{c_\beta} = \frac{1}{\sqrt{n}} K' C^2. \tag{102}$$

In particular, for the difference at the training points, $\|f(\theta_t) - f^{\text{lin}}(\theta_t)\|_2 \leq \frac{1}{\sqrt{n}} K C^2$.

Second step: Now, we bound the difference between $\theta_t - \theta_t^{\text{lin}}$. We write

$$\frac{d\theta_t}{dt} = -\eta_0 \left(J(\theta_t)^\top g(\theta_t) + \beta(\theta_t - \theta_0)\right) \tag{103}$$

$$= -\eta_0 \left(\left(J(\theta_t) - J(\theta_0)\right)^\top g(\theta_t) + J(\theta_0)^\top \left(g(\theta_t) - g^{\text{lin}}(\theta_t)\right) + J(\theta_0)^\top g^{\text{lin}}(\theta_t) + \beta(\theta_t - \theta_0)\right) \tag{104}$$

$$= -\eta_0 \Delta_t - \eta_0 \left(J(\theta_0)^\top g^{\text{lin}}(\theta_t) + \beta(\theta_t - \theta_0)\right), \tag{105}$$

where $\Delta_t := \left(J(\theta_t) - J(\theta_0)\right)^\top g(\theta_t) + J(\theta_0)^\top \left(g(\theta_t) - g^{\text{lin}}(\theta_t)\right)$. We now bound $\|\Delta_t\|_2$. For the first term, use Theorem 8:

$$\|\left(J(\theta_t) - J(\theta_0)\right)^\top g(\theta_t)\|_2 \leq \|J(\theta_t) - J(\theta_0)\|_2 \|g(\theta_t)\|_2 \leq \frac{1}{\sqrt{n}} K C R_0. \tag{106}$$

For the second term, use Theorem 8 and the bound we derived in the first step to write

$$\|J(\theta_0)^\top \left(g(\theta_t) - g^{\text{lin}}(\theta_t)\right)\|_2 = \|J(\theta_0)^\top \left(f(\theta_t) - f^{\text{lin}}(\theta_t)\right)\|_2 \tag{107}$$

$$\leq \|J(\theta_0)\|_2 \|f(\theta_t) - f^{\text{lin}}(\theta_t)\|_2 \tag{108}$$

$$\leq \frac{1}{\sqrt{n}} K^2 C^2. \tag{109}$$

Thus, defining $K^\Delta := KCR_0 + K^2C^2$, we can bound $\|\Delta_t\|_2 \leq \frac{1}{\sqrt{n}} K^\Delta$. Now, compute

$$\frac{d}{dt}(\theta_t - \theta_t^{\text{lin}}) \tag{110}$$

$$= -\eta_0 \Delta_t - \eta_0 \left(J(\theta_0)^\top g^{\text{lin}}(\theta_t) + \beta_N(\theta_t - \theta_0)\right) + \eta_0 \left(J(\theta_0)^\top g^{\text{lin}}(\theta_t^{\text{lin}}) + \beta_N(\theta_t^{\text{lin}} - \theta_0)\right) \tag{111}$$

$$= -\eta_0 \Delta_t - \eta_0 \left(J(\theta_0)^\top \left(g^{\text{lin}}(\theta_t) - g^{\text{lin}}(\theta_t^{\text{lin}})\right) + \beta(\theta_t - \theta_t^{\text{lin}})\right) \tag{112}$$

$$= -\eta_0 \Delta_t - \eta_0 \left(J(\theta_0)^\top J(\theta_0)(\theta_t - \theta_t^{\text{lin}}) + \beta(\theta_t - \theta_t^{\text{lin}})\right) \tag{113}$$

$$= -\eta_0 \Delta_t - \eta_0 \left(J(\theta_0) J(\theta_0)^\top + \beta I\right)(\theta_t - \theta_t^{\text{lin}}). \tag{114}$$

The solution to this inhomogeneous linear ODE in $\theta_t - \theta_t^{\text{lin}}$ is

$$\theta_t - \theta_t^{\text{lin}} = \int_0^t e^{-\eta_0\left(J(\theta_0)J(\theta_0)^\top + \beta I\right)(t-u)}(-\eta_0\Delta_u)du. \tag{115}$$

