# On the Implicit Bias of Initialization Shape: Beyond Infinitesimal Mirror Descent

Shahar Azulay [1] Edward Moroshko [2] Mor Shpigel Nacson [2] Blake Woodworth [3] Nathan Srebro [3]
Amir Globerson [1] Daniel Soudry [2]

## Abstract

Recent work has highlighted the role of initialization scale in determining the structure of the solutions that gradient methods converge to. In particular, it was shown that large initialization leads to the neural tangent kernel regime solution, whereas small initialization leads to so called "rich regimes". However, the initialization structure is richer than the overall scale alone and involves relative magnitudes of different weights and layers in the network. Here we show that these relative scales, which we refer to as initialization shape, play an important role in determining the learned model. We develop a novel technique for deriving the inductive bias of gradient-flow and use it to obtain closed-form implicit regularizers for multiple cases of interest.

## 1. Introduction

Gradient descent (GD) is the main optimization tool used in deep learning. A wealth of recent work has highlighted the key role of this specific algorithm in the generalization performance of the learned model, when it is over-parameterized. Namely, the solutions that gradient descent converges to do not merely minimize the training error, but rather reflect the specific implicit biases of the optimization algorithm.

In light of this role for GD, many works have attempted to precisely characterize the implicit bias of GD in over-parameterized models. Technically, these exact characterizations amount to identifying a function $Q(\mathbf{w})$ of the model parameters $\mathbf{w}$ such that GD converges to a minimizer (or, more generally, a stationary point) of $Q(\mathbf{w})$ under the constraint of having zero training error. The form of $Q(\mathbf{w})$ can depend on various hyper-parameters (e.g., initialization, architecture, depth) and its dependence sheds light on how these hyper-parameters affect the final solution. This approach worked very well in several regimes.

The first regime is the ”Neural Tangent Kernel” (NTK) regime, which arises in networks that have an unrealistically large width (Du et al., 2019; Jacot et al., 2018; Nguyen, 2021) or initialization scale (Chizat et al., 2019). In this regime, networks converge to a linear predictor where the features are not learned, but determined by the initialization (via the so-called "Tangent Kernel"), and in this case $Q(\mathbf{w})$ is just the RKHS norm for the linear predictor. Therefore, it is not surprising that models trained in this regime typically do not achieve state-of-the-art empirical performance in challenging datasets where deep networks perform well. Accordingly, this regime is typically considered to be less useful for explaining the success of deep learning.

The second regime is the diametrically opposed "rich" regime, which was analyzed specifically for classification problems with vanishing loss (Lyu & Li, 2020b; Chizat & Bach, 2020). In this regime, the parameters converge to a stationary point (or sometimes a global minimum) of the optimization problem for minimizing $Q(\mathbf{w}) = ||\mathbf{w}||^2$ subject to margin constraints. This has been shown, under various assumptions, for linear neural networks (Gunasekar et al., 2018b; Ji & Telgarsky, 2019) and non-linear neural networks (Nacson et al., 2019; Lyu & Li, 2020a; Chizat & Bach, 2020). This regime is arguably more closely related to the performance of practical neural networks but, as Moroshko et al. (2020) show, reaching this regime requires unrealistically small loss values, even in toy problems.

Understanding the implicit bias in more realistic and practically relevant regimes remains challenging in models with more than one weight layer. Current results are restricted to very simple models such as diagonal linear neural networks with shared weights in regression (Woodworth et al., 2020)

[*]Equal contribution [1]The Blavatnik School of Computer Science, Tel Aviv University [2]Technion - Israel Institute of Technology [3]Toyota Technological Institute at Chicago. Correspondence to: Shahar Azulay <shaharazulay@mail.tau.ac.il>, Edward Moroshko <edward.moroshko@gmail.com>, Mor Shpigel Nacson <mor.shpigel@gmail.com>, Blake Woodworth <blake@ttic.edu>, Nathan Srebro <nati@ttic.edu>, Amir Globerson <amir.globerson@gmail.com>, Daniel Soudry <daniel.soudry@gmail.com>.

*Proceedings of the 38th International Conference on Machine Learning*, PMLR 139, 2021. Copyright 2021 by the author(s).

and classification (Moroshko et al., 2020), as well as generalized tensor formulations of networks (Yun et al., 2021). These results show exactly how the initialization scale determines the implicit bias of the model. However, these models are quite limited. For example, when the weights in different layers are shared, we cannot understand how the relative scale between layers affects the implicit bias.

Extending these exact results to a more realistic architectures is a considerable technical challenge. In fact, recent work has provided negative results with the square loss, for ReLU networks (even with a single neuron) (Vardi & Shamir, 2020) and for matrix factorization (Razin & Cohen, 2020; Li et al., 2021). Thus, finding scenarios where such a characterization of the implicit bias is possible and deriving its exact form is an open question, which we address here, making progress towards more realistic models.

Previous work (Woodworth et al., 2020; Gunasekar et al., 2018a; Yun et al., 2021; Vaskevicius et al., 2019; Amid & Warmuth, 2020a;b) that analyzes the exact implicit bias in such scenarios mostly focuses on least squares regression. All these analyses can be shown to be equivalent to expressing the dynamics of the predictor (which is induced by gradient flow on the model parameters) as Infinitesimal Mirror Descent (IMD), where the implicit bias then follows from Gunasekar et al. (2018a). This approach severely limits the model class that we can analyze because it is not always clear how to express the predictor dynamics as infinitesimal mirror descent. In fact, we can verify this is impossible to do even for basic models such as linear fully connected networks.

**Our Contributions:** In this work, we sidestep the above difficulty by developing a new method for characterizing the implicit bias and we apply it to obtain several new results:

- We identify degrees of freedom that allow us to modify the dynamics of the model so that it can be understood as infinitesimal mirror descent, without changing its implicit bias. In some cases, we show that this modification is equivalent to a non-linear "time-warping" (see Section 5).
- Our approach facilitates the analysis of a strictly more general model class. This allows us to investigate the exact implicit bias for models that could not be analyzed using previous techniques. Specific examples include diagonal networks with untied weights, fully connected two-layer[1] linear networks with vanishing initialization, and a two-layer single leaky ReLU neuron (see Sections 4, 6, and 8 respectively).

Our improved methodology is another step in the path toward analyzing the implicit bias in more realistic and complex models. Also, by being able to handle models with additional complexities, it already allows us to extend the scope of phenomena we can understand, shedding light on the importance of the initialization structure to implicit bias. For example,

- We show that the ratio between weights in different layers at initialization (the initialization "shape") has a marked effect on the learned model. We find how this property affects the final implicit bias (see Section 7).
- We prove that balanced initialization in diagonal linear nets improves convergence to the "rich regime", when the scale of the initialization vanishes (see Section 7.1).
- For fully connected linear networks, we prove that vanishing initialization results in a simple $\ell_2$-norm implicit bias for the equivalent linear predictor. A similar result was shown recently by Yun et al. (2021) under a specific condition on the directions of the weights at initialization (see Section 6).

Taken together, our analysis and results show the potential of our approach for discovering new implicit biases, and the insights these can provide about the effect of initialization on learned models.

