Implicit Bias of SGD for Diagonal Linear Networks: a Provable Benefit of Stochasticity

Anonymous Author(s)
Affiliation
Address
email

Abstract

Understanding the implicit bias of training algorithms is of crucial importance in order to explain the success of overparametrised neural networks. In this paper, we study the dynamics of stochastic gradient descent over diagonal linear networks through its continuous time version, namely stochastic gradient flow. We explicitly characterise the solution chosen by the stochastic flow and prove that it always enjoys better generalisation properties than that of gradient flow. Quite surprisingly, we show that the convergence speed of the training loss controls the magnitude of the biasing effect: the slower the convergence, the better the bias. To fully complete our analysis, we provide convergence guarantees for the dynamics. We also give experimental results which support our theoretical claims. Our findings highlight the fact that structured noise can induce better generalisation and they help explain the greater performances observed in practice of stochastic gradient descent over gradient descent.

1 Introduction

Understanding the performance of neural networks is certainly one of the most thrilling challenges for the current machine learning community. From the theoretical point of view, progress has been made in several directions: we have a better functional analysis description of neural networks [2] and we steadily understand the convergence of training algorithms [26, 6] as well as the role of initialisation [17, 8]. Yet there remain many unanswered questions. One of them is why the training algorithms which are used in practice converge to solutions which generalise well, and this with very little use of explicit regularisation [35].

To understand this phenomenon, the elegant concept of *implicit bias* has emerged: if over-fitting is benign, it must be because the optimisation procedure converges towards particular global minima which have good generalising properties. Though no explicit regularisation is added, the algorithm is implicitly selecting a particular solution: this is referred to as the implicit bias of the training procedure. The implicit regularisation of several algorithms have now been shown, the simplest and most emblematic being that of gradient descent and stochastic gradient descent in the least-squares framework: they both converge towards the global solution which has the lowest squared distance from the initialisation. For logistic regression on separable data, Soudry et al. show in the seminal paper [27] that gradient descent selects the max-margin classifier. This type of result have been extended to neural networks and other frameworks. Overall, characterising the implicit bias of gradient methods has come down to unveiling a mirror-descent like structure to the algorithm.

While mostly all of these results focus on gradient descent, it must be pointed out this algorithm is not used in practice for neural networks since it does not lead to solutions which generalise well [20]. Instead, results on stochastic gradient descent, which is widely used and shows impressive results, are still missing or unsatisfactory. This has certainly to do with the fact that grasping the nature of

Submitted to 35th Conference on Neural Information Processing Systems (NeurIPS 2021). Do not distribute.

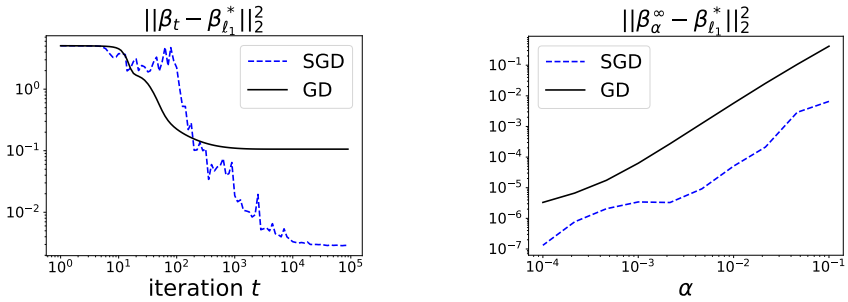

Figure 1: Sparse regression with $n = 40$, $d = 100$, $\|\beta^*_{\ell_1}\|_0 = 5$, $x_i \sim \mathcal{N}(0, I)$ $y_i = x_i^\top \beta^*_{\ell_1}$. Left: for initialisation scale $\alpha = 0.05$, SGD converges towards a solution which generalises better than GD. Right: for different values of the initialisation scale α, the solution recovered by SGD has better validation error than that of GD. The sparsifying effect due to their implicit biases differ by more than an order of magnitude. See section 5.1 for the precise experimental setup.

the noise induced by the stochasticity of the algorithm is particularly hard: it mixes properties from the model's architecture, the data's distribution and the loss. In our work we start to answer to the following fundamental questions for simplified neural networks: do SGD's and GD's implicit bias differ? What is the role of SGD's noise over the algorithm's implicit bias?

The simplified neural networks which we consider are linear diagonal neural networks and are popular since they already enable to grasp the complexity of more general networks. Indeed, they highlight important aspects of the theoretical concerns of modern machine learning: the neural tangent kernel regime, the roles of over-parametrisation, of the initialisation and of the step size. For a regression problem where we assume the existence of an interpolating solution, we study stochastic gradient descent through its continuous version, namely stochastic gradient flow (SGF). Though the continuous modelling of SGD has not yet led to many fruitful results compared to the well studied gradient flow, we believe it is because capturing the essence of the stochastic noise is particularly difficult. It has generally be done in a non realistic and over simplified manner, such as considering constant and isotropic noise. In our work, we attach peculiar attention in adequately modelling the noise. Tools from Itô calculus can then be leverage in order to derive exact formulas, quantitative bounds and interesting interpretations for our problem.

1.1 Main contributions and paper organization.

In Section 2, we start by introducing the setup of our problem as well as the continuous modelisation of stochastic gradient descent. Then, in Section 3, we give our main result on the implicit bias of the stochastic gradient flow. We illustrate our central result in Figure 1 and informally formulate it here:

Theorem 1 (Informal)**.** *Stochastic gradient flow over diagonal linear networks converges with high probability to a zero-loss solution which enjoys better generalisation properties than the one obtained by gradient flow. Furthermore, the speed of convergence of the training loss controls the magnitude of the biasing effect: the slower the convergence, the better the bias.*

Note that additionally to the implicit bias effect of SGF, unlike previous works [11, 32], we need not assume convergence of the iterates towards a zero-loss solution ; instead, we prove it with high probability. To accomplish this, we leverage, in Section 4, the fact that the iterates follow a stochastic continuous mirror descent with a time-changing potential. We support our results experimentally and validate our model in Section 5.