Hence (using $\|e^{-A}\|_2 \leq e^{-\lambda_{\min}(A)}$),

$$\|\theta_t - \theta_t^{\text{lin}}\|_2 \leq \int_0^t \|e^{-\eta_0\left(J(\theta_0)J(\theta_0)^\top + \beta I\right)(t-u)}\|_2 \eta_0 \|\Delta_u\|_2 du \tag{116}$$

$$\leq \int_0^t e^{-\eta_0(c_0+\beta)(t-u)}\eta_0 \frac{1}{\sqrt{n}} K^\Delta du \tag{117}$$

$$\leq \frac{1}{\sqrt{n}} \frac{K^\Delta}{c_0+\beta}. \tag{118}$$

Thus, $\sup_t \|\theta_t - \theta_t^{\text{lin}}\|_2 \leq \frac{K^\Delta}{c_0+\beta}\frac{1}{\sqrt{n}}$.

Third step: Using the bound on $\|\theta_t - \theta_t^{\text{lin}}\|_2$, we can easily bound $\|f^{\text{lin}}(x,\theta_t) - f^{\text{lin}}(x,\theta_t^{\text{lin}})\|_2$:

$$\|f^{\text{lin}}(x,\theta_t) - f^{\text{lin}}(x,\theta_t^{\text{lin}})\|_2 = \|J(x,\theta_0)(\theta_t - \theta_t^{\text{lin}})\|_2 \tag{119}$$

$$\leq \|J(x,\theta_0)\|_2 \|\theta_t - \theta_t^{\text{lin}}\|_2 \tag{120}$$

$$\leq K' \frac{K^\Delta}{c_0+\beta}\frac{1}{\sqrt{n}}. \tag{121}$$

Finally, use equation (98) to write

$$\|f(x,\theta_t) - f^{\text{lin}}(x,\theta_t^{\text{lin}})\|_2 \leq \left(K'C^2 + K'\frac{K^\Delta}{c_0+\beta}\right)\frac{1}{\sqrt{n}}, \tag{122}$$

which concludes the proof.

■

Appendix F. Proof for regularized gradient descent

F.1. Geometric decay of the regularized gradient and closeness of parameters to their initial value

Theorem 10 *Let $\beta \geq 0$. Let $\delta_0 > 0$ be arbitrarily small. There are $K', K, R_0, c_\beta, \eta_{\max} > 0$, such that for n large enough, the following holds with probability of at least $1-\delta_0$ over random initialization, when applying regularized gradient descent with learning rate $\eta = \eta_0 \leq \eta_{\max}$:*

$$\|\theta_{t+1} - \theta_t\|_2 = \eta_0 \|\nabla_\theta \mathcal{L}^\beta(\theta_t)\|_2 \leq \eta_0 K R_0 \left(1 - \eta_0 c_\beta\right)^t, \tag{123}$$

$$\|\theta_t - \theta_0\|_2 < \frac{KR_0}{c_\beta} =: C, \tag{124}$$

$$\forall \|x\|_2 \leq 1 : \|J(x,\theta_t) - J(x,\theta_0)\|_2 \leq \frac{1}{\sqrt{n}} K'C, \tag{125}$$

$$\|J(\theta_t) - J(\theta_0)\|_2 \leq \frac{1}{\sqrt{n}} KC. \tag{126}$$

Proof Consider the same high probability event as in the proof for the regularized gradient flow. Define $c_\beta := \frac{1}{2}\beta$ for $\beta > 0$, and $c_\beta := \frac{1}{3}\lambda_{\min}(\mathbf{\Theta})$ for $\beta = 0$. Further, let $C := \frac{KR_0}{c_\beta}$.

We prove the first two inequalities by induction. For $t = 0$,

$$\|\nabla_\theta \mathcal{L}^\beta(\theta_t)\|_2 = \|J(\theta_0)^\top g(\theta_0)\|_2 \leq KR_0. \tag{127}$$