In what follows, Sections 4-6 present derivations of implicit biases for several models of interest, and Section 7 uses these results to study the effect of initialization shape and scale on the learned models.

## 2. Preliminaries and Setup

Given a dataset of $N$ samples $\mathbf{X} = \left(\mathbf{x}^{(1)}, \cdots, \mathbf{x}^{(N)}\right) \in \mathbb{R}^{d\times N}$ with $N$ corresponding scalar labels $\mathbf{y} = \left(y^{(1)}, \cdots, y^{(N)}\right)^\top \in \mathbb{R}^N$ and a parametric model $f\left(\mathbf{x};\theta\right)$ with parameters $\theta$, we consider the problem of minimizing the square loss[2]

$$\mathcal{L}\left(\theta\right) \triangleq \frac{1}{2N}\sum_{n=1}^{N}\left(y^{(n)} - f(\mathbf{x}^{(n)};\theta)\right)^2 ,$$

using gradient descent with infinitesimally small stepsize (i.e., gradient flow)

$$\frac{d\theta}{dt} = -\nabla\mathcal{L}(\theta(t)) .$$

We focus on overparameterized models, where there are many solutions that achieve zero training loss, and assume that the loss is indeed (globally) minimized by gradient flow.

**Notation** For vectors $\mathbf{u}, \mathbf{v}$, we denote by $\mathbf{u}\circ\mathbf{v}$ the elementwise multiplication. In addition, $\|\cdot\|$ is the $\ell_2$-norm.

[1] By "two-layers" we mean two weight layers.

[2] The analysis in this paper can be extended to classification with the exp-loss along the lines of Moroshko et al. (2020).

## 3. Background: Deriving the Implicit Bias Using Infinitesimal Mirror Descent

We begin by describing the crux of current approaches to implicit bias analysis, and in Section 5 describe our "warping" approach that significantly extends these.

We focus on linear models that can be written as

$$f(\mathbf{x};\theta) = \tilde{\mathbf{w}}^\top \mathbf{x}\,,$$

where $\tilde{\mathbf{w}} = \tilde{\mathbf{w}}(\theta)$ is the equivalent linear predictor. Note that the model is linear in the input $\mathbf{x}$ but *not* in the parameters $\theta$. In Section 8, we show that our method can also be extended to non-linear models.

Our goal is to find a strictly convex function $Q(\tilde{\mathbf{w}})$ that captures the implicit regularization in the sense that the limit point of the gradient flow $\tilde{\mathbf{w}}(\infty)$ is the solution to the following optimization problem

$$\tilde{\mathbf{w}}(\infty) = \arg\min_{\mathbf{w}} Q(\mathbf{w}) \quad \text{s.t.} \quad \mathbf{X}^\top \mathbf{w} = \mathbf{y}\,. \tag{1}$$

We now describe a method used in Moroshko et al. (2020); Woodworth et al. (2020); Gunasekar et al. (2017); Amid & Warmuth (2020b) for obtaining $Q$ (below, we explain that these use essentially the same approach), and in Section 5 we present our novel approach. The KKT optimality conditions for Eq. (1) are that there exists $\boldsymbol{\nu} \in \mathbb{R}^N$ such that

$$\nabla Q(\tilde{\mathbf{w}}(\infty)) = \mathbf{X}\boldsymbol{\nu} \quad \text{and} \quad \mathbf{X}^\top \tilde{\mathbf{w}}(\infty) = \mathbf{y}\,. \tag{2}$$

Note that if $Q$ is strictly convex, Eq. (2) is sufficient to ensure that $\tilde{\mathbf{w}}(\infty)$ is the global minimum of Eq. (1). Therefore, our goal is to find a $Q$-function and $\boldsymbol{\nu} \in \mathbb{R}^N$ such that the limit point of gradient flow $\tilde{\mathbf{w}}(\infty)$ satisfies (2). Since we assumed that gradient flow converges to a zero-loss solution, we are only concerned with the stationarity condition

$$\nabla Q(\tilde{\mathbf{w}}(\infty)) = \mathbf{X}\boldsymbol{\nu}\,.$$

For the models we consider, the dynamics on $\tilde{\mathbf{w}}(t)$ can be written as

$$\frac{d\tilde{\mathbf{w}}(t)}{dt} = \mathbf{H}^{-1}(\tilde{\mathbf{w}}(t))\mathbf{X}\mathbf{r}(t) \tag{3}$$

for some $\mathbf{r}(t) \in \mathbb{R}^N$ and "metric tensor" $\mathbf{H} : \mathbb{R}^d \to \mathbb{R}^{d\times d}$, which is a positive definite matrix-valued function. In this case, we can write

$$\mathbf{H}(\tilde{\mathbf{w}}(t))\frac{d\tilde{\mathbf{w}}(t)}{dt} = \mathbf{X}\mathbf{r}(t) \tag{4}$$

and if

$$\exists Q : \mathbf{H}(\tilde{\mathbf{w}}(t)) = \nabla^2 Q(\tilde{\mathbf{w}}(t)) \tag{5}$$

we get that

$$\frac{d}{dt}(\nabla Q(\tilde{\mathbf{w}}(t))) = \mathbf{X}\mathbf{r}(t).$$

Therefore,

$$\nabla Q(\tilde{\mathbf{w}}(t)) - \nabla Q(\tilde{\mathbf{w}}(0)) = \int_0^t \mathbf{X}\mathbf{r}(t')dt'.$$

Denoting $\boldsymbol{\nu}(t) = \int_0^t \mathbf{r}(t')dt'$, if $\nabla Q(\tilde{\mathbf{w}}(0)) = 0$ then

$$\nabla Q(\tilde{\mathbf{w}}(t)) = \mathbf{X}\boldsymbol{\nu}(t)\,.$$

If a finite $\boldsymbol{\nu} = \lim_{t\to\infty} \boldsymbol{\nu}(t)$ exists then

$$\nabla Q(\tilde{\mathbf{w}}(\infty)) = \mathbf{X}\boldsymbol{\nu}\,,$$

which is the KKT stationarity condition. In case the above limit does not exist, we show in the next Lemma that if $\|\nabla Q(\tilde{\mathbf{w}})\| < \infty$ when $\|\tilde{\mathbf{w}}\| < \infty$ (which holds for the models we analyze in the next sections) we can still find some $\boldsymbol{\nu}$ such that the KKT stationarity condition holds.

**Lemma 3.1.** *For the dynamics defined in Eq.* (3) *with Eq.* (5)*, if* $\tilde{\mathbf{w}}(t)$ *converges to a global minimum then* $\|\tilde{\mathbf{w}}(\infty)\| < \infty$*. In addition, if* $\|\nabla Q(\tilde{\mathbf{w}})\| < \infty$ *when* $\|\tilde{\mathbf{w}}\| < \infty$ *then there exists a finite-norm* $\boldsymbol{\nu}$ *such that* $\nabla Q(\tilde{\mathbf{w}}(\infty)) = \mathbf{X}\boldsymbol{\nu}$*.*

The proof appears in Appendix A.