1.2 Related work

As recalled, implicit bias has a recent history that has been initiated by the seminal work [27] on max-margin classification with log-loss for a linear set-up with separable data. This work has been extended to other architectures, *e.g.* multiplicative parametrisations [11], linear networks [19] and more general homogeneous neural networks [24, 7]. In [32] the authors show that the scale of the initialisation leads to an interpolation between the neural tangent kernel regime [17, 8] (which is

a linear regression on fixed features) leading to ℓ_2 minimum norm solutions and the rich regimes leading to ℓ_1 minimum norm solutions. Note that these works focus on full batch gradient descent (or flow) and are deeply linked to mirror descent.

While the links between SGD's stochasticity and generalization have been looked into in numerous works [25, 18, 13, 15, 21], no such explicit characterisation of implicit regularisation have ever been given. It has been empirically observed that SGD often outputs models which generalize better than GD [20, 18, 13]. One suggested explanation is that SGD is prone to pick flatter solutions than GD and that low generalization solutions are correlated with sharp minima, i.e., with strong curvature, while good generalization solutions are correlated with flat minima, i.e., with low curvature[14, 20]. This idea has been further investigated by adopting a random walk on random landscape modeling [15], by suggesting that SGD's noise is smoothing the loss landscape, thus eliminating the sharp minima [21], by considering a dynamical stability perspective [34] or by interpreting SGD as a diffusion process [13, 18, 4]. Recently label-noise has been shown to influence the implicit bias of SGD, by biasing the solution towards the origin for quadratically-parameterized models [12] or by implicitly regularising the expected squared norm of the gradient of the model with respect to the weights [3]. Thus, if the notion of implicit bias of GD is fairly well understood both in the cases of regression and classification, it remains unclear for SGD, and its explicit characterisation is missing.

The linear diagonal neural networks we are considering have also been studied in the case of gradient descent [29] and stochastic gradient descent with label noise [12]. In both cases the authors show that this model has the ability to implicitly bias the training procedure to help retrieve a sparse predictor. The link between gradient descent and mirror descent for this model has been initiated by [10] and further exploited by the same author in [33, 30] for its sparse inducing property.

Contrary to the deterministic case, the modelling of stochastic gradient descent as a stochastic differential equation is quite recent, see [25, 18]. However, early attempts often suffer from the drawback that they model the noise as a fixed Gaussian. On the contrary, state dependant noise has now become the legitimate manner for modelling SGD as a stochastic gradient flow and it is shown in [23] that it can be done consistently. Yet, noise modelling still remains the principal issue [31] as it influences largely the behaviour of the dynamics [4, 5].

1.3 Notations

For input data $(x_1, \dots, x_n) \in (\mathbb{R}^d)^n$ and output $(y_1, \dots, y_n) \in \mathbb{R}^n$, we denote respectively $X \in \mathbb{R}^{n \times d}$ the design matrix whose i-th row is feature $x_i \in \mathbb{R}^d$ and $y \in \mathbb{R}^n$ the vector of outputs. $\mathbb{R}_+^*$ denotes the set of positive real numbers. For $p = 1, 2$, the ℓ_p-norm of $x \in \mathbb{R}^d$ is $\|x\|_p^p = \sum_i^d |x_i|^p$. The operations $\odot$ will stand for coordinate-wise product between vector: $[u \odot v]_i = u_i v_i$. For $p \in \mathbb{N}^*$, we also define $u^p := u \odot \dots \odot u$, the p times product of u with itself. All inequalities between vectors should be understood value by value. For $f, g \in \mathbb{R}$, the existence of $C > 0$ such that $f \leq Cg$ and $Cg \leq f$ will be denoted $f \leq O(g)$ and $\Omega(g) \leq f$ respectively. We shall use the symbole $\widetilde{O}$ when this is true up to $\log$ factors. The indexed vector β^* will stand for any β interpolating the data, i.e. any vector in the affine space $\{\beta \in \mathbb{R}^d \ s.t. \ X\beta = Y\}$ of dimension at least $d - n$. Out of all these, let $\beta^*_{\ell_1} = \underset{\beta \in \mathbb{R}^d \text{ s.t. } X\beta = y}{\arg\min} \|\beta\|_1$. For z any vector, z_∞ or z^∞ will always designate of $\lim_{t \to \infty} z_t$.

2 Setup and preliminaries

2.1 Architecture and algorithm.

Overparametrised noiseless regression. We consider a linear regression problem with outputs $(y_1, \dots, y_n) \in \mathbb{R}^n$ and inputs $(x_1, \dots, x_n) \in (\mathbb{R}^d)^n$. We study an overparametrised setting and assume that there exists at least one interpolating parameter $\beta^* \in \mathbb{R}^d$ which perfectly fits the training set, i.e. $y_i = \langle \beta^*, x_i \rangle$ for all $1 \leq i \leq n$. We parametrise the regression vector β as β_w with $w \in \mathbb{R}^p$. We will see that though in the end our final models $x \mapsto \langle \beta_w, x \rangle$ are classical linear models whatever the parametrisation $w \mapsto \beta_w$, the choice of this parametrisation has crucial consequences on the solution recovered by the learning algorithms. We study the quadratic loss and the overall loss is

written as:

$$L(w) = L(\beta_w) := \frac{1}{4n}\sum_{i=1}^{n}(\langle \beta_w, x_i\rangle - y_i)^2 = \frac{1}{4n}\sum_{i=1}^{n}\langle \beta_w - \beta^*, x_i\rangle^2,$$

where by abuse of notation we use $L(w) = L(\beta_w)$.

2-layer diagonal linear network. The simplest parametrisation of β_w is to consider $\beta_w = w$ which corresponds to the classical least-squares framework. It is well known that in this case, many first order methods (GD, SGD, with and without momentum) will converge towards the same solution: we say that they have the same implicit bias. This is experimentally not the case for neural networks where SGD has been shown to lead to solutions which have better generalisation properties compared to GD [20]. To theoretically confirm this observation, we study the simplest non-linear parametrisation : the quadratic parametrisation $\beta_w = w_+^2 - w_-^2$ with $w = [w_+, w_-]^\top \in \mathbb{R}^{2d}$. We point out that it has the particularity of being 2-positive homogeneous and that it is equivalent to the parametrisation $\beta_{u,v} = u \odot v$ with $u, v \in \mathbb{R}^d$. It should be thought of a simplified linear network of depth 2 and we refer to [32, Section 4] for more details. We consider two weight vectors w_+ and w_- (and not $\beta_w = w^2$) in order to ensure that our final linear predictor parameter β_w can take negative values. For the sake of completeness, the study of diagonal linear networks of arbitrary depth $p \geq 3$ is done in the Appendix. Also note that additionally to being a toy neural model, it has received recent attention on its own for its practical ability to induce sparsity [29, 30, 12] or solve phase retrieval problems [33].