Now, assume it holds true for $s \leq t$. We want to bound $\|\theta_{t+1} - \theta_t\|_2 = \eta_0 \|\nabla_\theta \mathcal{L}^\beta(\theta_t)\|_2$. Recall that $\nabla_\theta \mathcal{L}^\beta(\theta_t) = J(\theta_t)^\top g(\theta_t) + \beta(\theta_t - \theta_0)$. Write

$$\begin{aligned}
\|\nabla_\theta \mathcal{L}^\beta(\theta_t)\|_2^2 =& \|\nabla_\theta \mathcal{L}^\beta(\theta_{t-1}) + \nabla_\theta \mathcal{L}^\beta(\theta_t) - \nabla_\theta \mathcal{L}^\beta(\theta_{t-1})\|_2^2 && (128)\\
=& \|\nabla_\theta \mathcal{L}^\beta(\theta_{t-1})\|_2^2 && (129)\\
&+ 2\nabla_\theta \mathcal{L}^\beta(\theta_{t-1})^\top \left(\nabla_\theta \mathcal{L}^\beta(\theta_t) - \nabla_\theta \mathcal{L}^\beta(\theta_{t-1})\right) && (130)\\
&+ \|\nabla_\theta \mathcal{L}^\beta(\theta_t) - \nabla_\theta \mathcal{L}^\beta(\theta_{t-1})\|_2^2. && (131)
\end{aligned}$$

In the following, we will look at how to bound the second and the third terms. We have

$$\begin{aligned}
\nabla_\theta \mathcal{L}^\beta(\theta_t) - \nabla_\theta \mathcal{L}^\beta(\theta_{t-1}) &= \nabla_\theta \mathcal{L}^\beta(\theta_{t-1} - \eta_0 \nabla_\theta \mathcal{L}^\beta(\theta_{t-1})) - \nabla_\theta \mathcal{L}^\beta(\theta_{t-1}) && (132)\\
&= -\int_0^{\eta_0} \left(\nabla_\theta^2 \mathcal{L}^\beta \left(\theta_{t-1} - u\nabla_\theta \mathcal{L}^\beta(\theta_{t-1})\right)\right) \cdot \nabla_\theta \mathcal{L}^\beta(\theta_{t-1}) du. && (133)
\end{aligned}$$

As in the proof for the gradient flow, the following part only holds for $\beta > 0$. Note that for any $u \in [0, \eta_0]$, $\theta_{t-1} - u\nabla_\theta \mathcal{L}^\beta(\theta_{t-1}) \in B(\theta_0, C)$, as we know by induction that $\theta_{t-1}, \theta_t \in B(\theta_0, C)$. Thus, similar to the proof for the gradient flow, for n large enough, $\forall u \in [0, \eta_0]$:

$$\lambda_{\min}\left(\nabla_\theta^2 \mathcal{L}^\beta \left(\theta_{t-1} - u\nabla_\theta \mathcal{L}^\beta(\theta_{t-1})\right)\right) \geq \frac{3}{2} c_\beta. \tag{134}$$

Note that we use $\frac{3}{2}c_\beta = \frac{3}{4}\beta$, which is slightly higher than c_β which we used in the gradient flow case, to arrive at the equivalent result in the end. For the second term (130) we get

$$\begin{aligned}
& 2\nabla_\theta \mathcal{L}^\beta(\theta_{t-1})^\top \left(\nabla_\theta \mathcal{L}^\beta(\theta_t) - \nabla_\theta \mathcal{L}^\beta(\theta_{t-1})\right) && (135)\\
=& -2\int_0^{\eta_0} \nabla_\theta \mathcal{L}^\beta(\theta_{t-1}) \left(\nabla_\theta^2 \mathcal{L}^\beta \left(\theta_{t-1} - u\nabla_\theta \mathcal{L}^\beta(\theta_{t-1})\right)\right) \cdot \nabla_\theta \mathcal{L}^\beta(\theta_{t-1}) du && (136)\\
\leq& -2\eta_0 \frac{3}{2} c_\beta \|\nabla_\theta \mathcal{L}^\beta(\theta_{t-1})\|_2^2. && (137)
\end{aligned}$$

