Stochasticity helps to navigate rough landscapes: comparing gradient-descent-based algorithms in the phase retrieval problem

IOP Publishing Mach. Learn.: Sci. Technol. 2 (2021) 035029 https://doi.org/10.1088/2632-2153/ac0615

MACHINE LEARNING
Science and Technology

OPEN ACCESS

RECEIVED
14 April 2021

REVISED
25 May 2021

ACCEPTED FOR PUBLICATION
27 May 2021

PUBLISHED
13 July 2021

PAPER

Stochasticity helps to navigate rough landscapes: comparing gradient-descent-based algorithms in the phase retrieval problem

Francesca Mignacco[1,*], **Pierfrancesco Urbani**[1] and **Lenka Zdeborová**[2]

[1] Université Paris-Saclay, CNRS, CEA, Institut de physique théorique, 91191 Gif-sur-Yvette, France
[2] SPOC laboratory, Ecole Polytechnique Fédérale de Lausanne (EPFL), Lausanne, Switzerland
* Author to whom any correspondence should be addressed.

E-mail: francesca.mignacco@ipht.fr

Keywords: disordered systems, neural networks, non-convex optimization

Abstract

In this paper we investigate how gradient-based algorithms such as gradient descent (GD), (multi-pass) stochastic GD, its persistent variant, and the Langevin algorithm navigate non-convex loss-landscapes and which of them is able to reach the best generalization error at limited sample complexity. We consider the loss landscape of the high-dimensional phase retrieval problem as a prototypical highly non-convex example. We observe that for phase retrieval the stochastic variants of GD are able to reach perfect generalization for regions of control parameters where the GD algorithm is not. We apply dynamical mean-field theory from statistical physics to characterize analytically the full trajectories of these algorithms in their continuous-time limit, with a warm start, and for large system sizes. We further unveil several intriguing properties of the landscape and the algorithms such as that the GD can obtain better generalization properties from less informed initializations.

1. Introduction

Algorithms based on gradient-descent (GD) are the workhorses of many machine learning applications involving the optimization of a high-dimensional non-convex loss function. In particular, stochastic GD (SGD) has proved to be extremely efficient in navigating complex loss landscapes. However, despite its practical success, the theoretical understanding of the reasons behind the good generalization properties of the algorithm remains sparse. Empirical evidence suggests that the interplay between the optimization algorithm and the landscape is crucial to achieve good performances. It has been shown, for instance, that the loss landscape of state-of-the-art deep neural networks is far from simple: adversarial initialization can trap SGD into global minima with poor generalization [1]. Therefore, understanding the dynamics of SGD is paramount in machine learning and optimization.

Investigating the dynamics of SGD and the role of the stochasticity is consequently an active direction of research. While the practical success of SGD compared to GD is rather generally accepted, it is still far from clear what is really the key factor responsible for this. Cases where the superiority of SGD with respect to GD was shown theoretically are sparse, but see e.g. [2, 3]. One hurdle that appears in theoretical analysis is how to properly define the continuous limit of SGD. In the limit of learning rate going to zero, SGD is considered to lead to the gradient-flow limit, see e.g. [4], thus the difference with gradient flow disappears. For learning rate kept finite, a line of works characterizes SGD as a discretization of a continuous-time Langevin-type process [4–7]. The dependence of the variance of the noise on the current-weights and time is, however, not given in a closed form in these works and thus difficult to analyze explicitly. Another line of work challenged the central-limit-theorem assumption of finite noise-variance behind these works by proposing the stochastic noise is heavy-tail distributed [8]. In our work, we instead consider a variant of the stochastic GD called persistent-SGD, as recently defined in [9]. Persistent-SGD has a well define flow limit $\eta \to 0$ and our analysis thus does not require other assumptions about the nature of stochastic noise.

Learning theory and computer science usually proceed in a manner that makes minimalistic assumptions on the data distribution. Statistical physics usually takes a complementary way of understanding well prototypical settings that capture the essence of the question. This is the path we take in this paper and compare the behavior of GD-based algorithms on a prototypical choice of data and learning model leading to high-dimensional and non-convex landscape. Specifically, we consider the problem of phase retrieval where the task consists in recovering an unknown signal from a set of observations—the absolute value of the signal's projections onto measurement vectors. This problem appears in a series of applications, including optics [10, 11], acoustics [12], and quantum mechanics [13]. We will consider the problem where the measurements are i.i.d. Gaussian vectors and the number of measurements M is only constant times the dimensionality of the signal N, $\alpha = M/N$. We consider the high-dimensional limit where both the number of training samples and the input dimensions go to infinity, at ratio α of order one, typically 2–3. In this work we view the phase retrieval as a prototypical example of a simple single-layer neural network where the measurement vectors correspond to the input samples, and the signal corresponds to the teacher-network weights. The measurements then correspond to the output labels. In the spirit of learning with neural networks we are interested in the corresponding generalization error, i.e. the ability to predict labels for previously unseen samples. We stress that it is not the goal of this work to provide a competitive algorithm for the phase retrieval. In the setting considered in this paper (i.e. i.i.d. Gaussian inputs and teacher produced labels) it was conjectured that the approximate message passing algorithm (AMP) cannot be beaten in the large size limit [15]. Instead, the main goal of this paper is to study the performance of gradient-based algorithms and the loss landscape of the phase retrieval problem serves us as a high-dimensional intrinsically non-convex prototype having multiple spurious minima and only one solution (with a $\mathbb{Z}_2$ symmetry) leading to perfect generalization error.

We note that the landscape of phase retrieval problem is somewhat different than the one of deep neural networks, that are highly overparametrized and present entire regions of solutions with zero training error and a good generalization. Consequences of this difference and thus relevance of the present work for learning with state-of-the art neural networks is left for future work. Instead the present work investigates the performance of gradient-based algorithms in an archetypal non-convex high-dimensional setting providing a benchmark to assess the role played by stochasticity in non-convex optimization problems in general.

Our main contributions can be summarized as follows.

- We perform a series of numerical simulations in order to assess the generalization performance of the GD, multi-pass stochastic GD, its persistent-SGD version, and the Langevin algorithm as a function of the control parameters (mini-batch size, persistence time, temperature). Our experimental findings reveal that in the considered problem stochasticity is beneficial for generalization. We also shed light on the qualitative difference between the sources of noise in the algorithms.
- We investigate the role of the warm start and we find that GD can be trapped very close to the signal, while perfect recovery can be reached starting from less informed initializations.
- We then apply dynamical mean-field theory (DMFT) from statistical physics to provide an analytic characterization of the full trajectory of the continuous limit of the GD, persistent-SGD and Langevin algorithms in the high-dimensional limit where the number of samples and dimension are both large, but their ratio $\alpha = \mathcal{O}(1)$, at times linear in the dimension. We use the theoretical curve as a baseline to show that the observed behavior is not due to finite-size or finite-learning-rate effects.