Thus, in this case, it is possible to find the $Q$-function by solving the differential equation

$$\mathbf{H}(\tilde{\mathbf{w}}(t)) = \nabla^2 Q(\tilde{\mathbf{w}}(t)). \tag{6}$$

The aforementioned papers now proceed to solve the differential equation $\mathbf{H} = \nabla^2 Q$ for $Q$. However, this proof strategy fundamentally relies on this differential equation having a solution, i.e., on $\mathbf{H}$ being a Hessian map. We emphasize that $\mathbf{H}$ being a Hessian map is a very special property, which does not hold for general positive definite matrix-valued functions.[3] Indeed, Eq. (6) only has a solution if $\mathbf{H}$ satisfies the Hessian-map condition (e.g., see Gunasekar et al., 2020)

$$\forall_{i,j,k} : \frac{\partial \mathbf{H}_{i,j}(\mathbf{w})}{\partial \mathbf{w}_k} = \frac{\partial \mathbf{H}_{i,k}(\mathbf{w})}{\partial \mathbf{w}_j}\,. \tag{7}$$

As we discuss in Section 6, this condition is not met for natural models like fully connected linear neural networks, and therefore a new approach is needed.

[3] Indeed, Gunasekar et al. (2020) show that the innocent-looking $\mathbf{w} \mapsto I + \mathbf{w}\mathbf{w}^\top$ is provably not the Hessian of any function, which can be confirmed by checking the condition Eq. (7).

### 3.1. Relation to Infinitesimal Mirror Descent

The approach described above is a different presentation of the equivalent view of Gunasekar et al. (2018a). They show that when the dynamics on $\tilde{\mathbf{w}}$ can be expressed as "Infinitesimal Mirror Descent" (IMD) with respect to a strongly convex potential $\psi$

$$\frac{d\tilde{\mathbf{w}}(t)}{dt} = -\nabla^2\psi(\tilde{\mathbf{w}}(t))^{-1}\nabla\mathcal{L}(\tilde{\mathbf{w}}(t)) \,, \tag{8}$$

then the limit point $\tilde{\mathbf{w}}(\infty)$ is described by

$$\tilde{\mathbf{w}}(\infty) = \arg\min_{\mathbf{w}} D_\psi(\mathbf{w}, \mathbf{w}(0)) \quad \text{s.t.} \quad \mathbf{X}^\top\mathbf{w} = \mathbf{y} \,,$$

where $D_\psi(\mathbf{w}, \mathbf{w}') = \psi(\mathbf{w}) - \psi(\mathbf{w}') - \langle\nabla\psi(\mathbf{w}'), \mathbf{w} - \mathbf{w}'\rangle$ is the Bregman divergence associated with $\psi$. Furthermore, when $\tilde{\mathbf{w}}$ is initialized with $\nabla\psi(\tilde{\mathbf{w}}(0)) = 0$, then

$$\tilde{\mathbf{w}}(\infty) = \arg\min_{\mathbf{w}} \psi(\mathbf{w}) \quad \text{s.t.} \quad \mathbf{X}^\top\mathbf{w} = \mathbf{y} \,.$$

Comparing Eqs. (3) and (8), we see that the infinitesimal mirror descent view is equivalent to the approach we have described, with $\psi$ corresponding exactly to $Q$.

Although it may have been presented in different ways, these analysis techniques have formed the basis for all of the existing exact[4] characterizations of implicit bias for linear models with square loss (outside of the NTK regime) that we are aware of (e.g. Gunasekar et al. (2017); Woodworth et al. (2020); Amid & Warmuth (2020a); Moroshko et al. (2020)). In Section 5 we show how to extend this analysis to cases where $\mathbf{H}$ is not a Hessian map.

## 4. Diagonal Linear Networks

All previous analyses of the exact implicit bias for linear models with square loss (outside of the NTK regime) are limited to cases where the different layers share weights. In this section, we will remove this assumption, which allows us to analyze the effect of the relative scales of initialization between different layers in Section 7.1. To begin, we examine a two-layer "diagonal linear network" with untied weights

$$\begin{aligned} f(\mathbf{x}; \mathbf{u}_+, \mathbf{u}_-, \mathbf{v}_+, \mathbf{v}_-) &= (\mathbf{u}_+ \circ \mathbf{v}_+ - \mathbf{u}_- \circ \mathbf{v}_-)^\top \mathbf{x} \\ &= \tilde{\mathbf{w}}^\top \mathbf{x} \,, \end{aligned} \tag{9}$$

where $\tilde{\mathbf{w}} = \mathbf{u}_+ \circ \mathbf{v}_+ - \mathbf{u}_- \circ \mathbf{v}_-$.

**Previous Results:** Woodworth et al. (2020); Moroshko et al. (2020) analyzed these models for the special case of shared weights where $\mathbf{u}_+ = \mathbf{v}_+$ and $\mathbf{u}_- = \mathbf{v}_-$, corresponding to the model

$$f(\mathbf{x}; \mathbf{u}_+, \mathbf{u}_-) = \left(\mathbf{u}_+^2 - \mathbf{u}_-^2\right)^\top \mathbf{x} \,.$$

Both of these works focused on unbiased initialization, i.e., $\mathbf{u}_+(0) = \mathbf{u}_-(0) = \alpha\mathbf{u}$ (for some fixed $\mathbf{u}$). In Yun et al. (2021) these results were generalized to a tensor formulation, yet one which does not allow untied weights (as in Eq. (9)).

For regression with the square loss, Woodworth et al. (2020) showed how the scale of initialization $\alpha$ controls the limit point of gradient flow between two extreme regimes. When $\alpha$ is large, gradient flow is biased towards the minimum $\ell_2$-norm solution (Chizat et al., 2019), corresponding to the kernel regime; when $\alpha$ is small, gradient flow is biased towards the minimum $\ell_1$-norm solution, corresponding to the rich regime; and intermediate $\alpha$ leads to some combination of these biases. For classification with the exponential loss, Moroshko et al. (2020) showed how both the scale of initialization and the optimization accuracy control the implicit bias between the NTK and rich regimes.

**Our Results:** In this work, we analyze the model (9) for the square loss and show how both the initialization scale and the initialization shape (see Section 7.1) affect the implicit bias. To find the implicit bias of this model, we show how to express the training dynamics of this model in the form Eq. (3), which enables the use of the IMD approach (Sec. 3).

To simplify the presentation, we focus on unbiased initialization, where $\mathbf{u}_+(0) = \mathbf{u}_-(0)$ and $\mathbf{v}_+(0) = \mathbf{v}_-(0)$, which allows scaling the initialization without scaling the output (Chizat et al., 2019). See Appendix B for a more general result with any initialization.