Further, for any $\theta \in B(\theta_0, C)$ we have that

$$\begin{aligned}
\|\nabla_\theta^2 \mathcal{L}^\beta(\theta)\|_2 &\leq \|g(\theta)^\top \nabla_\theta^2 f(\theta)\|_2 + \|J(\theta)^\top J(\theta)\|_2 + \beta I_p && (138)\\
&\leq \frac{1}{\sqrt{n}} KR_0 + \lambda_{\max}\left(J(\theta)^\top J(\theta)\right) + \beta && (139)\\
&\leq \frac{1}{\sqrt{n}} KR_0 + 2\lambda_{\max}(\mathbf{\Theta}) + \beta && (140)\\
&\leq 2(\lambda_{\max}(\mathbf{\Theta}) + \beta), && (141)
\end{aligned}$$

for n large enough. Using this with $\theta = \theta_{t-1} - u\nabla_\theta \mathcal{L}^\beta(\theta_{t-1})$, we get for the third term (131),

$$\|\nabla_\theta \mathcal{L}^\beta(\theta_t) - \nabla_\theta \mathcal{L}^\beta(\theta_{t-1})\|_2^2 \tag{142}$$

$$= \left\| \int_0^{\eta_0} \left(\nabla_\theta^2 \mathcal{L}^\beta \left(\theta_{t-1} - u\nabla_\theta \mathcal{L}^\beta(\theta_{t-1}) \right) \right) \cdot \nabla_\theta \mathcal{L}^\beta(\theta_{t-1}) du \right\|_2^2 \tag{143}$$

$$\leq \left(\int_0^{\eta_0} \|\nabla_\theta^2 \mathcal{L}^\beta \left(\theta_{t-1} - u\nabla_\theta \mathcal{L}^\beta(\theta_{t-1}) \right)\|_2 \|\nabla_\theta \mathcal{L}^\beta(\theta_{t-1})\|_2 du \right)^2 \tag{144}$$

$$\leq \eta_0^2 \left(2(\lambda_{\max}(\mathbf{\Theta}) + \beta)\right)^2 \|\nabla_\theta \mathcal{L}^\beta(\theta_{t-1})\|_2^2 \tag{145}$$

$$\leq \eta_0 c_\beta \|\nabla_\theta \mathcal{L}^\beta(\theta_{t-1})\|_2^2. \tag{146}$$

In the last inequality, we chose the learning rate $\eta_0 \leq \frac{c_\beta}{4(\lambda_{\max}(\mathbf{\Theta})+\beta)^2}$ small enough. Similarly to the gradient flow case, one can derive such bounds for $\beta = 0$. Summing up the three terms,

$$\|\nabla_\theta \mathcal{L}^\beta(\theta_t)\|_2^2 \leq (1 - 2\eta_0 \frac{3}{2} c_\beta + \eta_0 c_\beta) \|\nabla_\theta \mathcal{L}^\beta(\theta_{t-1})\|_2^2 = (1 - 2\eta_0 c_\beta) \|\nabla_\theta \mathcal{L}^\beta(\theta_{t-1})\|_2^2. \tag{147}$$

Thus, by Bernoulli's inequality, and the induction hypothesis,

$$\|\nabla_\theta \mathcal{L}^\beta(\theta_t)\|_2 \leq \sqrt{1 - 2\eta_0 c_\beta} \|\nabla_\theta \mathcal{L}^\beta(\theta_{t-1})\|_2 \tag{148}$$

$$\leq (1 - \eta_0 c_\beta) \|\nabla_\theta \mathcal{L}^\beta(\theta_{t-1})\|_2 \tag{149}$$

$$\leq K R_0 (1 - \eta_0 c_\beta)^t. \tag{150}$$

Hence, $\|\theta_{t+1} - \theta_t\|_2 = \eta_0 \|\nabla_\theta \mathcal{L}^\beta(\theta_t)\|_2 \leq \eta_0 K R_0 (1 - \eta_0 c_\beta)^t$. From this, it follows that

$$\|\theta_{t+1} - \theta_0\|_2 \leq \sum_{u=0}^{t} \|\theta_{u+1} - \theta_u\|_2 \tag{151}$$

$$\leq \eta_0 K R_0 \sum_{u=0}^{t} (1 - \eta_0 c_\beta)^u \tag{152}$$

$$= \eta_0 K R_0 \frac{1 - (1 - \eta_0 c_\beta)^{t+1}}{\eta_0 c_\beta} \tag{153}$$

$$< \frac{K R_0}{c_\beta} = C. \tag{154}$$

This proves the first two inequalities. The rest follows directly from the local Lipschitzness of the Jacobian, like in the proof for the gradient flow. ■

F.2. Closeness to the linearized network along the regularized gradient descent

The following Theorem is the same as in the gradient flow case, and the proof is very similar, which is why we only provide the main idea.