Further related works

In this paper, we consider the phase retrieval problems with Gaussian measurements and signal in the high-dimensional limit. The loss landscape complexity of this problem was studied using the Kac-Rice method in [14], however, bringing this analysis to concrete results seemed to be technically challenging. Signal recovery in this problem was studied from the information-theoretic point of view and using AMP algorithms that are considered optimal among all polynomial algorithms for this case [15–17]. In particular it is known that while information-theoretically zero generalization error can be reached for $\alpha > 1$, the AMP algorithm is able to do so for $\alpha > 1.13$.

Performance of GD for phase retrieval is worse than the one of AMP in terms of sample complexity and also harder to analyze. In practice, one often uses GD initialized spectrally [18], i.e. in the eigenvector corresponding to the leading eigenvalue of a suitable matrix constructed from the labels and the measurement vectors [19]. Such spectral initialization is also motivating our use of warm start that is mimicking it. Concerning randomly initialized GD, [20] showed that GD needs a training set of size $\sim \mathcal{O}(N\mathrm{poly}(\log N))$ to at each time step. If we introduce other works in computer science consider GD-type algorithms for phase retrieval requiring $\mathcal{O}(N\mathrm{poly}(\log N))$ samples [25]. The analysis carried out in [21] suggests that the randomly-initialized algorithm can achieve perfect generalization with much lower linear

sample complexity. Authors of [23] then show that linear (with unspecified large constant) sample complexity is achievable with randomly initialized GD for a suitably chosen loss function. Finally [24] have shown that over-parametrization can bring the sample complexity of randomly initialized GD down to $\alpha = 2$. While in the present work we will not be considering overparametrization, we are interested in performance of gradient-based algorithms for similarly small sample complexity α. We will be investigating several gradient-based algorithms and judge their performance by the number of samples they require for recovery of the signal. The fewer samples the better. This is why we focus on the regime of $\alpha = \mathcal{O}(1)$.

The online SGD for phase retrieval has been studied, e.g. in [26]. Results for (multi-pass) stochastic GD in phase retrieval are not known up to our best knowledge. A theoretical understanding of the performance of (multi-pass) SGD at small sample complexity requires taking into account the full trajectory of the algorithm which is challenging and done in the present paper.

2. The model and the data

We study the supervised learning problem of recovering an $N-$ dimensional real-valued vector: $\underline{w}^{(0)} = \{w_1^{(0)}, \dots, w_N^{(0)}\}$ from a set of $M = \alpha N$ real-valued noiseless measurements $\underline{\xi}^{\mu} = \{\xi_1^{\mu}, \dots, \xi_N^{\mu}\}$ of dimension N. We consider the signal $\underline{w}^{(0)}$ to be extracted with the uniform measure on the N-dimensional hyper sphere $|\underline{w}^{(0)}|^2 = N$. We take the components of the vectors $\underline{\xi}^{\mu}$ to be i.i.d. Gaussian random variables with zero mean and unit variance. The non-linear measures of the signal vector $\underline{w}^{(0)}$ are encoded in the labels:

$$y^{\mu} = \left| \frac{1}{\sqrt{N}} \underline{\xi}^{\mu} \cdot \underline{w}^{(0)} \right|, \quad \forall \mu = 1, \dots, M. \tag{1}$$

We note that in applications the complex-valued phase retrieval is more relevant, yet for the purpose of the present paper, which is studying the performance of the gradient-based algorithms, the real-valued version is sufficiently rich. We consider learning with a single-layer neural network by the minimization of the empirical risk:

$$\mathscr{L}(\underline{w}) = \sum_{\mu=1}^{M} v(h_{\mu}; h_{\mu}^{(0)}), \tag{2}$$

where v is a cost function having a global minimum at $h_{\mu} = h_{\mu}^{(0)}$ and we have defined:

$$h_{\mu} = \frac{1}{\sqrt{N}} \underline{\xi}^{\mu} \cdot \underline{w}, \qquad h_{\mu}^{(0)} = \frac{1}{\sqrt{N}} \underline{\xi}^{\mu} \cdot \underline{w}^{(0)}. \tag{3}$$

In what follows we consider a loss of the form:

$$v(h, h_0) = \frac{1}{4}(h^2 - h_0^2)^2. \tag{4}$$

However, the analytic derivation of the DMFT can be carried out for every twice-differentiable function v. Note that the empirical risk depends on the labels y^{μ} only through $h_{\mu}^{(0)}$. We consider a particular regularization of the weights where the training dynamics of $\underline{w}(t)$ is constrained on the hyper-sphere. In section C, we show that our results hold in a qualitative same manner for the more standard ridge regularization.

3. The analyzed algorithms

In this section we define the GD-based algorithms under consideration and their continuous-time limit that will then be studied using dynamical mean-field theory. The discrete dynamics of full-batch GD is given by the weights update:

$$\begin{aligned} w_i(t+\eta) &= w_i(t) - \eta \left[\partial_{w_i} \mathscr{L}(\underline{w}) + \hat{\nu}(t) w_i(t) \right] \\ &= w_i(t) - \eta \left[\sum_{\mu=1}^{\alpha N} \partial_1 v(h_{\mu}; h_{\mu}^{(0)}) \frac{1}{\sqrt{N}} \xi_i^{\mu} + \hat{\nu}(t) w_i(t) \right], \end{aligned} \tag{5}$$

for all $i = 1, ..., N$, where $\eta > 0$ is the discrete time step and $\partial_1 v(h; h_0)$ indicates the derivative of the loss function with respect to its first argument. We have introduced a Lagrange multiplier $\hat{\nu}(t)$ to enforce the

spherical constraint on the weights at each time step. This is equivalent to a projection on the sphere at each iteration, which is how we implement the numerical simulations. In the following, we analyze different ways to add stochasticity to the dynamics.