Stochastic Gradient Descent. Our loss now rewrites as: $L(w) = \frac{1}{4n}\sum_{i=1}^{n}\langle w_+^2 - w_-^2 - \beta^*, x_i\rangle^2$. Note that despite its simplicity, this loss is non convex and its minimisation is non trivial. The algorithm we shall consider is the well known SGD algorithm, where for a step size $\gamma > 0$:

$$\begin{aligned} w_{t+1,+} &= w_{t,+} - \gamma\langle \beta_w - \beta^*, x_{i_t}\rangle\, x_{i_t} \odot w_{t,+} \\ w_{t+1,-} &= w_{t,-} + \gamma\langle \beta_w - \beta^*, x_{i_t}\rangle\, x_{i_t} \odot w_{t,-} \end{aligned} \qquad \text{where } i_t \sim \mathrm{unif}(1, n). \tag{1}$$

It is convenient to rewrite this recursion as

$$w_{t+1,\pm} = w_{t,\pm} - \gamma\nabla_{w_\pm} L(w_t) \pm \gamma\,\mathrm{diag}(w_{t,\pm})X^\top \xi_{i_t}(w_t), \tag{2}$$

where $\xi_{i_t}(\beta) = -\big(\langle \beta - \beta^*, x_{i_t}\rangle \mathbf{e}_{i_t} - \mathbb{E}_{i_t}\big[\langle \beta - \beta^*, x_{i_t}\rangle \mathbf{e}_{i_t}\big]\big) \in \mathbb{R}^n$ is a zero-mean *multiplicative* noise which vanishes at any global optimum ($\mathbf{e}_i$ denotes the ith element of the canonical basis). We point out that all the results we shall give hold for any initialisation such that $w_{t=0,+} = w_{t=0,-} \in \mathbb{R}^d$ under which we have that $\beta_{w_{t=0}} = 0$. To try and understand under what conditions the SGD procedure converges and towards which point it does we shall consider its continuous counterpart which has the advantage of leading to clean and intuitive calculations.

2.2 Stochastic gradient flow

Continuous time modelling of sequential processes offer a large set of tools, such as derivation, which come in helpful to understand the dynamics of the processes. This has led to a large part of the recent literature to consider continuous gradient flow in order to try and understand the behaviour of gradient descent on complicated architectures such as neural nets. However, the continuous time modelling of stochastic gradient descent is more challenging: it requires to add on top of the gradient flow a diffusion term whose covariance matches the one of SGD. Hence, it is fundamental to understand its structure and scale.

Understanding the noise's structure. As seen in equation (2), evaluated at $w_\pm$, the stochastic noise $\gamma\,\mathrm{diag}(w_\pm)X^\top\xi_{i_t}(w)$ has two main characteristics which we want to preserve:

- It belongs to $\mathrm{span}(w_\pm \odot x_1, \ldots, w_\pm \odot x_n)$
- It has covariance $\Sigma_{\text{SGD}}(w_\pm) := \gamma^2\,\mathrm{diag}(w_\pm)X^\top \mathrm{Cov}_{i_t}(\xi_{i_t})X\,\mathrm{diag}(w_\pm)$

It remains to understand the structure of the covariance of ξ_{i_t} which has the following closed form: $\mathrm{Cov}_{i_t}(\xi_{i_t}(\beta)) = \frac{1}{n}\,\mathrm{diag}(\langle \beta - \beta^*, x_i\rangle^2)_{1\leq i\leq n} - \frac{1}{n^2}\,\mathrm{diag}(\langle \beta - \beta^*, x_i\rangle\langle \beta - \beta^*, x_j\rangle)_{1\leq i,j\leq n}$. We identify the two following key facts: (i) it is diagonal at the leading n^{-1} order and (ii) its trace is linked

to the loss as $\mathrm{Var}_{i_t}(\|\xi_{i_t}(w)\|_2) = \frac{4}{n}L(w) + O(\frac{1}{n^2})$. This leads us in modelling $\xi_{i_t}(\beta)$'s covariance matrix as $\frac{4}{n}L(\beta)I_n$ as it preserves the two key characteristics [1]. Finally this brings us to consider the following modeling of the overall noise's structure: $\Sigma_{\text{SGD}}(w_\pm) \cong \frac{1}{n}\gamma^2 L(w)[\mathrm{diag}(w_\pm)X^\top]^{\otimes 2}$.

Stochastic differentiable equation modelling. Consider the following stochastic gradient flow:

$$\begin{aligned} \mathrm{d}w_{t,+} &= -\nabla_{w_+}L(w_t)\,\mathrm{d}t + \sqrt{\gamma n^{-1}L(w_t)}\; w_{t,+} \odot [X^\top \mathrm{d}B_t] \\ \mathrm{d}w_{t,-} &= -\nabla_{w_-}L(w_t)\,\mathrm{d}t - \sqrt{\gamma n^{-1}L(w_t)}\; w_{t,-} \odot [X^\top \mathrm{d}B_t], \end{aligned} \tag{3}$$

where $\mathrm{d}B_t$ is a standard $\mathbb{R}^n$ Brownian motion. The SDE is a perturbed gradient flow with a diffusion term that is defined such that its Euler discretisation with step size γ leads to a Markov Chain whose covariance exactly matches SGD's noise covariance $\Sigma_{\text{SGD}}(w_\pm)$. We refer to [23] or any textbook on numerical approximation of SDE [22] for the technical details regarding consistency of such a procedure in the limit of small step sizes. This stochastic differential equation is the starting point of the analysis.