Theorem 11 *Let $\beta \geq 0$. Let $\delta_0 > 0$ be arbitrarily small. Then, there are $C_1, C_2 > 0$, such that for n large enough, with probability of at least $1 - \delta_0$ over random initialization,*

$$\sup_{t\geq 0}\|\theta_t^{\text{lin}} - \theta_t\|_2 \leq C_1\frac{1}{\sqrt{n}}, \quad \forall \|x\|_2 \leq 1 : \sup_{t\geq 0}\|f^{\text{lin}}(x, \theta_t^{\text{lin}}) - f(x, \theta_t)\|_2 \leq C_2\frac{1}{\sqrt{n}}. \tag{155}$$

Proof Recall that

$$f^{\text{lin}}(x, \theta) = f(x, \theta_0) + J(x, \theta_0)(\theta - \theta_0), \tag{156}$$

and

$$\theta_{t+1}^{\text{lin}} = \theta_t^{\text{lin}} - \eta_0 \left(J(\theta_0)^\top g^{\text{lin}}(\theta_t^{\text{lin}}) + \beta(\theta_t^{\text{lin}} - \theta_0)\right). \tag{157}$$

The structure of the proof is the same as for the gradient flow. We only show how to bound the term $\|f(x, \theta_t) - f^{\text{lin}}(x, \theta_t)\|_2$. The bounds for the other terms can be done similarly. In particular, we will show by induction that

$$\|f(x, \theta_t) - f^{\text{lin}}(x, \theta_t)\|_2 \leq \eta_0\frac{1}{\sqrt{n}}K'CKR_0\sum_{u=0}^{t-1}(1 - \eta_0 c_\beta)^u. \tag{158}$$

For $t = 0$, this is true. Now, assume this holds for $s \leq t$, then

$$\|f(x, \theta_{t+1}) - f^{\text{lin}}(x, \theta_{t+1})\|_2 \tag{159}$$

$$=\|f(x, \theta_t) - f^{\text{lin}}(x, \theta_t)\|_2 + \|f(x, \theta_{t+1}) - f(x, \theta_t) - \left(f^{\text{lin}}(x, \theta_{t+1}) - f^{\text{lin}}(x, \theta_t)\right)\|_2. \tag{160}$$

By the chain rule and the fundamental theorem of calculus, we can write

$$\|f(x, \theta_{t+1}) - f(x, \theta_t) - \left(f^{\text{lin}}(x, \theta_{t+1}) - f^{\text{lin}}(x, \theta_t)\right)\|_2 \tag{161}$$

$$=\left\|\int_0^{\eta_0} J\left(x, \theta_t - u\nabla_\theta\mathcal{L}^\beta(\theta_t)\right)\nabla_\theta\mathcal{L}^\beta(\theta_t)du - \int_0^{\eta_0} J(x, \theta_0)\nabla_\theta\mathcal{L}^\beta(\theta_t)du\right\|_2 \tag{162}$$

$$\leq\int_0^{\eta_0}\|J\left(x, \theta_t - u\nabla_\theta\mathcal{L}^\beta(\theta_t)\right) - J(x, \theta_0)\|_2\|\nabla_\theta\mathcal{L}^\beta(\theta_t)\|_2 du \tag{163}$$

$$\leq\eta_0\frac{1}{\sqrt{n}}K'CKR_0\left(1 - \eta_0 c_\beta\right)^t. \tag{164}$$

In the last step we used Theorem 10. Thus,

$$\|f(x, \theta_{t+1}) - f^{\text{lin}}(x, \theta_{t+1})\|_2 \leq \eta_0\frac{1}{\sqrt{n}}K'CKR_0\left(\sum_{u=0}^{t-1}(1 - \eta_0 c_\beta)^u + \left(1 - \eta_0 c_\beta\right)^t\right). \tag{165}$$

This finishes the induction proof. Next, using the geometric series

$$\|f(x, \theta_t) - f^{\text{lin}}(x, \theta_t)\|_2 \leq \eta_0\frac{1}{\sqrt{n}}K'CKR_0\sum_{u=0}^{t-1}(1 - \eta_0 c_\beta)^u \tag{166}$$

$$< \eta_0\frac{1}{\sqrt{n}}K'CKR_0\frac{1}{\eta_0 c_\beta} \tag{167}$$

$$= \frac{1}{\sqrt{n}}K'C^2. \tag{168}$$

For the other inequalities one can proceed in the same way, using the fundamental theorem of calculus and the geometric series. ■

Appendix G. Shifting the network at initialization

Here, we prove that shifting the network at initialization makes it possible to include any prior mean, and compute the posterior mean with a single training run.