3.1. Multi-pass stochastic GD dynamics

We study multi-pass stochastic GD, where the samples are reused multiple times during training. The mini-batches are sampled with replacement with size $B = \mathscr{C}M$, $\mathscr{C} \in (0,1]$ at each time step. If we introduce a set of binary variables $s_\mu(t) \in \{0,1\}$, $\mu = 1,...,M$ to select which samples are used compute the gradient, then in the large N limit the vanilla-SGD algorithm described above is equivalent to draw:

$$s_\mu(t) = \begin{cases} 1 \text{ w.p. } \mathscr{C} \\ 0 \text{ otherwise} \end{cases} \tag{6}$$

independently at each time step. However, the continuous-time limit $\eta \to 0$ different from the gradient-flow is not well-defined in this case. As done in [9], in order to consider the continuous-time dynamics, we define a discrete-time process for the variables $s_\mu(t)$ that has a well-defined continuous-time limit. We divide the time interval in finite steps of size η and we define the following *persistent* version of the stochastic GD algorithm:

$$\begin{aligned} &s_\mu(t=0) = \begin{cases} 1 \text{ w.p. } \mathscr{C} \\ 0 \text{ otherwise} \end{cases}, \\ &\text{Prob}\left(s_\mu(t+1) = 1 | s_\mu(t) = 0\right) = 1 - \text{Prob}\left(s_\mu(t+1) = 0 | s_\mu(t) = 0\right) = \frac{\eta}{\tau}, \\ &\text{Prob}\left(s_\mu(t+1) = 0 | s_\mu(t) = 1\right) = 1 - \text{Prob}\left(s_\mu(t+1) = 1 | s_\mu(t) = 1\right) = \frac{1-b}{b\tau}\eta. \end{aligned} \tag{7}$$

In this case, each pattern stays out of the training mini-batch for a typical time τ, that we will refer to as the persistence time. The stochastic gradient flow (SGF) dynamics is obtained by taking the $\eta \to 0$ limit of equation (7):

$$\frac{\partial w_i(t)}{\partial t} = -\frac{1}{\mathscr{C}} \sum_{\mu=1}^{\alpha N} s_\mu(t) \partial_1 v(h_\mu(t); h_\mu^{(0)}) \frac{1}{\sqrt{N}} \xi_i^\mu - \hat{\nu}(t) w_i(t), \tag{8}$$

for all $i = 1,...,N$, where we have rescaled the gradient by the fraction of samples in the mini-batch. Note that, in this setting, there are two parameters controlling the stochasticity of the algorithm: the mini-batch size $\mathscr{C}$ and the persistence time τ. The standard SGD algorithm is recovered from (7) by setting $\tau = \eta/\mathscr{C}$ and finite learning rate η. In appendix D, we show by numerical simulations with decreasing learning rate that the $\eta \to 0$ limit of the persistent SGD algorithm is different than GD, while this is not the case for standard SGD.

3.2. Langevin dynamics

A different kind of stochastic dynamics is provided by the Langevin algorithm at temperature T, whose flow limit is defined by the following system of stochastic differential equations:

$$\frac{\partial w_i(t)}{\partial t} = -\sum_{\mu=1}^{\alpha N} \partial_1 v(h_\mu(t); h_\mu^{(0)}) \frac{1}{\sqrt{N}} \xi_i^\mu + \varsigma_i(t) - \hat{\nu}(t) w_i(t). \tag{9}$$

for all $i = 1,...N$. The random vector $\underline{\varsigma}(t)$ is Gaussian white noise:

$$\begin{aligned} &\langle \varsigma_i(t) \rangle = 0, && \forall i = 1,...N, \\ &\langle \varsigma_i(t) \varsigma_j(t') \rangle = 2T\delta_{ij}\delta(t-t'), && \forall i,j = 1,...N. \end{aligned} \tag{10}$$

Note that by setting $\mathscr{C} = 1$ in equation (8) or $T = 0$ in equation (9) we recover the full-batch gradient-flow algorithm.

3.3. Warm initialization

In order to explore the energy landscape more thoroughly we consider here, next to the usual random initialization, informed/warm initializations. We initialize the weight vector as follows:

$$\underline{w}(t=0) = m_0 \underline{w}^{(0)} + c\underline{z} \in \mathbb{R}^N, \tag{11}$$

where $m_0 > 0$ is (on average) the initial projection of the weight vector onto the signal, i.e. the average *magnetization*:

$$m(t) = \frac{1}{N}\underline{w}(t)\cdot\underline{w}^{(0)}, \tag{12}$$

at time $t = 0$. The components of $\underline{z}$ are i.i.d. standard Gaussian variables and the coefficient c is such that $|\underline{w}(t=0)|^2 = N$. Note that the warm initialization breaks the $\mathbb{Z}_2$ symmetry of the problem. Therefore, in the following $m(t)\in(0,1]$, $\forall t$. We stress here that while in learning we are usually concerned with the test error/performance, in the setting considered here (under the spherical constraint) the test error is monotonic in the magnetization, see appendix B. Thus, in the following we directly use the magnetization as a measure of accuracy.

This warm initialization can be thought of as a proxy for algorithms where GD (or its variants) is run after the weights have been spectrally initialized, i.e. using the principal eigenvalue of a given pre-processing matrix as initial guess for the weights. Spectrally initialized GD is used in a range of applications, see e.g. [18], as well as studied theoretically, see e.g. [27]. Warm initialization is formally needed to obtain non-trivial results for times linear in the dimension. Indeed, because of the $\mathbb{Z}_2$ symmetry we would need time at least logarithmic in dimension in order to escape the space of weights uncorrelated with the solution. This was referred to as the 'escape from mediocrity' in [28].

4. Characterization of the dynamics

In this section we provide a closed-form characterization of the flow dynamics of the persistent-SGD and Langevin algorithms presented above in the high-dimensional limit. To this end, we apply dynamical mean-field theory from statistical physics [29–31]. This analytic framework is useful to study the stochastic evolution of large systems of interacting degrees of freedom [32–34]. DMFT has been rigorously proven in some specific cases [35], but not yet in the present one. Here we present the main analytic results, more details are provided in appendix A. The derivation follows the line of [9, 36], for a different data structure and loss function. We also need to include the spherical constraint and Langevin noise in the dynamics. We consider the high-dimensional limit $N\to\infty$, at fixed sample complexity $\alpha = M/N$, mini-batch fraction $\mathscr{b}$, persistent time τ and temperature T. For simplicity, we regroup the flow dynamics of multi-pass SGD (8) and Langevin (9) in the same equation:

$$\begin{aligned}\frac{\partial w_i(t)}{\partial t} &= -\hat{\nu}(t)w_i(t) + \varsigma_i(t)\\ &\quad - \frac{1}{\mathscr{b}}\sum_{\mu=1}^{\alpha N} s_\mu(t)\partial_1 v(h_\mu(t);h_\mu^{(0)})\frac{1}{\sqrt{N}}\xi_i^\mu.\end{aligned} \tag{13}$$

The performance of the algorithms as a function of training time is encoded in the magnetization $m(t)$ defined in equation (12), that is equal to 1 for perfect recovery of the signal. In the high-dimensional limit, we obtain that the evolution of the magnetization is described by the following deterministic differential equation:

$$\partial_t m(t) = -\hat{\nu}(t)m(t) - \mu(t), \qquad m(0) = m_0\,, \tag{14}$$

where

$$\begin{aligned}\hat{\nu}(t) &= -\frac{\alpha}{\mathscr{b}}\langle\tilde{h}(t)s(t)\partial_1 v(\tilde{h}(t);h_0))\rangle + T,\\ \mu(t) &= \frac{\alpha}{\mathscr{b}}\langle s(t)h_0\,\partial_1 v(\tilde{h}(t);h_0)\rangle,\\ \tilde{h}(t) &\equiv h(t) + h_0 m(t).\end{aligned} \tag{15}$$

The brackets $\langle\cdot\rangle$ stand for the average over different sources of noise:

- the binary variable $s(t)$, distributed as in equation (7) for $\eta\to 0$;
- the standard Gaussian variable $h_0\sim\mathcal{N}(0,1)$;
- the effective stochastic process for the typical gap $h(t)$ defined in equation (3).