3 The implicit bias of the stochastic gradient flow

Implicit bias and hyperbolic entropy. To understand the relevance of our result and how stochasticity induces a preferable bias, we start by recalling some known results for gradient flow. In [32] it is shown, assuming global convergence, that the solution selected by the gradient flow initialized at $\alpha \in \mathbb{R}^d$ and denoted β_α^∞ solves a constrained optimisation problem involving the *hyperbolic entropy* introduced by [10]:

$$\beta_\alpha^\infty = \underset{\beta\in\mathbb{R}^d \text{ s.t. } X\beta=y}{\arg\min}\ \phi_\alpha(\beta) := \frac{1}{4}\Big[\sum_{i=1}^d \beta_i \mathrm{arcsinh}(\frac{\beta_i}{2\alpha_i^2}) - \sqrt{\beta_i^2 + 4\alpha_i^4}\Big], \tag{4}$$

Though the hyperbolic entropy function has a non-trivial expression, its principal characteristic is that it interpolates between the ℓ_1 and the ℓ_2 norms according to the scale of α. More precisely for $\alpha \in \mathbb{R}$ [2]: $\phi_\alpha(\beta) \underset{\alpha\to 0}{\sim} \frac{1}{2}\ln\left(\frac{1}{\alpha}\right)\|\beta\|_1$ and $\phi_\alpha(\beta) \underset{\alpha\to+\infty}{=} 2\alpha^2 + \frac{1}{4\alpha^2}\|\beta\|_2^2 + o(\alpha^{-2})$. We refer to [32, Theorem 2] for more details on the asymptotic analysis. The implicit optimisation problem (4) therefore highlights the fact that the initialisation scale of the weights controls the shape of the recovered solution. Small initialisations lead to low ℓ_1-type solutions which are known to induce good generalisation properties, this is what is often referred to as the *rich regime*. Large initialisations lead to low ℓ_2-type solutions, this is referred to as the *kernel regime* or *lazy regime* in which the weights move only very slightly. The dynamics of the gradient flow are then very similar to the one of kernel linear regression with the kernel depending on the initialization ([17, 8]). Overall, to retrieve a sparse solution, one should initialise with the smallest α possible. However, as is clearly explained in [32], it is important to stress out that there is generalisation - optimisation tradeoff: the point $w = 0$ happens to be a saddle point for the loss and a smaller α will lead to a longer training time.

Main result. In our main theorem we show that, for an initialisation scale α, the stochasticity of SGF biases the flow towards solutions which still minimise the hyperbolic entropy. However, what is remarkable is that it does so with an effective parameter α_∞ which is strictly smaller than α. The recovered solution therefore minimises an optimisation problem which has better sparsity inducing properties than that of gradient flow.

Theorem 1. *For $p \le \frac{1}{2}$ and $w_{0,\pm} = \alpha \in (\mathbb{R}_+^*)^d$, let $(w_t)_{t\ge 0}$ follow the stochastic gradient flow* (3) *with step size $\gamma \le \widetilde{O}\big(\max\{\lambda_{max}\|\beta_{\ell_1}^*\|_1, \|\alpha\|_2^2\}^{-1}\big)$ where λ_{max} is the largest eigenvalue of $X^\top X/n$. Then, with probability at least $1-p$:*

- *$(\beta_t)_{t\ge 0}$ converges towards a zero-training error solution β_α^∞*
- *the solution β_α^∞ satisfies*

$$\beta_\alpha^\infty = \underset{\beta\in\mathbb{R}^d \text{ s.t. } X\beta=y}{\arg\min}\ \phi_{\alpha_\infty}(\beta) \quad \textit{where} \quad \alpha_\infty = \alpha \odot \exp\left(-\frac{\gamma}{2}\,\mathrm{diag}\left(\frac{X^\top X}{n}\right)\int_0^{+\infty} L(\beta_s)\,\mathrm{d}s\right). \tag{5}$$

[1] the general case is discussed in the Appendix

[2] If $\alpha \in \mathbb{R}$ we consider the abuse of notation $\phi_\alpha := \phi_{\alpha\mathbf{1}}$.

Our theorem is three-fold: with high probability and for an explicit choice of constant step size γ, (i) the flow $(\beta_t)_{t\geq 0}$ converges, (ii) β_α^∞ is an interpolating solution, i.e. $X\beta_\alpha^\infty = y$, (iii) this solution minimises the hyperbolic entropy problem with a parameter that depends on the dynamics. We illustrate these results in Figure 2. Now let us comment further the theorem.

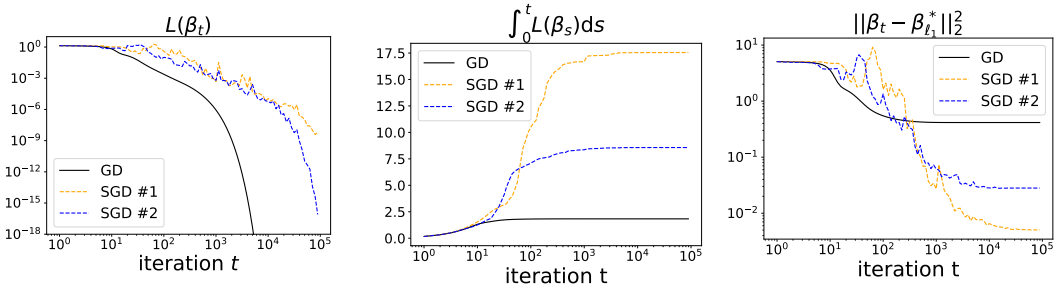

Figure 2: Sparse regression (see section 5.1 for the detailed experimental setting). Both SGD and GD are initialised at $\alpha = 0.1$. 2 different runs of SGD over the training set are performed, they differ due to the inner stochasticity of the algorithm. Left: GD and SGD both converge towards a global minimum. Middle and right: for two different trajectories of SGD, the higher the value of the loss integral at convergence, the better the validation error. In both cases SGD converges towards a solution which generalises better. This figure illustrates Theorem 1.