Theorem 3 *(Shifted Network.) Consider any function m. Given a random initialization θ_0, define shifted predictions $\tilde{f}_{\theta_0}(\mathbf{x}, \theta)$ as follows:*

$$\tilde{f}_{\theta_0}(\mathbf{x}, \theta) := f(\mathbf{x}, \theta) - f(\mathbf{x}, \theta_0) + m(\mathbf{x}). \tag{12}$$

Training this modified network (starting with θ_0) leads to the following output (in the infinite-width limit)

$$\tilde{f}_{\theta_0}(\mathbf{x}', \theta_\infty) = m(\mathbf{x}') + \mathbf{\Theta}_{\mathbf{x}',\mathbf{x}}(\mathbf{\Theta}_{\mathbf{x},\mathbf{x}} + \beta I)^{-1}(\mathbf{y} - m(\mathbf{x})). \tag{13}$$

This can be interpreted as the posterior mean of an NTK-GP with prior mean function m.

Proof The Jacobian of the shifted network is equal to the Jacobian of the original network:

$$J_{\tilde{f}}(x, \theta) = J_f(x, \theta). \tag{169}$$

Define the shifted labels $\tilde{\mathbf{y}} := \mathbf{y} + f(\mathbf{x}, \theta_0) - m(\mathbf{x})$. Then, $\tilde{f}(\mathbf{x}, \theta) - \mathbf{y} = f(\mathbf{x}, \theta) - \tilde{\mathbf{y}}$. Thus, training the network $\tilde{f}$ with regularized gradient flow/descent is equivalent to training f using the shifted labels $\tilde{\mathbf{y}}$, in the sense that the parameter update rule is the same. The latter leads to parameters θ_∞, for which (in the infinite-width limit)

$$f(\mathbf{x}', \theta_\infty) = f(\mathbf{x}', \theta_0) + \mathbf{\Theta}_{\mathbf{x}',\mathbf{x}} \left(\mathbf{\Theta}_{\mathbf{x},\mathbf{x}} + \beta I\right)^{-1} (\tilde{\mathbf{y}} - f(\mathbf{x}, \theta_0)). \tag{170}$$

By adding $-f(\mathbf{x}, \theta_0) + m(\mathbf{x})$ to both sides of the equation, and using $\tilde{\mathbf{y}} - f(\mathbf{x}, \theta_0) = \mathbf{y} - m(\mathbf{x})$, we get

$$\tilde{f}(\mathbf{x}', \theta_\infty) = m(\mathbf{x}') + \mathbf{\Theta}_{\mathbf{x}',\mathbf{x}} \left(\mathbf{\Theta}_{\mathbf{x},\mathbf{x}} + \beta I\right)^{-1} (\mathbf{y} - m(\mathbf{x})). \tag{171}$$

■

Appendix H. The Output of the Linearized Network is Gaussian over Random Initializations

Corollary 12 (Convergence under Regularized Gradient Flow/Descent) *Under regularized gradient flow/descent training, the output of a wide neural network converges in distribution to a Gaussian over random initialization as the width $n \to \infty$. Specifically, for test inputs $\mathbf{x}'$ and $t \to \infty$, the mean and covariance of the output distribution at convergence are*

$$\boldsymbol{\mu}(\mathbf{x}') = \mathbf{\Theta}_{\mathbf{x}',\mathbf{x}} \left(\mathbf{\Theta}_{\mathbf{x},\mathbf{x}} + \beta I\right)^{-1} \mathbf{y}, \tag{172}$$

$$\mathbf{\Sigma}(\mathbf{x}') = \mathbf{K}_{\mathbf{x}',\mathbf{x}'} + \mathbf{\Theta}_{\mathbf{x}',\mathbf{x}} \left(\mathbf{\Theta}_{\mathbf{x},\mathbf{x}} + \beta I\right)^{-1} \mathbf{K}_{\mathbf{x},\mathbf{x}} \left(\mathbf{\Theta}_{\mathbf{x},\mathbf{x}} + \beta I\right)^{-1} \mathbf{\Theta}_{\mathbf{x},\mathbf{x}'} \tag{173}$$

$$- \mathbf{\Theta}_{\mathbf{x}',\mathbf{x}} \left(\mathbf{\Theta}_{\mathbf{x},\mathbf{x}} + \beta I\right)^{-1} \mathbf{K}_{\mathbf{x},\mathbf{x}'} - \mathbf{K}_{\mathbf{x}',\mathbf{x}} \left(\mathbf{\Theta}_{\mathbf{x},\mathbf{x}} + \beta I\right)^{-1} \mathbf{\Theta}_{\mathbf{x},\mathbf{x}'}. \tag{174}$$