**Theorem 4.1.** *For unbiased initialization, if the gradient flow solution $\tilde{\mathbf{w}}(\infty)$ satisfies $\mathbf{X}^\top\tilde{\mathbf{w}}(\infty) = \mathbf{y}$, then:*

$$\tilde{\mathbf{w}}(\infty) = \arg\min_{\mathbf{w}} Q_{\boldsymbol{k}}(\mathbf{w}) \quad \text{s.t.} \ \mathbf{X}^\top\mathbf{w} = \mathbf{y}$$

*where*

$$Q_{\boldsymbol{k}}(\mathbf{w}) = \sum_{i=1}^{d} q_{k_i}(w_i) \ , \tag{10}$$

$$\begin{aligned} q_k(x) &= \frac{1}{2}\int_0^x \operatorname{arcsinh}\left(\frac{2z}{\sqrt{k}}\right) dz \\ &= \frac{\sqrt{k}}{4}\left[1 - \sqrt{1 + \frac{4x^2}{k}} + \frac{2x}{\sqrt{k}}\operatorname{arcsinh}\left(\frac{2x}{\sqrt{k}}\right)\right] \end{aligned}$$

*and* $\sqrt{k_i} = 2\left(u_{+,i}^2(0) + v_{+,i}^2(0)\right)$.

The proof appears in Appendix B.

The function $Q_{\boldsymbol{k}}(\mathbf{w})$ in (10) generalizes the implicit regularizer found by Woodworth et al. (2020) to two layers

[4] There are some statistical (i.e. non-exact) results for matrix factorization with vanishing initialization under certain data assumptions (Li et al., 2018).

with untied parameters. As expected, Eq. (10) reduces to Woodworth et al. (2020) when $\mathbf{u}_+(0) = \mathbf{v}_+(0)$ and $\mathbf{u}_-(0) = \mathbf{v}_-(0)$. Unlike the previous result, $Q_{\boldsymbol{k}}(\mathbf{w})$ can be used to study how the relative magnitude of $\mathbf{u}$ versus $\mathbf{v}$ at initialization affects the implicit bias. We present this analysis in Section 7.1, and highlight how initialization scale and shape have separate effects on the resulting model.

## 5. Warping Infinitesimal Mirror Descent

Our next goal is to go beyond the simplistic "diagonal" architecture to a fully connected one. However, deriving the implicit bias for non-diagonal models using the IMD approach (Section 3) is not always possible since the $\mathbf{H}$ in Eq. (3) might not be a Hessian map. Indeed, this condition does not hold for linear fully connected neural networks. To sidestep this issue, we next present our new technique for finding the implicit bias when $\mathbf{H}$ is not a Hessian map. We begin by multiplying both sides of Eq. (4) by a smooth, positive function $g : \mathbb{R}^d \to (0, \infty)$ to get

$$g(\tilde{\mathbf{w}}(t))\mathbf{H}(\tilde{\mathbf{w}}(t))\frac{d\tilde{\mathbf{w}}(t)}{dt} = g(\tilde{\mathbf{w}}(t))\mathbf{X}\mathbf{r}(t) \, .$$

Perhaps surprisingly, for the right choice of $g$, the differential equation $g(\mathbf{w})\mathbf{H}(\mathbf{w}) = \nabla^2 Q(\mathbf{w})$ can have a solution even when $\mathbf{H}(\mathbf{w}) = \nabla^2 Q(\mathbf{w})$ does not! When such a $g$ can be found, we can continue the analysis just as before,

$$g(\tilde{\mathbf{w}}(t))\mathbf{H}(\tilde{\mathbf{w}}(t)) = \nabla^2 Q(\tilde{\mathbf{w}}(t)) \, . \tag{11}$$

We see that

$$\frac{d}{dt}(\nabla Q(\tilde{\mathbf{w}}(t))) = g(\tilde{\mathbf{w}}(t))\mathbf{X}\mathbf{r}(t) \, , \tag{12}$$

and we conclude

$$\nabla Q(\tilde{\mathbf{w}}(t)) - \nabla Q(\tilde{\mathbf{w}}(0)) = \int_0^t g(\tilde{\mathbf{w}}(t'))\mathbf{X}\mathbf{r}(t')dt'.$$

We require that for our chosen $g$ function $\int_0^\infty g(\tilde{\mathbf{w}}(t'))\mathbf{r}(t')dt'$ exists and is finite, in which case, as before, we denote $\boldsymbol{\nu} = \int_0^\infty g(\tilde{\mathbf{w}}(t'))\mathbf{r}(t')dt'$ so $\tilde{\mathbf{w}}(\infty)$ satisfies the stationarity condition when $\nabla Q(\tilde{\mathbf{w}}(0)) = 0$:

$$\nabla Q(\tilde{\mathbf{w}}(\infty)) = \mathbf{X}\boldsymbol{\nu} \, .$$

This establishes that $Q$ captures the implicit bias, and all that remains is to describe how to find a $g$ such that Eq. (11) has a solution. For example, for a two-layer linear fully connected network with single neuron, we begin from the Ansatz that $Q\left(\tilde{\mathbf{w}}(t)\right)$ can be written as

$$Q\left(\tilde{\mathbf{w}}(t)\right) = \hat{q}\left(\|\tilde{\mathbf{w}}(t)\|\right) + \mathbf{z}^\top \tilde{\mathbf{w}}(t) \tag{13}$$

for some scalar function $\hat{q}$ and a fixed vector $\mathbf{z} \in \mathbb{R}^d$.

By comparing Eq. (11) with the Hessian of Eq. (13), we solve for $\hat{q}$ and $g$, and use the condition $\nabla Q(\tilde{\mathbf{w}}(0)) = 0$ to determine $\mathbf{z}$. For more than one neuron, the analysis becomes more complicated because we will choose different $g$ functions for each neuron.

**The $g$ Function as a "Time Warping".** The above approach can also be interpreted as a non-linear warping of the time axis. The key idea is that rescaling "time" for an ODE affects neither the set of points visited by the solution nor the eventual limit point. Our approach essentially finds a rescaling that yields dynamics that allow solving for $Q$.

Specifically, if $\mathbf{w}(t) \in \mathbb{R}^d$ is a solution to the ODE

$$\frac{d}{dt}\mathbf{w}(t) = f(\mathbf{w}(t)) \tag{14}$$

for any "time warping" $\tau : \mathbb{R} \to \mathbb{R}$ such that $\tau(0) = 0$, $\lim_{t\to\infty} \tau(t) = \infty$, and $\exists c > 0 : c < \tau'(t) < \infty$, then $\mathbf{w}(\tau(t))$ is a solution to the ODE

$$\frac{d}{dt}\mathbf{w}(\tau(t)) = \tau'(t)f(\mathbf{w}(\tau(t))) \, . \tag{15}$$

Therefore, the set of points visited by $\mathbf{w}(t)$ and $\mathbf{w}\left(\tau(t)\right)$ are the same, and so are their limit points $\mathbf{w}(\infty) = \mathbf{w}(\tau(\infty))$. All that changes is the time at which these points are reached. Furthermore, since $\tau' > 0$, $\tau$ is invertible so, conversely, a solution for Eq. (15) can also be converted into a solution for Eq. (14) via the warping $\tau^{-1}$. In this way, we can interpret $g$ as a time warping function which transforms the ODE

$$\frac{d}{d\tau}\nabla Q(\tilde{\mathbf{w}}(\tau)) = \mathbf{X}\mathbf{r}(\tau) \tag{16}$$

into Eq. (12), which is equivalent in the sense that it does not affect the set of models visited by gradient flow (it only affects the time they are visited). In particular, let $\tilde{\mathbf{w}}(\tau)$ be a solution to Eq. (16), then $\tilde{\mathbf{w}}(\tau(t))$ is a solution for Eq. (12) for $\tau(t) = \int_0^t g(\tilde{\mathbf{w}}(t'))dt'$. So long as $\tau(\infty) = \int_0^\infty g(\tilde{\mathbf{w}}(t'))dt' = \infty$ so that $\tilde{\mathbf{w}}(\tau(t))$ does not "stall out," we conclude that the limit points of Eqs. (12) and (16) are the same.