Beneficial implicit bias through effective initialisation. The most remarkable aspect of our result is that the recovered solution β_α^∞ minimises the same potential as for gradient flow but with an *effective parameter* α_∞ which is strictly smaller than α. Hence, the hyperbolic entropy is closer to the ℓ_1 norm compared to the deterministic case, proving a systematic benefit of stochasticity. Note that this effective parameter is random and controlled by the loss integral $\int_0^{+\infty} L(\beta_s)\,\mathrm{d}s$: the higher the integral, the smaller the effective initialisation scale. In other words and quite surprisingly, the slower the loss converges to 0, the "richer" the implicit bias. However it must be kept in mind that, as explained in [32], there is a tension between generalisation and optimisation: a longer training time might improve generalisation but comes at the cost of... a longer training time. However it is clear experimentally that SGD systematically largely wins the trade-off over GD (see Figure 2). Yet, problem 5 tells us that the implicit bias of SGD initialised at α acts as if we run GD initialised at α_∞. Note that the minimisation problem (5) only makes sense *a posteriori* since the quantity α_∞ depends on the whole stochastic trajectory. We finally point out that the value of this integral still depends on γ and α, this dependency will be made clear in the next section (see Proposition 3).

Kernel regime. Though it is less our focus, our result still holds as $\alpha \to +\infty$ which corresponds to the kernel regime. We show in the Appendix that in this regime, $\int_0^{+\infty} L(\beta_s)ds \underset{\alpha\to\infty}{\to} 0$ and hence SGF and GF converge towards the same solution. This is expected since in the NTK regime, the iterates follow a kernel linear regression for which the bias of SGF and GF are the same.

Step-size. Note that our convergence result holds for a constant step size. This is not illogical since in our overparametrised setting, the noise vanishes at the optimum (see [28] for a convergence result in the overparametrised least-squares setup). The explicit formula for the γ upper bound is $\gamma \leq \left(30 \ln\left(\frac{2}{p}\right)\lambda_{\max}(\frac{X^\top X}{n}) \max\left\{\|\beta^*_{\ell_1}\|_1 \ln\left(\sqrt{2}\frac{\|\beta^*_{\ell_1}\|_1}{\min_i \alpha_i^2}\right), \|\alpha\|_2^2\right\}\right)^{-1}$. It has a classical dependence on $\lambda_{\max}(X^\top X/n)$ which can be computed, but also on the unknown value of $\|\beta^*_{\ell_1}\|_1$. However in practice we choose the highest value of γ for which the iterates converge. Note that in practice the weights are often initialised such that $\|\alpha\|_2^2$ is roughly equal to 1 and hence it is sensible to consider $\|\alpha\|_2^2 < \|\beta^*_{\ell_1}\|_1$. In the explicit bound, there is a $\ln\left(\|\beta^*\|_1/\min_i \alpha_i^2\right)^{-1}$ factor, we believe that it is an artefact of our analysis and could be removed. It is hence best to think of the upperbound on γ to simply be $\gamma \leq O(\frac{1}{\lambda_{\max}\|\beta^*\|_1})$.

Convergence and proof sketch. Let us put emphasis on the fact that since we deal with a non-convex problem, neither convergence nor convergence towards a global minimum are obvious. In

most of similar works, convergence of the iterates is assumed [32, 11]. In fact, the hardest and most technical part of our result is to show the convergence of the flow with high probability: once the convergence is shown, describing the minimisation problem β_α^∞ verifies is straightforward. In the following section we give several properties which constitute the major keys of the theorem's proof.

4 Links with mirror descent

The aim of this section is to show that the sequence $(\beta_t)_{t\geq 0}$ follows a stochastic version of continuous mirror descent with a time dependent mirror. From this crucial property, we show how the convergence and implicit bias characterisation follow. Finally, as it is one of the central objects of our main theorem, we give an estimation of $\int_0^\infty L(\beta_s)\,\mathrm{d}s$.

4.1 Stochastic continuous mirror descent with time-varying potential

We start by recalling known results on the link between implicit bias and mirror descent. We recall also convergence guarantees for mirror descent dynamics.

Mirror descent: convergence and implicit bias. For any $\beta_0 \in \mathbb{R}^d$ and convex potential function Ψ, consider the mirror descent flow $(\beta_t)_t$ which corresponds to $\mathrm{d}\nabla\Psi(\beta_t) = -\nabla L(\beta_t)\mathrm{d}t$. Though the convergence of the loss to 0 is straightforward, showing the convergence of the iterates requires more work and is shown in [9, Theorem 1] for strongly convex potentials. Yet, once convergence is shown, deriving the implicit minimisation problem is straightforward. We recall the reasoning here (see Section 3 of [1] for more details): integrating the flow yields $\nabla\Psi(\beta\infty) - \nabla\Psi(\beta_0) = -\int_0^\infty \nabla L(\beta_s)\,\mathrm{d}s = -4X^\top \int_0^\infty X(\beta_s - \beta^\infty)\,\mathrm{d}s \in \mathrm{span}(X)$. This condition, along with the fact that $X\beta_\infty = y$ exactly corresponds to the KKT conditions of the following problem:

$$\beta_\infty = \underset{\beta\in\mathbb{R}^d \text{ s.t. } X\beta=y}{\arg\min} D_\Psi(\beta, \beta_0), \tag{6}$$

where $D_\Psi(\beta, \beta_0) = \Psi(\beta) - \Psi(\beta_0) - \langle \nabla\Psi(\beta_0), \beta - \beta_0\rangle$ is the Bregman divergence w.r.t. Ψ.

Link with our model. It turns out that these general observations on mirror descent apply to our framework when $(w_t)_t$ follows the gradient flow $\mathrm{d}w_{t,\pm} = -\nabla_{w_\pm} L(w_t)\odot w_{t,\pm}\,\mathrm{d}t$. Indeed it has been shown in [32] that the corresponding iterates $\beta_t = w_{t,+}^2 - w_{t,-}^2$ follow a mirror descent with potential ϕ_α defined in Eq.(4). Therefore we can apply the previous remarks to obtain the convergence towards an interpolator[3], as well as the associated implicit minimisation problem which in our case can be rewritten as $\beta_\infty = \underset{\beta\in\mathbb{R}^d \text{ s.t. } X\beta=y}{\arg\min} \phi_\alpha(\beta)$ since $\nabla\phi_\alpha(\beta_0 = 0) = 0$.

Stochastic Mirror descent with a time varying potential. To address our problem where $(w_t)_t$ follows a stochastic gradient flow instead of a gradient flow, it is natural, as in the deterministic framework, to see what type of flow $(\beta_t)_t$ follows. Because of the noise we cannot hope to simply recover a classical mirror descent. However interestingly the next property shows that it follows a stochastic mirror-like descent with a geometry that depends on time.