The evolution of (t) is given by:

$$\partial_t h(t) = -\left(\hat{\nu}(t) + \delta\nu(t)\right)h(t) - \frac{1}{\mathscr{b}}s(t)\partial_1 v(\tilde{h}(t);h_0) + \int_0^t \mathrm{d}t'\, M_R(t,t')h(t') + \chi(t), \tag{16}$$

with initial condition:

$$P(h(0)) = \frac{1}{\sqrt{2\pi(1-m_0^2)}} e^{-h(0)^2/2(1-m_0^2)}. \tag{17}$$

The dynamical noise $\chi(t)$ in equation (16) is Gaussian distributed, with:

$$\begin{aligned} \langle \chi(t) \rangle &= 0, \\ \langle \chi(t)\chi(t') \rangle &= 2T\delta(t-t') + M_C(t,t'). \end{aligned} \tag{18}$$

The expressions for the kernels $M_C(t,t')$ and $M_R(t,t')$ and the auxiliary function $\delta\nu(t)$ are given by:

$$\begin{aligned} M_C(t,t') &= \frac{\alpha}{\mathscr{B}^2} \langle s(t)s(t')\partial_1 v(\tilde{h}(t);h_0)\partial_1 v(\tilde{h}(t');h_0)) \rangle, \\ M_R(t,t') &= \frac{\alpha}{\mathscr{B}^2} \frac{\delta}{\delta P(t')} \langle s(t)\partial_1 v(\tilde{h}(t);h_0)) \rangle \Big|_{P=0}, \\ \delta\nu(t) &= \frac{\alpha}{\mathscr{B}} \langle s(t)\partial_1^2 v(\tilde{h}(t);h_0)) \rangle, \end{aligned} \tag{19}$$

where $P(t')$ is a linear perturbation applied to h at time t' and then set to zero, that we only need to define $M_R(t,t')$. Overall, we obtain that the performance of the algorithms over time is described by a system of integro-differential equations that must be solved numerically in a self-consistent way. As done in [9], we start from a simple guess of the kernels and the auxiliary functions and we compute many realizations of the curve $h(t)$ in equation (16). We use these curves to update the kernels and we iterate the procedure until convergence. The details are relegated to appendix A.1. A first implementation of this procedure was proposed in [37, 38]. Once the stochastic process defined in equation (16) has reached convergence, we can compute other quantities of interest. For instance, we can track the evolution of the average training loss defined in equation (2) in the high-dimensional limit:

$$\ell(t) = \langle v(\tilde{h}(t);h_0) \rangle. \tag{20}$$

These equations provide a dimension-independent way to track the performance of the algorithm in the limit of high-dimensions and infinitesimal learning rate as a function of time. Indeed, since the solution of the problem is planted and the measurements are noiseless, in this case zero training loss corresponds to zero generalization error.

Note that this formalism allows to study the dynamics of the corresponding algorithms without any approximation on the distribution of the noise introduced by stochasticity. This is at variance with the works that consider SGD as a noisy approximation of GD invoking variants of central limit theorem [4–8].

5. Results for the dynamics

In this section, we discuss our findings on the dynamics of the gradient-based algorithms under consideration. We compare the results from simulations to the DMFT theoretical prediction. The DMFT is valid in the continuous flow limit and in the high-dimensional limit. The simulations are performed for sizes large enough and learning rate small enough so that this limit is closely approached. This analysis sheds light on how stochasticity helps to navigate the loss landscape and on the impact of the different control parameters, notably the batch size $\mathscr{B}$, temperature T, and persistence time τ, on the test performance.

5.1. The trapping landscape

Figure 1 illustrates the performance of GD starting from three increasing initializations: $m_0 = 0.5$ (left), $m_0 = 0.65$ (center), and $m_0 = 0.8$ (right) at $\alpha = 2$, i.e. number of samples twice the dimension. In the three lower panels, we plot the magnetization for different seeds—corresponding to different realizations of the noise vector $\underline{z}$ defined in equation (11)—with a dataset, i.e. the inputs and labels, drawn at random and fixed. The evolution of different instances from simulations is thus probing the very same loss-landscape, the figure then highlights the complexity of the landscape. First, we observe that a warm start is not enough to reach perfect recovery. This suggests that the landscape is very rough, with multiple local minima at all heights. Indeed, we see that GD can get stuck even very close to the global minimum at $m = 1$. From the right panel of the figure, we see that at time $t \sim 10$ all seeds initialized with magnetization $m_0 = 0.8$ have achieved perfect recovery $m = 1$. However, the left and center panels show that some seeds starting at $m_0 < 0.8$ and