## 6. Fully Connected Linear Networks

In this section we examine the class of fully connected linear networks of depth 2, defined as

$$f(\mathbf{x}; \{a_i\}, \{\mathbf{w}_i\}) = \sum_{i=1}^m a_i \mathbf{w}_i^\top \mathbf{x} = \tilde{\mathbf{w}}^\top \mathbf{x} \, ,$$

where $\tilde{\mathbf{w}} \triangleq \sum_{i=1}^m \tilde{\mathbf{w}}_i$, and $\tilde{\mathbf{w}}_i \triangleq a_i \mathbf{w}_i$.

For this model, the Hessian-map condition (Eq. (7)) does not hold and thus our analysis uses the "warped IMD" technique described in Section 5. In addition, our analysis of the implicit bias employs the following *balancedness* properties for gradient flow shown by Du et al. (2018):

Theorem 2.1 of Du et al. (2018) states that

$$\forall t: \;\; a_i^2(t) - \left\|\mathbf{w}_i(t)\right\|^2 = a_i^2(0) - \left\|\mathbf{w}_i(0)\right\|^2 \triangleq \delta_i \,.$$

In addition, Theorem 2.2 (a stronger balancedness property for linear activations) of Du et al. (2018) states that

$$\begin{aligned} \forall t: \;\; & \mathbf{a}(t)\mathbf{a}(t)^T - \mathbf{W}(t)^\top \mathbf{W}(t) \\ & = \mathbf{a}(0)\mathbf{a}(0)^T - \mathbf{W}(0)^\top \mathbf{W}(0) \triangleq \mathbf{\Delta} \,, \end{aligned}$$

where $\mathbf{\Delta} \in \mathbb{R}^{m \times m}$, $\mathbf{a} = (a_1, ..., a_m)^\top$ and $\mathbf{W} = (\mathbf{w}_1, ..., \mathbf{w}_m) \in \mathbb{R}^{d \times m}$.

First, we derive the implicit bias for a fully connected single-neuron assuming $\delta_i \geq 0$ (which ensures that we can write the dynamics in the form (4) for invertible $\mathbf{H}$), and then expand our results to multi-neuron networks under more specific settings.

**Theorem 6.1.** *For a depth* $2$ *fully connected network with a single hidden neuron* ($m = 1$)*, any* $\delta \geq 0$*, and initialization* $\tilde{\mathbf{w}}(0) = a(0)\mathbf{w}(0) \neq \mathbf{0}$*, if the gradient flow solution* $\tilde{\mathbf{w}}(\infty)$ *satisfies* $\mathbf{X}^\top \tilde{\mathbf{w}}(\infty) = \mathbf{y}$*, then:*

$$\tilde{\mathbf{w}}(\infty) = \arg\min_{\mathbf{w}} q_{\delta, \tilde{\mathbf{w}}(0)}(\mathbf{w}) \quad \text{s.t.} \;\; \mathbf{X}^\top \mathbf{w} = \mathbf{y}$$

*where* $q_{\delta, \tilde{\mathbf{w}}(0)}(\mathbf{w}) = \hat{q}_\delta\left(\|\mathbf{w}\|\right) + \mathbf{z}^\top \mathbf{w}$ *for*

$$\hat{q}_\delta(x) = \frac{\left(x^2 - \frac{\delta}{2}\left(\frac{\delta}{2} + \sqrt{x^2 + \frac{\delta^2}{4}}\right)\right)\sqrt{\sqrt{x^2 + \frac{\delta^2}{4}} - \frac{\delta}{2}}}{x}$$

$$\mathbf{z} = -\frac{3}{2}\sqrt{\sqrt{\left\|\tilde{\mathbf{w}}(0)\right\|^2 + \frac{\delta^2}{4}} - \frac{\delta}{2}}\,\frac{\tilde{\mathbf{w}}(0)}{\left\|\tilde{\mathbf{w}}(0)\right\|} \,.$$

The proof appears in Appendix C.

The function $q_{\delta, \tilde{\mathbf{w}}(0)}(\mathbf{w})$ above again reveals interesting tradeoffs between initialization scale and shape, which we discuss in Section 7.2.

In order to extend this result beyond a single neuron we require additional conditions to be met. For a multi-neuron network, in contrast to the single neuron case, we cannot use globally the "time warping" technique since it requires multiplying each neuron by a different $g$ function. However, for the special case of strictly balanced initialization, $\mathbf{\Delta} = 0$, we can extend this result to $m > 1$.

**Proposition 6.2.** *For a multi-neuron network* ($m > 1$) *with strictly balanced initialization* ($\mathbf{\Delta} = 0$)*, assume* $\tilde{\mathbf{w}}(0) \neq \mathbf{0}$*. If the gradient flow solution* $\tilde{\mathbf{w}}(\infty)$ *satisfies* $\mathbf{X}^\top \tilde{\mathbf{w}}(\infty) = \mathbf{y}$*, then:*

$$\tilde{\mathbf{w}}(\infty) = \arg\min_{\mathbf{w}} \left[\left\|\mathbf{w}\right\|^{3/2} - \frac{3}{2}\left\|\tilde{\mathbf{w}}(0)\right\|^{-1/2} \tilde{\mathbf{w}}(0)^\top \mathbf{w}\right]$$

$$\text{s.t.} \;\; \mathbf{X}^\top \mathbf{w} = \mathbf{y} \,.$$

The proof appears in Appendix D.

Next, we show that for infinitesimal nonzero initialization, the equivalent linear predictor of the multi-neuron linear network is biased towards the minimum $\ell_2$-norm.

**Theorem 6.3.** *For a multi-neuron network, and for nonzero infinitesimal initialization, i.e.* $\forall i: \mathbf{0} \neq \|\tilde{\mathbf{w}}_i(0)\| \to 0$*, if the gradient flow solution* $\tilde{\mathbf{w}}(\infty)$ *satisfies* $\mathbf{X}^\top \tilde{\mathbf{w}}(\infty) = \mathbf{y}$*, then:*

$$\tilde{\mathbf{w}}(\infty) = \text{argmin}_{\mathbf{w}} \left\|\mathbf{w}\right\| \;\; \text{s.t.} \;\; \mathbf{X}^\top \mathbf{w} = \mathbf{y} \,.$$

The proof appears in Appendix E.