Proposition 1. *Consider the iterates $(w_t)_{t\geq 0}$ issued from the stochastic gradient flow in Eq.*(3) *with initialisation $w_{0,\pm} = \alpha \in (\mathbb{R}_+^*)^d$. Then the corresponding flow $(\beta_t)_{t\geq 0}$ follows a "stochastic continuous mirror descent with time varying potential" defined by:*

$$\mathrm{d}\nabla\phi_{\alpha_t}(\beta_t) = -\nabla L(\beta_t)\,\mathrm{d}t + \frac{1}{2}\sqrt{\gamma n^{-1} L(\beta_t)}X^\top \mathrm{dB}_t, \tag{7}$$

where $\alpha_t = \alpha\odot\exp\left(-\frac{\gamma}{2}\,\mathrm{diag}\left(\frac{X^\top X}{n}\right)\int_0^t L(\beta_s)\,\mathrm{d}s\right)$ and ϕ_α is the hyperbolic entropy defined in (4).

Under this form we clearly see that the iterates $(\beta_t)_t$ follow a flow which closely resembles that of mirror descent but with two major differences: (i) the potential ϕ_{α_t} changes over time according to the random quantity $\int_0^t L(\beta_s)\,\mathrm{d}s$, (ii) the flow is perturbed by noise. We highlight the fact that viewing the dynamics this way has the major advantage of giving a clear roadmap for the proof

[3] In our case, ϕ_α is not strongly convex so a bit more work is necessary to show convergence of the iterates.

of Theorem 1: (1) we can adapt classical mirror-descent results to our framework and construct appropriate Lyapunov functions to prove the convergence of the flow with high probability to some interpolator β^∞, (2) we immediately recover the corresponding minimisation problem as in the deterministic case. Indeed, integrating Eq.(7) still yields $\nabla\phi_{\alpha_\infty} \in \mathrm{span}(X)$ which, along with $X\beta_\infty = y$, are the KKT conditions of the implicit minimisation problem (5). We emphasise the fact that the structure of the noise, belonging to $\mathrm{span}(X)$, is crucial in order to obtain this minimisation problem. This would for instance clearly not be true if we considered isotropic noise in the SDE modelling. This highlights the fact that every form of noise does not necessarily improve the implicit bias: the shape of the intrinsic SGD noise is of primal importance [12].

4.2 Convergence and control of $\int_0^\infty L(\beta_s)\,\mathrm{d}s$

Though it seems easy to derive the implicit minimisation problem (5) from the mirror-like structure of Eq.(7), it is necessary to ensure that the iterates converge towards an interpolator β^∞. This is the purpose of the following proposition.

Proposition 2 (Convergence of the iterates)**.** *Consider the iterates $(w_t)_{t\geq 0}$ issued from the stochastic gradient flow* (3)*, initialized at $w_{0,\pm} = \alpha \in (\mathbb{R}_+^*)^d$. For $p \leq \frac{1}{2}$ and γ such as in Theorem 1, then with probability at least $1-p$, the flow $(\beta_t)_t$ converges to an interpolating solution β_α^∞.*

The convergence of the iterates is technical and requires several intermediate results. We start by considering an appropriate Bregman-type stochastic function with a time-varying potential and show that it converges with high probability. Leveraging the fact that we are able to bound the iterates β_t, we are able to show that the limit of the function is in fact 0. Owing to the fact that the function we consider also controls the distance of β_t to a particular β^* we finally get that the iterates converge.

However for the objects (such as α_∞) and functions we introduce to be well defined, we need to guarantee the convergence of $\int_0^\infty L(\beta_s)\mathrm{d}s$. Besides, it is crucial to grasp the scale of this quantity since it gives the overall scale of α_∞. This is done in the following proposition where we lower and upper bound its value.

Proposition 3 (Control of the loss integral)**.** *Under the same setting as in proposition 2, we have:*

$$\Omega\Big(\|\beta^*_{\ell_1}\|_1 \ln\Big(\frac{\|\beta^*_{\ell_1}\|_1}{\alpha^2}\Big)\Big) \underset{\alpha\to 0}{\leqslant} \int_0^{+\infty} L(\beta_s)\,\mathrm{d}s \leqslant O\Big(\max\big\{\|\beta^*_{\ell_1}\|_1 \ln\Big(\frac{\|\beta^*_{\ell_1}\|_1}{\alpha^2}\Big), \|\alpha\|_2^2\big\}\Big).$$

We point out that the lower bound is given for small α's for simplicity but we provide in the Appendix a lower bound which holds for all α's. Note that when $\gamma = 0$, which corresponds to deterministic gradient flow, we can give the exact value for the integral: $\int_0^{+\infty} L(\beta_s)\,\mathrm{d}s = \phi_\alpha(\beta^\infty) - \phi_\alpha(0)$. This matches the scale of the bounds given in Proposition 3, hence showing tightness of the result.

Again, the proof of this proposition is technical and relies on considering appropriate Lyapunov functions which highly resemble to Bregman divergences, but which take into account the fact that the geometry changes over time. These overall decreasing Lyapunov's enable to bound the iterates as well as lower and upper bound the integral of the loss. The stochastic integrals which naturally appear are controlled with high probability using time-uniform concentration of martingales [16].

5 Experiments

5.1 Experimental setup for sparse regression

We consider the following sparse regression setup for our experiments. We consider $n = 40$, $d = 100$ and randomly generate a sparse model $\beta^*_{\ell_1}$ such that $\|\beta^*_{\ell_1}\|_0 = 5$. We generate the features as $x_i \sim \mathcal{N}(0, I)$ and the labels as $y_i = x_i^\top \beta^*_{\ell_1}$. SGD, GD and the SGF are always initialised using the same scale $\alpha > 0$ and it is specified each time. We use the same step size for GD and SGD and choose it to be the biggest as possible why still ensuring convergence. Note that since the true population covariance $\mathbb{E}[xx^\top]$ is equal to identity, the quantity $\|\beta_t - \beta^*_{\ell_1}\|_2^2$ corresponds to the validation error.