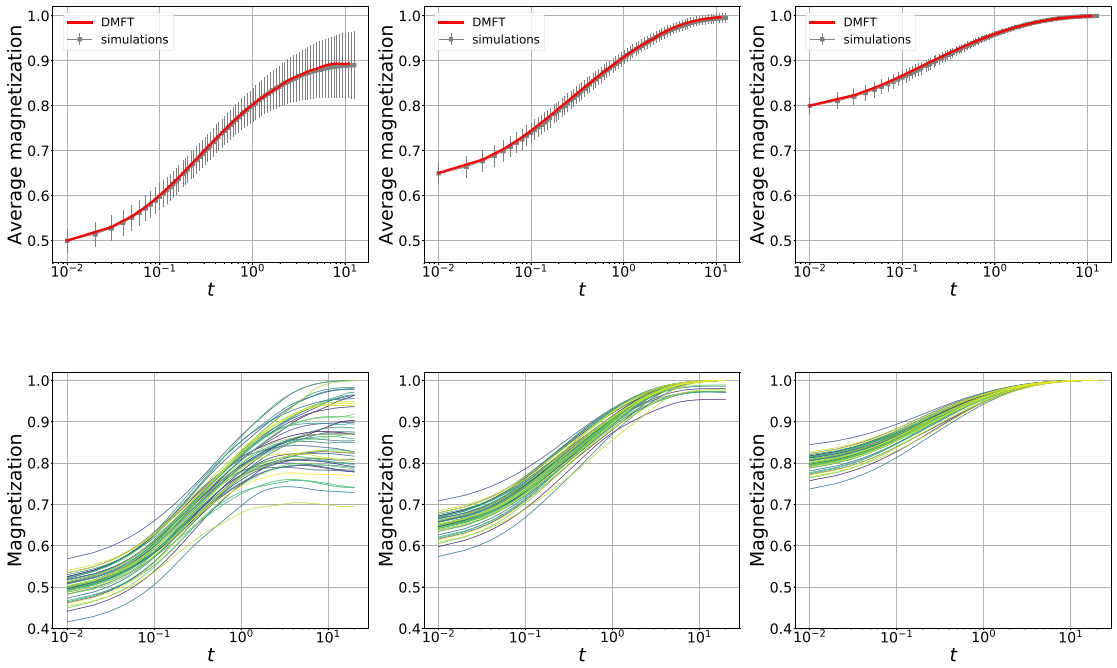

Figure 1. Upper. Average magnetization as a function of training time for the full-batch GD/flow algorithm. We consider $\alpha = M/N = 2$ and three different initializations: $m_0 = 0.5$ (left), $m_0 = 0.65$ (center), $m_0 = 0.8$ (right). The grey dots represent numerical simulations ($N = 1000$, $\eta = 0.01$), averaged over 1000 seeds (generating a new dataset and signal for each seed). The full red line marks the theoretical prediction from DMFT obtained in the high-dimensional limit of gradient-flow. **Bottom.** We fix the landscape by fixing the dataset and we show 50 instances of the magnetization as a function of time, for different realizations of the noise vector $\underline{z}$ defined in equation (11), at $m_0 = 0.5$ (left), $m_0 = 0.65$ (center), $m_0 = 0.8$ (right). For visibility purposes, we plot $t + \eta$ on the x-axes.

reaching $m = 0.8$ only at $t > 0$ can get stuck for long times. Hence we deduce that the topological complexity of the landscape is such that some regions of the weights space can be trapping even if they are closer to the signal than others that do not trap the dynamics. We observe that a more informed initialization does not guarantee a better generalization. This can be further seen comparing the left panel to the central one. Indeed, we find that some seeds initialized at $m_0 > 0.6$ are stuck at $m < 1$ at time $t \sim 10$, while some seeds starting at $m_0 < 0.6$ have already reached perfect generalization. Consequently, in this regime of parameters, the full trajectory of the algorithm is crucial to achieve perfect recovery.

In the upper panels of figure 1, we compare the average magnetization from numerical simulations at finite system size and finite learning rate (grey dots) to the theoretical prediction (red line) obtained by integrating the DMFT Equations derived in the high-dimensional continuous limit. In this case, we generate a new dataset for each simulations in order to remove sample-to-sample fluctuations. We find a very good agreement between asymptotic theory and the average from simulations already for the used system sizes and learning rates, indicating that the observed behavior is not a feature of finite size or finite learning rate effects. Additional simulations supporting this evidence are left to appendix D.

5.2. Multi-pass SGD outperforms GD

Figure 2 shows the average magnetization—defined in equation (12)—and the average training loss—defined in equation (2)—as a function of time for full-batch GD, multi-pass SGD and its persistent version. In the case of multi-pass SGD, we sample (with replacement) minibatches of size $\mathscr{B}M$ at each time step. In figure 3, we depict different instances of the dynamics, corresponding to different realizations of the dataset and the noise vector $\underline{z}$ (equation (11)). We find that SGD and persistent-SGD with $\tau = 1$ outperform GD in recovering the hidden signal. Indeed, at time scales at which persistent-SGD has already reached magnetization one and zero loss, GD is stuck in regions of poorer generalization. The average magnetization of SGD lies between the two. Therefore, a finite batch size is beneficial for the performance. Furthermore, the behavior of the curves for different seeds unveils an important role played by the persistence time. Indeed, while the evolution of the magnetization for GD is characterized by long plateaus alternated by sudden jumps, persistent-SGD is not stuck in the same region for long times. Again, the behavior of SGD is intermediate between the two: we see from the central panel of figure 3 that the disappearance of the plateaus

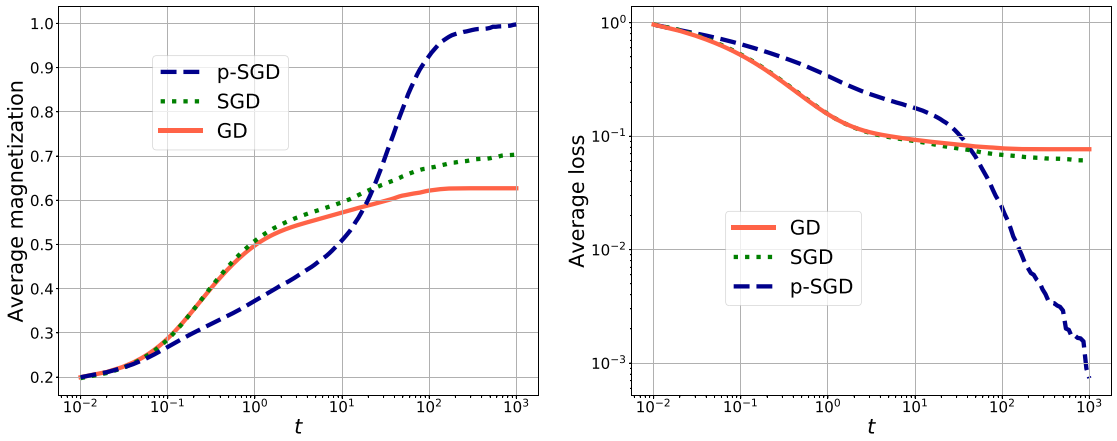

Figure 2. Average magnetization (left) and average training loss (right) as a function of training time, at fixed $\alpha = M/N = 3$, initial magnetization $m_0 = 0.2$, input dimension $N = 1000$, learning rate $\eta = 0.01$. We show the performance of full-batch gradient descent (red line), multi-pass vanilla SGD at $\mathscr{b} = 0.5$ (dotted green line), and persistent SGD at $\tau = 1$, $\mathscr{b} = 0.5$ (dashed blue line). The averages are computed over 500 seeds (generating a new instance for each seed). At time $t = 1000$, the percentages of seeds that have reached training loss below 10^{-7} are: 9% (GD), 30% (SGD), 99% (persistent-SGD). For visibility purposes, we plot $t + \eta$ on the x-axes.