Note that for infinitesimal initialization, as above, the training dynamics of fully connected linear networks is not captured by the neural tangent kernel (Jacot et al., 2018), i.e., the tangent kernel is not fixed during training, so that we are not in the NTK regime (Chizat et al., 2019; Woodworth et al., 2020). Yet, the implicit bias is towards a solution that can be captured by a kernel ($\ell_2$-norm). Though in other models, this limit coincides with the "rich" regime (Woodworth et al., 2020), in these cases the $Q$ function is not an RKHS. Since in our case the $Q$ function is an RKHS, calling this regime "rich" is problematic. Therefore, we propose to call this vanishing initialization regime as the *Anti-NTK* regime — since this limit is diametrically opposed to the NTK regime, which is reached at the limit of infinite initialization (Chizat et al., 2019; Woodworth et al., 2020). This regime coincides with the "rich" regimes in models where the $Q$ function is not an RKHS norm in that limit.

Yun et al. (2021) (Theorem 7) provide a similar observation of such an $\ell_2$ minimization result as in Theorem 6.3 for fully connected linear nets in a regression setting, under vanishing initialization. However, their result is under a specific condition on the directions of the weights at initialization, whereas we assume that gradient flow converges to a global minimum.

For classification problems (e.g. with exponential or logistic loss) it was proven that the predictor of fully connected linear nets converges to the max-margin solution with the minimum $\ell_2$ norm (Ji & Telgarsky, 2019), in the regime where the loss vanishes. This regime is closely related to the Anti-NTK regime since in a classification setting, vanishing loss and vanishing initialization can yield similar $Q$ function (Moroshko et al., 2020).

## 7. The Effect of Initialization Shape and Scale

Chizat et al. (2019) identified the scale of the initialization as the crucial parameter for entering the NTK regime, and Woodworth et al. (2020) further characterized the transition

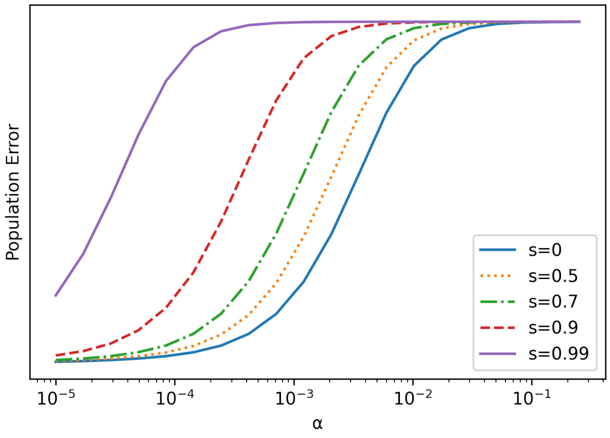

*Figure 7.1.* The population error of the gradient flow solution for a diagonal linear network as a function of initialization scale $\alpha$ and shape $s$, in the sparse regression problem described in Section 9.

between the NTK and rich regimes as a function of the initialization scale, and how this affects the generalization properties of the model. Both showed the close relation between the initialization scale and the model width.

However, we identify another hyper-parameter that controls this transition between NTK and rich regimes for two-layer models, the ***shape of the initialization***, which describes the relative scale between different layers.

We first demonstrate this by using the example of two-layer diagonal linear networks described in Section 4.

### 7.1. Diagonal Linear Networks

We denote the per-neuron initialization shape $s_i$ and scale $\alpha_i$ as

$$s_i = \frac{\frac{|v_{+,i}(0)|}{|u_{+,i}(0)|} - 1}{\frac{|v_{+,i}(0)|}{|u_{+,i}(0)|} + 1} \quad , \quad \alpha_i = |u_{+,i}(0)|\,|v_{+,i}(0)| \ .$$

We can notice from Theorem 4.1 that $\sqrt{k_i} = 2\left(u^2_{+,i}(0) + v^2_{+,i}(0)\right)$ controls the transition between the NTK and rich regimes. Using the definitions of the initialization shape and scale we write

$$\sqrt{k_i} = 4\alpha_i \frac{1+s_i^2}{1-s_i^2} \, .$$

Since $-1 < s_i < 1$, we can more accurately say that $\hat{k}_i = \frac{\alpha_i}{1-s_i^2}$ is the factor controlling the transition.

For simplicity, we next assume that $\alpha_i = \alpha, \quad s_i = s \ \forall i \in [d]$. We can notice that for $k \to \infty$, i.e. $\frac{\alpha}{1-s^2} \to \infty$ we get that

$$Q(w) = \sum_{i=1}^{d} q(w_i) = \sum_{i=1}^{d} \frac{1}{2\left(u^2_{+,i}(0) + v^2_{+,i}(0)\right)} w_i^2 \ ,$$

which is exactly the minimum RKHS norm with respect to the NTK at initialization. Therefore, $k \to \infty$ leads to the NTK regime. However, for $k \to 0$, i.e. $\frac{\alpha}{1-s^2} \to 0$ we get that

$$Q(\mathbf{w}) = \sum_{i=1}^{d} |w_i| = \|\mathbf{w}\|_1 \ ,$$

which describes the rich regime. The proof for the above two claims appears in Appendix F.

Therefore, both the initialization scale $\alpha$ and the initialization shape $s$ affect the transition between NTK and rich regimes. While $\alpha \to 0$ pushes to the rich regime, $|s| \to 1$ pushes towards the NTK regime. Since both limits can take place simultaneously, the regime we will converge to in this case is captured by the joint limit

$$\hat{k}^\star = \lim_{\alpha \to 0, |s| \to 1} \frac{\alpha}{1-s^2} \ .$$

Intuitively, when $\alpha \to 0$ faster than $s \to 1$ we will be in the rich regime, corresponding to $\hat{k}^\star = 0$. However, when $s \to 1$ faster than $\alpha \to 0$ we will be in the NTK regime, corresponding to $\hat{k}^\star = \infty$. For any $0 < \hat{k}^\star < \infty$ the $Q$-function in Eq. (10) captures the implicit bias.

Figure 7.1 demonstrates the interplay between the scale and the shape of initialization. See Section 9 for details. The figure shows the population error (i.e., test error) of the learned model for different choices of scale $\alpha$ and shape $s$. Since in this case the ground truth is a sparse regressor, low error corresponds to the rich regime whereas high error corresponds to the NTK regime. It can be seen that as the shape $s$ approaches 1, the model tends to converge to a solution in the NTK regime, or an intermediate regime even for very small initialization scales. These results give further credence to the idea that the learned model will perform best when trained with balanced initialization ($s = 0$).

### 7.2. Fully Connected Linear Networks

We begin by characterizing the effect of the initialization scale and shape for a single linear neuron with two layers, analyzed in Section 6. Our characterization is based on the $q_{\delta,\tilde{\mathbf{w}}(0)}(\mathbf{w})$ function in Theorem 6.1. Due to the lack of space we defer the detailed analysis to Appendix G and provide here a summary of the results.