5.2 Validation of the SDE model

In this section, we present an experimental validation of our stochastic gradient flow model. Figure 3, for the same step size, we run: (i) the trajectory of gradient descent, (ii) 5 trajectories of stochastic

gradient descent, they correspond to different realisations of the uniform sampling over the data (iii) 5 trajectories of the stochastic gradient flow (its Euler discretisation with $\mathrm{d}t = \gamma/10$)) corresponding to different realisations of the Brownian. We clearly see (left) that the loss behaves similarly for SGD and SGF across time. We also see that the validation errors (right) of the iterates of SGD and SGF have very similar behaviours. This tends to validate our continuous modelling from section 2.2.

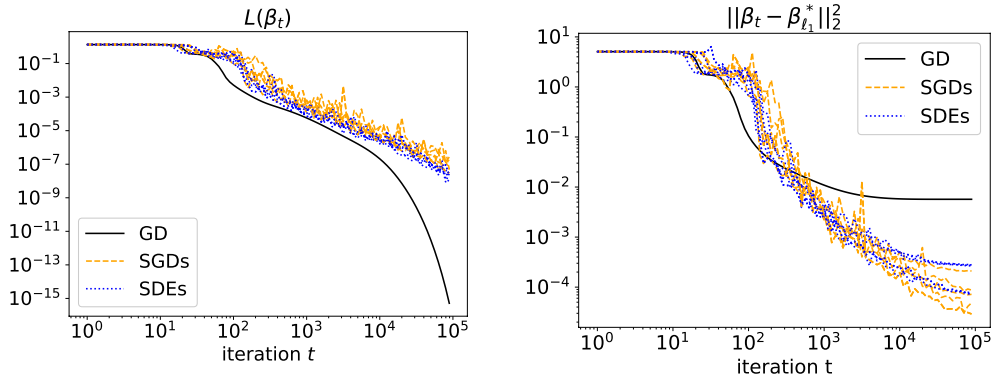

Figure 3: Sparse regression (see section 5.1 for the detailed experimental setup). Left and right: the training and the validation losses behave very similarly, corroborating our continuous modelling.

5.3 Doping the implicit bias with label noise

As largely discussed throughout the paper, the effect of the implicit bias is controlled by the convergence speed of the loss: the slower it converges, the sparser the selected solution will be. Hence the following question: can we leverage this knowledge to dope the implicit bias? We argue in this Section that the answer to this question is affirmative. Indeed, if we artificially inject some label noise on the data at time t, say $\Delta_t = \delta_t$ with probability $1/2$ and $-\delta_t$ otherwise, where $\delta_t > 0$, this would result in slowing down the convergence of the loss and hence improving the implicit biasing. This injected label noise perturbs SGD recursion as the following:

$$w_{t+1,\pm} = w_{t,\pm} \mp \gamma \left(\langle \beta_w - \beta^*, x_{i_t} \rangle + \Delta_t \right) \, x_{i_t} \odot w_{t,+} \, , \qquad \text{where } \; i_t \sim \mathrm{unif}(1, n). \tag{8}$$

And as in Section 2.2, we can derive its related stochastic gradient flow:

$$\mathrm{d}w_{t,\pm} = -\nabla_{w_\pm} L(w_t)\mathrm{d}t \pm \sqrt{\gamma n^{-1}(L(w_t) + \delta_t)} \, w_{t,+} \odot [X^\top \mathrm{d}B_t]. \tag{9}$$

Hence, assume that $(\delta_t)_t \in L^1(\mathbb{R}_+)$ and that the iterates converge, our implicit bias structure is preserved and changes only with respect to a slowed down loss: $\tilde{L}(w_t) := L(w_t) + \delta_t$. The effective initialization writes: $\tilde{\alpha}_\infty = \alpha \odot \exp\left(-\frac{\gamma}{2} \, \mathrm{diag}(\frac{X^\top X}{n}) \int_0^{+\infty} (L(\beta_s) + \delta_s) \, \mathrm{d}s \right)$. As expected, introducing label noise has a beneficial impact through increasing the new loss integral. However, label noise may prevent the stochastic gradient descent to converge if the step-sizes are too large, hence γ would be smaller in this case. Yet, putting these two things together it is experimentally clear that adding the right amount of label noise improves on the resulting bias as our experiments show in the Appendix.

6 Conclusion and Perspectives

In this paper, we have shown the benefit of using stochastic gradient descent over gradient descent for linear diagonal networks in terms of their implicit bias. Indeed, we prove that SGD acts as GD but initialised at a smaller scale: this induces a sparser finale iterate. This effect is controlled by the speed of convergence of the loss. Moreover, we prove the convergence of the flow and exhibit an interesting link with mirror descent. Fully understand this novel type of dynamics could help to grasp the implicit biasing properties of stochastic gradient descent in other frameworks. It is also natural to ask ourselves whether the integral of the loss also controls the difference of implicit regularisation for more general architectures. It would be interesting to analyse how this property adapts to $\log$ losses known to lead to max-margin solutions in classification.

References

[1] Shahar Azulay, Edward Moroshko, Mor Shpigel Nacson, Blake Woodworth, Nathan Srebro, Amir Globerson, and Daniel Soudry. On the implicit bias of initialization shape: Beyond infinitesimal mirror descent. *arXiv preprint arXiv:2102.09769*, 2021.

[2] Francis Bach. Breaking the curse of dimensionality with convex neural networks. *Journal of Machine Learning Research*, 18(19):1–53, 2017. URL `http://jmlr.org/papers/v18/14-546.html`.

[3] Guy Blanc, Neha Gupta, Gregory Valiant, and Paul Valiant. Implicit regularization for deep neural networks driven by an ornstein-uhlenbeck like process. In *Conference on learning theory*, pages 483–513. PMLR, 2020.

[4] Pratik Chaudhari and Stefano Soatto. Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks. In *2018 Information Theory and Applications Workshop (ITA)*, pages 1–10. IEEE, 2018.

[5] Xiang Cheng, Dong Yin, Peter Bartlett, and Michael Jordan. Stochastic gradient and Langevin processes. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1810–1819. PMLR, 13–18 Jul 2020.