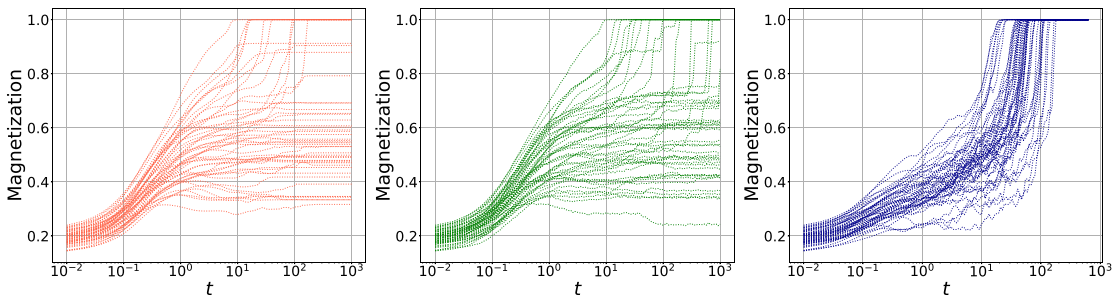

Figure 3. We show 50 instances of the magnetization as a function of training time for GD (left), SGD at $\mathscr{b} = 0.5$ (center) and persistent-SGD at $\mathscr{b} = 0.5$, $\tau = 1$ (right). All other parameters are as in figure 2. For each seed, a new instance is generated. For visibility purposes, we plot $t + \eta$ on the x-axes.

is a feature of a finite persistence time. These findings suggest that the interplay of the finite batch size and the persistence time is crucial to achieve the optimal performance. Additional simulations supporting this numerical evidence are provided in appendix D.

5.3. The role of the noise

Figure 4 illustrates the effect of different sources of stochasticity on the generalization performance. In particular, we compare the role played by the white noise at temperature T in the Langevin algorithm to the double source of noise in the SGD algorithm: the finite batch size $\mathscr{b}$ and the persistence time τ. In the left panel, we depict the dependence of the SGD algorithm on the batch size, at fixed persistence time. We find that the generalization performance is non-monotonic in the batch size and the optimal value is attained at intermediate $\mathscr{b}$. Therefore, at variance with what observed in deep neural networks trained on real datasets [6, 40], in our case we obtain that the optimal batch size is an extensive fraction of the total number of samples. The central panel displays the (median) performance of SGD for different values of the persistence time τ, at fixed batch size. For times $t \leq \tau$, the samples used to compute the gradient (on average) do not change, and thus the dynamics presents plateaus. However, as soon as $t > \tau$, the mini-batch is refreshed. This results in a sudden increase in performance at times $t \sim \tau$, that becomes more visible the larger τ. Moreover, we observe a non-monotonic behavior of the performance as a function of τ. On the one hand, increasing τ shifts the final plateau at larger times, delaying the recovery of the signal. On the other hand, if the persistence time is too small, the dynamics gets trapped close to the signal, displaying plateaus followed by sudden jumps similarly as for GD (see Figure 3). There is therefore an intermediate range of persistence times τ for which the performance is the best (better than vanilla SGD). Since the literature often compares the SGD noise to the Langevin noise [4–7, 39] we compare here to the performance achieved by the Langevin

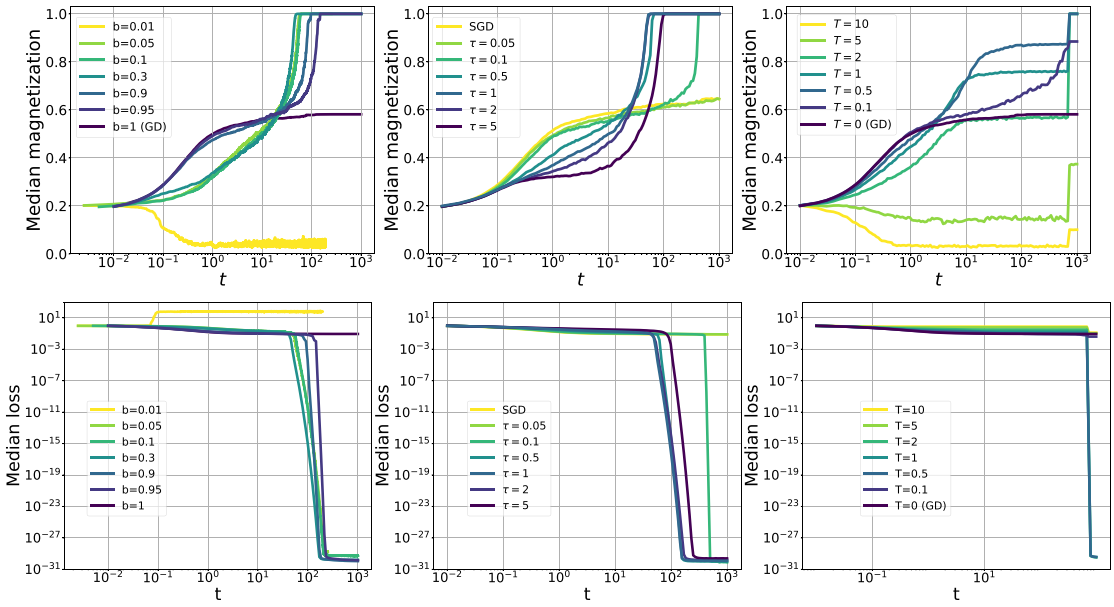

Figure 4. Median magnetization (above) and median loss (below) as a function of training time, at fixed $\alpha = M/N = 3$, initialization $m_0 = 0.2$, dimension $N = 1000$, learning rate $\eta = 0.01$. The median is computed over 250 seeds, drawing a new dataset, signal and initialization for each seed. For visibility purposes, we plot $t + \eta$ on the x-axes. **Left.** We show persistent-SGD for increasing values of batch size $\mathscr{b} = 0.01, 0.05, 0.1, 0.3, 0.9, 0.95, 1$ and fixed persistence time $\tau = 1$. In the case $b = 0.1$, the learning rate has been reduced to $\eta = 0.005$, for $\mathscr{b} = 0.01, 0.05$ we have used $\eta = 0.0025$. **Center.** We show persistent-SGD at increasing values of persistence time $\tau = 0.05, 0.1, 0.5, 1, 2, 5$ and vanilla SGD, both algorithms at fixed batch size $\mathscr{b} = 0.5$. **Right.** We show the Langevin algorithm at increasing temperatures $T = 0$ (GD), $0.1, 0.5, 1, 2, 5, 10$. At time $t = 700$ the temperature is set to zero.

algorithm at fixed temperature. The right panel of figure 4 depicts the performance of the Langevin algorithm for different values of temperature T. At large times ($t = 700$ in the figure) the temperature is switched to zero. We find that the best performance is again reached for intermediate values of the temperature T. We underline the qualitative difference between the effective noise introduced by multi-pass SGD and the white noise of Langevin algorithm. The variance of the noise in Langevin is fixed by the temperature, therefore—in order to reach a minimum—an annealing protocol must be implemented and optimized. In contrast, the noise introduced by SGD is automatically reduced during training and it is zero at the global minimum. Therefore, multi-pass SGD has a built-in self annealing protocol, that can be optimized by tuning only two parameters ($\mathscr{b}$ and τ) instead of the whole trajectory of the temperature over time.