Similarly to the diagonal model, we again define the initialization shape parameter $s$ and scale parameter $\alpha$ as

$$s = \frac{\frac{|a(0)|}{\|\mathbf{w}(0)\|} - 1}{\frac{|a(0)|}{\|\mathbf{w}(0)\|} + 1} \quad , \quad \alpha = |a(0)|\,\|\mathbf{w}(0)\| \ .$$

Note that Theorem 6.1 is correct for $0 \leq s < 1$ and any $\alpha > 0$. We also employ the initialization orientation, defined as $\mathbf{u} = \frac{\mathbf{w}(0)}{\|\mathbf{w}(0)\|}$. Given $\alpha, s, \mathbf{u}$ we identify a few limit cases.

First, consider some fixed shape $0 \leq s < 1$. When $\alpha \to 0$ we will be in the Anti-NTK regime, where we obtain the minimum $\ell_2$-norm predictor. However, when $\alpha \to \infty$ we will be in the NTK regime, where the tangent kernel is fixed during training, and the implicit bias is given by the minimum RKHS norm predictor. Indeed, in this case we show in Appendix G that

$$q(\tilde{\mathbf{w}}) \propto \left(\tilde{\mathbf{w}} - \tilde{\mathbf{w}}\left(0\right)\right)^{\top} \mathbf{B} \left(\tilde{\mathbf{w}} - \tilde{\mathbf{w}}\left(0\right)\right) \ ,$$

where

$$\mathbf{B} = \mathbf{I} - \frac{\left(1-s\right)^2}{2\left(1+s^2\right)} \mathbf{u}\mathbf{u}^{\top}$$

and it is easy to verify that the tangent kernel is given by $K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^{\top}\mathbf{B}^{-1}\mathbf{x}'$.

Therefore, for any fixed shape, taking $\alpha$ from 0 to $\infty$ we move from the Anti-NTK regime (with $\ell_2$ implicit bias) to the NTK regime where the bias is given by a Mahalanobis norm that depends on the shape and initialization orientation. Note that when $s \approx 1$, we have $\mathbf{B} \approx \mathbf{I}$, and thus we obtain the $\ell_2$ bias about the initialization, namely $\arg\min_{\tilde{\mathbf{w}}} \|\tilde{\mathbf{w}} - \tilde{\mathbf{w}}(0)\|$. In the bottom row of Figure 7.2 we illustrate the $q$ function for $s = 0.1$ and different values of $\alpha$. Note that for intermediate $\alpha$ we obtain non-kernel implicit bias.

On the other hand, for any fixed scale $\alpha$, taking $s \to 1$ we will be in the NTK regime. This is because in this case the gradients of $a$ are much smaller that the gradients of $\mathbf{w}$, and thus effectively, only the $\mathbf{w}$ parameters will optimize. Therefore, in this case we obtain a linear model (linear in the parameters) and the $\ell_2$ bias about the initialization, $\arg\min_{\tilde{\mathbf{w}}} \|\tilde{\mathbf{w}} - \tilde{\mathbf{w}}(0)\|$. This phenomenon is illustrated in the top row of Figure 7.2.

To sum-up, here we show the existence of an intermediate regime, which describes an implicit bias that cannot be captured by a kernel, and that it is obtainable under a balanced initialization ($s \approx 0$). This observation is in line with our observation for diagonal models in Section 7.1. However, for the fully connected case the "non-kernel" intermediate regime is not necessarily preferable to the Anti-NTK regime, where a minimum $\ell_2$ norm solution is obtained.

## 8. Two-Layer Single Leaky ReLU Neuron

We further extend our analysis to the class of fully connected two-layer single neuron with Leaky ReLU activations, $\sigma(x) = \max(x, \rho x)$ for $\rho > 0$. This is a first step in analyzing the implicit bias of practical non-linear fully connected models for regression with the square loss.

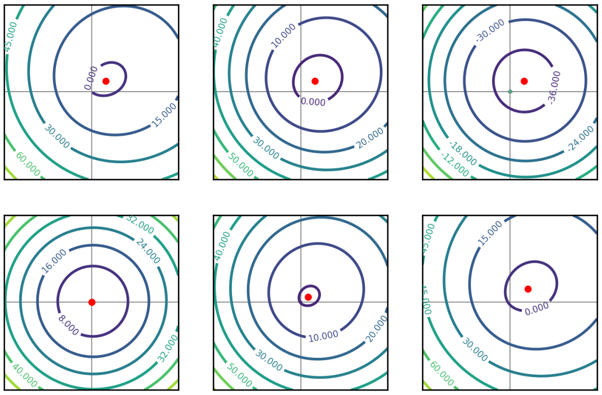

*Figure 7.2.* Contour plots of $q_{\delta,\tilde{\mathbf{w}}(0)}(\tilde{\mathbf{w}})$ presented in Theorem 6.1 for the case of $d = 2$, $\tilde{\mathbf{w}}(0) = \alpha \cdot [0.6, 0.8]$. Top row: $\alpha = 2$ and $s = [0, 0.2, 0.8]$ (left to right in order). Bottom row: $s = 0.1$ and $\alpha = [0.01, 1, 2.5]$ (left to right in order). The red dot marks the vector $\tilde{\mathbf{w}}(0)$.

**Theorem 8.1.** *For a single-neuron network with Leaky ReLU activation $\sigma$ of any slope $\rho > 0$, and for any $\delta \geq 0$, assume $a(0)\mathbf{w}(0) \neq \mathbf{0}$. If the gradient flow solution $(a(\infty), \mathbf{w}(\infty))$ satisfies $a(\infty)\sigma(\mathbf{X}^{\top}\mathbf{w}(\infty)) = \mathbf{y}$, then*

$$a(\infty)\mathbf{w}(\infty) = \underset{\tilde{\mathbf{w}} = a\mathbf{w}}{\arg\min}\, q_{\delta}(\tilde{\mathbf{w}}) \quad \text{s.t. } a\sigma(\mathbf{X}^{\top}\mathbf{w}) = \mathbf{y}$$

*and $q_{\delta}(\mathbf{w})$ is identical to the definition given in Theorem 6.1.*

*Proof.* Since we assumed that $\delta \geq 0$, a non-zero initialization $a(0)\mathbf{w}(0) \neq \mathbf{0}$ and that we converge to zero-loss solution, we get that $a(t)$ does not change sign during training. In case $a(t) \geq 0$ for all $t$, since $\sigma$ is 1-positive homogeneous, the single-neuron model $a(t)\sigma\left(\mathbf{w}(t)^{\top}\mathbf{x}\right)$ can be written as $\sigma\left(a(t)\mathbf{w}(t)^{\top}\mathbf{x}\right)$. Next, denoting $\tilde{\mathbf{w}} = a\mathbf{w}$, since $\sigma$ is strictly monotonic we obtain a linear network $\tilde{\mathbf{w}}(t)^{\top}\mathbf{x}$ that regresses to labels $\sigma^{-1}(y)$. Therefore, we can apply Theorem 6.1 to obtain the result. In case $a(t) < 0$ for all $t$, we write $a(t)\sigma\left(\mathbf{w}(t)^{\top}\mathbf{x}\right) = -\sigma\left(-a(t)\mathbf{w}(t)^{\top}\mathbf{x}\right) = -\sigma\left(-\tilde{\mathbf{w}}(t)^{\top}\mathbf{x}\right)$. In this case the linear network $\tilde{\mathbf{w}}(t)^{\top}\mathbf{x}$ regresses to labels $-\sigma^{-1}(-y)$ and again we apply Theorem 6.1, which concludes the proof. □

Recently, Vardi & Shamir (2020) proved a negative result for depth 2 single ReLU neuron with the square loss. They showed that it is impossible to characterize the implicit regularization by any explicit function of the model parameters. We note that Theorem 8.1 does not contradict the result of Vardi & Shamir (2020) since it does not include the ReLU case ($\rho = 0$), where $\sigma$ is not strictly monotonic.