Note that the resulting covariance combines contributions from the NTK and NNGP kernels and therefore does not directly correspond to the posterior covariance of any GP.

Proof As we showed, for large enough layer width,

$$f(\mathbf{x}', \theta_\infty) = f(\mathbf{x}', \theta_0) + \mathbf{\Theta}_{\mathbf{x}',\mathbf{x}} \left(\mathbf{\Theta}_{\mathbf{x},\mathbf{x}} + \beta I\right)^{-1} (\mathbf{y} - f(\mathbf{x}, \theta_0)). \tag{175}$$

$f(\mathbf{x}', \theta_0)$ and $f(\mathbf{x}, \theta_0)$ jointly converge to a Gaussian with mean zero and covariance matrix given through the NNGP-kernel $\mathbf{K}$. From this, it directly follows that $f(\mathbf{x}', \theta_\infty)$ converges to a Gaussian with the given mean and covariance matrices.[9] ■

Appendix I. Convergence Plots for Different Noise Coefficients and Network Depths

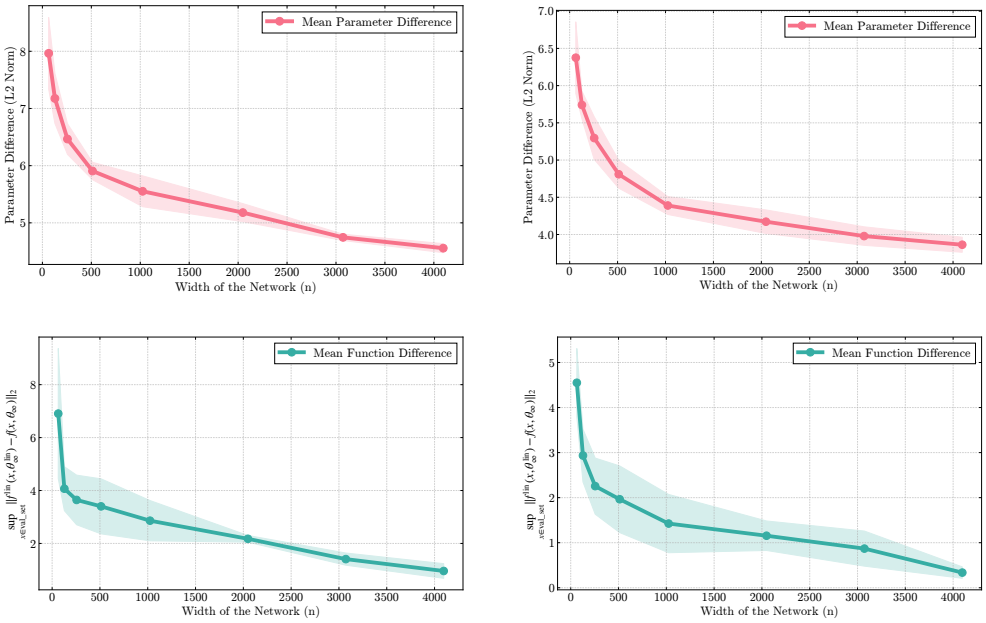

Figure 2: Parameter and function differences for additional network depths. (Top row) Parameter difference plots from Section 5.1. (Bottom row) Function difference plots from Section 5.2. (Left) Results for one fully connected hidden layer. (Right) Results for an MLP with three fully connected hidden layers. Increasing the network width reduces both parameter and function differences, backing up the theory. $\beta = 0.1$ was used.

9. The covariance matrix of $X + AY$, where X and Y are jointly Gaussian, is given by $\Sigma_X + A\Sigma_Y A^\top + A\Sigma_{X,Y} + \Sigma_{Y,X} A^\top$.