5.4. The analytic characterization

Figure 5 shows the comparison between the average performance of persistent-SGD obtained from numerical simulations (grey symbols) with the prediction derived by integrating the DMFT Equations (red line). The left panel depicts the average magnetization, while the right panel displays the average training loss as a function of time. Figure 6 displays the same comparison for the Langevin algorithm. In both cases, we find a very good agreement between theory and simulations.

5.5. Random initialization

Figure 7 investigates the behavior of full-batch GD (full red lines) and persistent SGD (dashed blue lines) starting from random initialization at fixed $\alpha = 2.5$. Persistent SGD is run at fixed $\mathscr{b} = 0.5$, $\tau = 2$. We show the median magnetization (main plots) and the median loss (insets) as a function of time for increasing values of the dimension: $N = 100$ (above-left panel), $N = 500$ (above-right panel), $N = 1000$ (below-left panel). and $N = 2500$ (below-right panel). In this case $m_0 = 0$ and the warm start in the four panels is only given by finite size effects. We clearly see that, at time scales shown here, GD is stuck at a plateau of height decreasing as the dimension N increases. As studied in [21], the recovery transition of GD starting from random initialization for comparable system sizes happens at $\alpha \approx 6$, which is few times larger than the value $\alpha = 2.5$ considered here. However, we observe that already at $\alpha = 2.5$ the persistent-SGD algorithm can reach perfect recovery for the system sizes under consideration. The time to reach the solution from random initialization is, as expected, compatible with logarithmic increase in the system size. These observations suggest that the recovery transition for stochastic GD starting from random initialization is shifted to lower values of α when compared to GD. This is an interesting direction for future investigations.

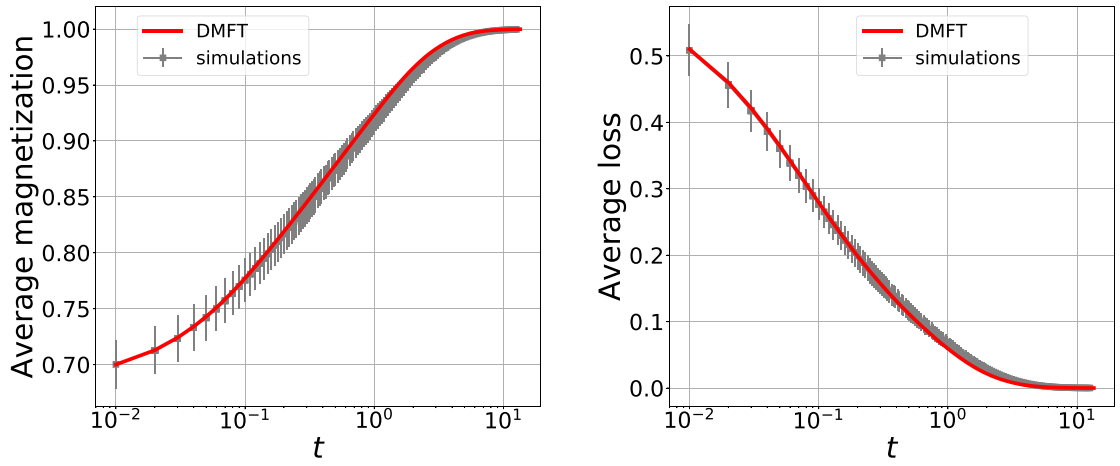

Figure 5. Average magnetization (left) and average training loss (right) as a function of time for the persistent SGD algorithm in the spherical setting, at fixed $\alpha = M/N = 3$, warm start $m_0 = 0.7$, persistence time $\tau = 2$, batch size $\mathscr{b} = 0.6$. The grey dots represent the result from numerical simulations, averaged over 500 seeds at learning rate $\eta = 0.01$ and dimension $N = 1000$. The red curve marks the performance predicted by the numerical integration of DMFT equations.

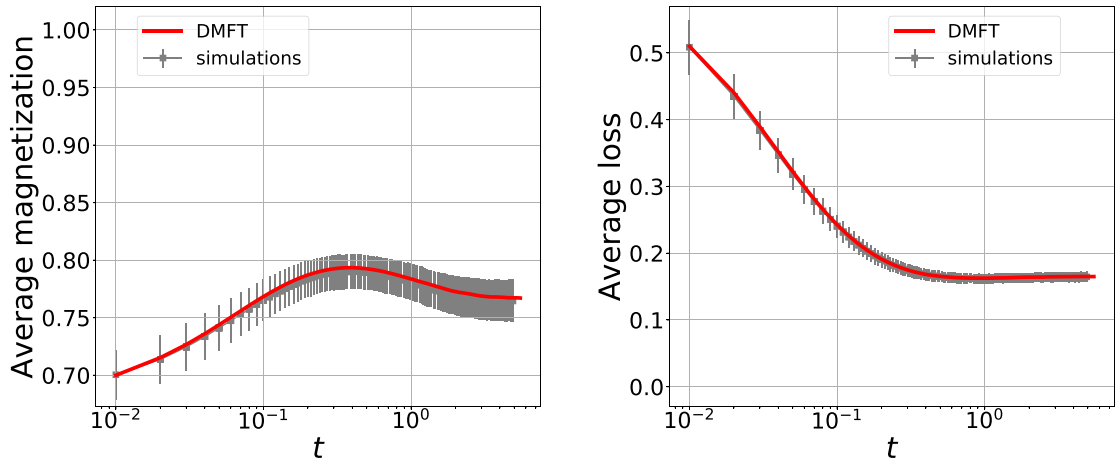

Figure 6. Average magnetization (left) and average training loss (right) as a function of time for the Langevin algorithm in the spherical setting, at fixed $\alpha = M/N = 3$, warm start $m_0 = 0.7$, temperature $T = 1$. The grey dots represent the result from numerical simulations, averaged over 1000 seeds at learning rate $\eta = 0.01$ and dimension $N = 1000$. The red curve marks the performance predicted by the numerical integration of DMFT equations.