## 9. Numerical Simulations Details

In order to study the effect of initialization over the implicit bias of gradient flow, we follow the sparse regression problem suggested by Woodworth et al. (2020), where $\mathbf{x}^{(1)}, ..., \mathbf{x}^{(N)} \sim \mathcal{N}(0, I)$ and $y^{(n)} \sim \mathcal{N}(\langle \beta^*, \mathbf{x}^{(n)} \rangle, 0.01)$ and $\beta^*$ is $r^*$-sparse, with non-zero entries equal to $1/\sqrt{r^*}$. For every $N \leq d$, gradient flow will generally reach a zero training error solution, however not all of these solutions will be the same, allowing us to explore the effect of initialization over the implicit bias.

This setting was also shown by Woodworth et al. (2020) to be tightly linked to generalization in certain settings, since the minimal $\ell_1$ solution has a sample complexity of $N = \Omega(r^* \log d)$, while the minimal $\ell_2$ solution has a much higher sample complexity of $N = \Omega(d)$. Throughout all the simulations, unless stated otherwise, we have used $N = 100$, $d = 1000$, $r^* = 5$.

See Figure 7.1 for results, and Section 7.1 for discussion.

## 10. Conclusion

Understanding generalization in deep learning requires understanding the implicit biases of gradient methods. Much remains to be understood about these, and even a complete understanding of linear networks is yet to be attained. Here we make progress in this direction by developing a new technique, which we apply to derive biases for diagonal and fully connected networks with independently trained layers (i.e., without shared weights). This allows us to study the effect of the initialization shape on implicit bias.

From a practical perspective it has been previously observed that balance plays an important role in initialization. For example, Xavier initialization (Glorot & Bengio, 2010) is roughly balanced by construction, and our results now provide additional theoretical support for the practical utility of this commonly used approach. We believe it is likely that further theoretical results like those presented here, can lead to improved initialization methods that lead to more effective convergence to rich regime solutions.

## 11. Acknowledgements

This research is supported by the European Research Council (ERC) under the European Unions Horizon 2020 research and innovation programme (grant ERC HOLI 819080) and by the Yandex Initiative in Machine Learning at Tel Aviv University. The research of DS is supported by the Israel Science Foundation (grant No. 31/1031), by the Israel Inovation Authority (the Avatar Consortium), and by the Taub Foundation. The research of BW is supported by a Google Research PhD Fellowship.

## References

Amid, E. and Warmuth, M. K. Winnowing with gradient descent. In *Proceedings of Thirty Third Conference on Learning Theory*, pp. 163–182, 2020a.

Amid, E. and Warmuth, M. K. K. Reparameterizing mirror descent as gradient descent. In *Advances in Neural Information Processing Systems*, volume 33, pp. 8420–8429, 2020b.

Chizat, L. and Bach, F. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on Learning Theory*, pp. 1305–1338, 2020.

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

Du, S. S., Hu, W., and Lee, J. D. Algorithmic regularization in learning deep homogeneous models: Layers are automatically balanced. In *NeurIPS*, 2018.

Du, S. S., Zhai, X., Poczos, B., and Singh, A. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2019.

Glorot, X. and Bengio, Y. Understanding the difficulty of training deep feedforward neural networks. In *AISTATS*, 2010.

Gunasekar, S., Woodworth, B. E., Bhojanapalli, S., Neyshabur, B., and Srebro, N. Implicit regularization in matrix factorization. In *Advances in Neural Information Processing Systems*, pp. 6151–6159, 2017.

Gunasekar, S., Lee, J. D., Soudry, D., and Srebro, N. Characterizing implicit bias in terms of optimization geometry. In *International Conference on Machine Learning*, pp. 1827–1836, 2018a.

Gunasekar, S., Lee, J. D., Soudry, D., and Srebro, N. Implicit bias of gradient descent on linear convolutional networks. In *Advances in Neural Information Processing Systems*, pp. 9461–9471, 2018b.

Gunasekar, S., Woodworth, B., and Srebro, N. Mirrorless mirror descent: A more natural discretization of riemannian gradient flow, 2020.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in Neural Information Processing Systems*, pp. 8571–8580, 2018.

Ji, Z. and Telgarsky, M. J. Gradient descent aligns the layers of deep linear networks. In *International Conference on Learning Representations*, 2019.

Li, Y., Ma, T., and Zhang, H. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. In *Conference On Learning Theory*, pp. 2–47, 2018.

Li, Z., Luo, Y., and Lyu, K. Towards resolving the implicit bias of gradient descent for matrix factorization: Greedy low-rank learning. In *International Conference on Learning Representations*, 2021.

Lyu, K. and Li, J. Gradient descent maximizes the margin of homogeneous neural networks. *ArXiv*, abs/1906.05890, 2020a.

Lyu, K. and Li, J. Gradient descent maximizes the margin of homogeneous neural networks. In *International Conference on Learning Representations*, 2020b.

Moroshko, E., Woodworth, B. E., Gunasekar, S., Lee, J. D., Srebro, N., and Soudry, D. Implicit bias in deep linear classification: Initialization scale vs training accuracy. In *Advances in Neural Information Processing Systems*, volume 33, pp. 22182–22193, 2020.

Nacson, M. S., Gunasekar, S., Lee, J., Srebro, N., and Soudry, D. Lexicographic and depth-sensitive margins in homogeneous and non-homogeneous deep models. In *International Conference on Machine Learning*, pp. 4683–4692, 2019.

Nguyen, Q. On the proof of global convergence of gradient descent for deep relu networks with linear widths. *arXiv preprint arXiv:2101.09612*, 2021.

Razin, N. and Cohen, N. Implicit regularization in deep learning may not be explainable by norms. *arXiv preprint arXiv:2005.06398*, 2020.

Vardi, G. and Shamir, O. Implicit regularization in relu networks with the square loss. *ArXiv*, abs/2012.05156, 2020.

Vaskevicius, T., Kanade, V., and Rebeschini, P. Implicit regularization for optimal sparse recovery. In *Advances in Neural Information Processing Systems*, volume 32, pp. 2972–2983, 2019.

Woodworth, B., Gunasekar, S., Lee, J. D., Moroshko, E., Savarese, P., Golan, I., Soudry, D., and Srebro, N. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*, pp. 3635–3673, 2020.

Yun, C., Krishnan, S., and Mobahi, H. A unifying view on implicit bias in training linear neural networks. In *International Conference on Learning Representations*, 2021.