[6] Lénaïc Chizat and Francis Bach. On the global convergence of gradient descent for over-parameterized models using optimal transport. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL `https://proceedings.neurips.cc/paper/2018/file/a1afc58c6ca9540d057299ec3016d726-Paper.pdf`.

[7] Lenaic Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In *Conference on Learning Theory*, pages 1305–1338. PMLR, 2020.

[8] Lénaïc Chizat, Edouard Oyallon, and Francis Bach. On lazy training in differentiable programming. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d''Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper/2019/file/ae614c557843b1df326cb29c57225459-Paper.pdf`.

[9] Thinh T Doan, Subhonmesh Bose, D Hoa Nguyen, and Carolyn L Beck. Convergence of the iterates in mirror descent methods. *IEEE control systems letters*, 3(1):114–119, 2018.

[10] Udaya Ghai, Elad Hazan, and Yoram Singer. Exponentiated gradient meets gradient descent. In Aryeh Kontorovich and Gergely Neu, editors, *Proceedings of the 31st International Conference on Algorithmic Learning Theory*, volume 117 of *Proceedings of Machine Learning Research*, pages 386–407, San Diego, California, USA, 08 Feb–11 Feb 2020. PMLR.

[11] Suriya Gunasekar, Jason Lee, Daniel Soudry, and Nathan Srebro. Characterizing implicit bias in terms of optimization geometry. In Jennifer Dy and Andreas Krause, editors, *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 1832–1841. PMLR, 10–15 Jul 2018. URL `http://proceedings.mlr.press/v80/gunasekar18a.html`.

[12] Jeff Z HaoChen, Colin Wei, Jason D Lee, and Tengyu Ma. Shape matters: Understanding the implicit bias of the noise covariance. *arXiv preprint arXiv:2006.08680*, 2020.

[13] Fengxiang He, Tongliang Liu, and Dacheng Tao. Control batch size and learning rate to generalize well: Theoretical and empirical evidence. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

[14] Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Comput.*, 9(1):1–42, jan 1997.

[15] Elad Hoffer, Itay Hubara, and Daniel Soudry. Train longer, generalize better: Closing the generalization gap in large batch training of neural networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 1729–1739, 2017.

[16] Steven R. Howard, Aaditya Ramdas, Jon McAuliffe, and Jasjeet Sekhon. Time-uniform Chernoff bounds via nonnegative supermartingales. *Probability Surveys*, 17(none):257 – 317, 2020. doi: 10.1214/18-PS321. URL `https://doi.org/10.1214/18-PS321`.

[17] Arthur Jacot, Franck Gabriel, and Clement Hongler. Neural tangent kernel: Convergence and generalization in neural networks. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31. Curran Associates, Inc., 2018. URL https://proceedings.neurips.cc/paper/2018/file/5a4be1fa34e62bb8a6ec6b91d2462f5a-Paper.pdf.

[18] Stanislaw Jastrzebski, Zac Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Amos Storkey, and Yoshua Bengio. Three factors influencing minima in SGD. In *International Conference on Learning Representations*, 2018.

[19] Ziwei Ji and Matus Telgarsky. Gradient descent aligns the layers of deep linear networks. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=HJflg30qKX.

[20] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations*, 2017.

[21] Bobby Kleinberg, Yuanzhi Li, and Yang Yuan. An alternative view: When does SGD escape local minima? In *Proceedings of the 35th International Conference on Machine Learning*, volume 80 of *Proceedings of Machine Learning Research*, pages 2698–2707. PMLR, 10–15 Jul 2018.

[22] Peter E Kloeden and Eckhard Platen. Stochastic differential equations. In *Numerical Solution of Stochastic Differential Equations*, pages 103–160. Springer, 1992.

[23] Qianxiao Li, Cheng Tai, and Weinan E. Stochastic modified equations and dynamics of stochastic gradient algorithms i: Mathematical foundations. *Journal of Machine Learning Research*, 20(40):1–47, 2019. URL http://jmlr.org/papers/v20/17-526.html.

[24] Kaifeng Lyu and Jian Li. Gradient descent maximizes the margin of homogeneous neural networks. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=SJeLIgBKPS.

[25] Stephan Mandt, Matthew D. Hoffman, and David M. Blei. A variational analysis of stochastic gradient algorithms. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, page 354–363, 2016.

[26] Song Mei, Andrea Montanari, and Phan-Minh Nguyen. A mean field view of the landscape of two-layer neural networks. *Proceedings of the National Academy of Sciences*, 115(33): E7665–E7671, 2018.

[27] Daniel Soudry, Elad Hoffer, Mor Shpigel Nacson, Suriya Gunasekar, and Nathan Srebro. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

[28] Aditya Varre, Loucas Pillaud-Vivien, and Nicolas Flammarion. Last iterate convergence of sgd for least-squares in the interpolation regime. 2021.

[29] Tomas Vaškevičius, Varun Kanade, and Patrick Rebeschini. Implicit regularization for optimal sparse recovery. *arXiv preprint arXiv:1909.05122*, 2019.

[30] Tomas Vaskevicius, Varun Kanade, and Patrick Rebeschini. The statistical complexity of early-stopped mirror descent. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 253–264. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/024d2d699e6c1a82c9ba986386f4d824-Paper.pdf.

[31] Stephan Wojtowytsch. Stochastic gradient descent with noise of machine learning type. Part I: Discrete time analysis, 2021.

[32] Blake Woodworth, Suriya Gunasekar, Jason D Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. In *Conference on Learning Theory*, pages 3635–3673. PMLR, 2020.

[33] Fan Wu and Patrick Rebeschini. A continuous-time mirror descent approach to sparse phase retrieval. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 20192–20203. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/e9470886ecab9743fb7ea59420c245d2-Paper.pdf.

[34] Lei Wu, Chao Ma, and Weinan E. How SGD selects the global minima in over-parameterized learning: A dynamical stability perspective. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

[35] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017. URL `https://openreview.net/forum?id=Sy8gdB9xx`.

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes] See our main Theorem.
   (b) Did you describe the limitations of your work? [Yes] We discuss throughout the paper the limitations of our model.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes] See our different theorems and propositions.
   (b) Did you include complete proofs of all theoretical results? [Yes] See the Appendix.
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Under figures.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No]
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [N/A] Everything runs on a laptop.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [N/A]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]