6. Discussion

In this paper, we have considered the real-valued phase retrieval problem as a paradigmatic highly non-convex optimization problem to test the generalization performance of full-batch GD and some of its stochastic variants: multi-pass SGD, its persistent version, and the Langevin algorithm. We have shown that stochasticity is crucial to achieve perfect recovery of the hidden signal at low sample complexity so that stochastic GD outperforms GD in this task. We have observed intriguing features of the loss profile and illustrated how various sources of noise allow the dynamics to circumvent the traps in the landscape. We have provided an analytic description of the learning curve in the infinite-dimensional and continuous-time limit via the dynamical mean-field theory, showing that the observed behavior is not due to finite size effects or to a finite learning rate. The present work leads to interesting extensions both on the analytic and numerical sides. On the one hand, the characterization of the dynamical evolution of the algorithms via DMFT can be extended to include realistic initializations (e.g. spectral initialization). On the other hand, it would be interesting to test the persistent variant of multi-pass SGD and investigate the role of the persistence time on real datasets and architectures, which we leave for future work. In this regard, another relevant extension of this work could be generalizing the DMFT analysis of SGD models of structured data with low intrinsic dimension embedded in large dimension, such as the Hidden Manifold Model presented in [22]. Finally, the

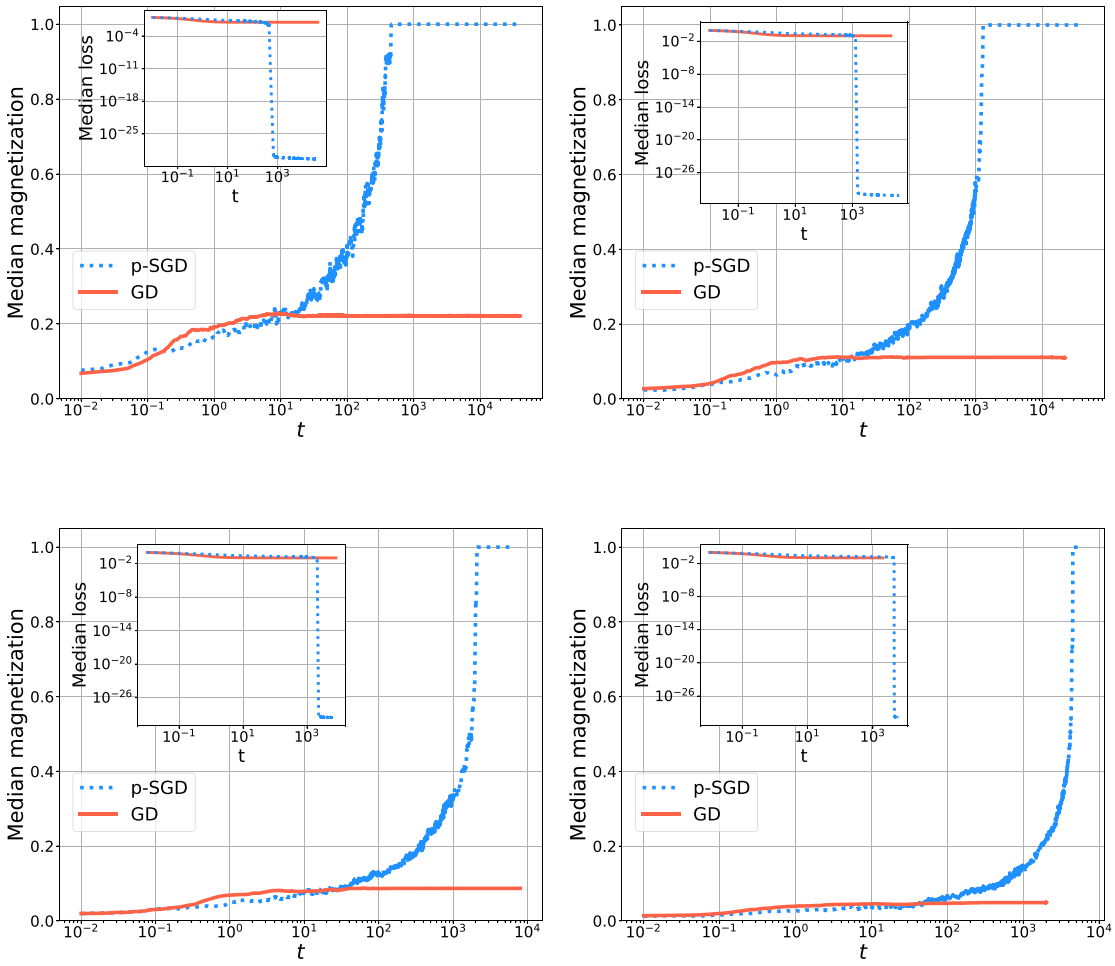

Figure 7. Median magnetization (main plot) and median loss (inset) as a function of time from numerical simulation for the spherical setting at fixed $\alpha = M/N = 2.5$, learning rate $\eta = 0.01$, 100 seeds, and increasing dimension $N = 100$ (above-left), $N = 500$ (above-right), $N = 1000$ (below-left), $N = 2500$ (below-right). We consider random initialization $m_0 = 0$, so the finite initial overlap with the signal is only due to finite size effects. The full red line marks the performance of full-batch gradient descent, while the dotted blue line represents the persistent-SGD algorithm at batch size $\gamma = 0.5$ and persistence time $\tau = 2$.

DMFT framework presented in this work provides the tools to characterize the stochastic dynamics via a detailed analysis of two time quantities, which is a promising direction for future work.

Data availability statement

No new data were created or analysed in this study.

Acknowledgments

We thank Stefano Sarao Mannelli for useful discussions. This work was supported by ERC under the European Union's Horizon 2020 Research and Innovation Programme Grant Agreement No. 714608-SMiLe, and by 'Investissements d'Avenir' LabExPALM (ANR-10-LABX-0039-PALM).

Appendix A. Derivation of DMFT equations

In this section, we provide additional details on the derivation and numerical integration of the theoretical equations describing the learning performance via dynamical mean-field theory (DMFT), presented in section 4 of the main text. The computation is on the line of the one presented in [9, 36]. Here we consider a different loss function, i.i.d. Gaussian input data and labels generated by a teacher vector. We have to take into account the spherical constraint and the additional white noise of the Langevin algorithm. We use the Martin–Siggia–Rose–Janssen–deDominicis (MSRJD) path-integral formalism. We start by writing the

dynamical partition function:

$$1 = Z_{\rm dyn} = \int \mathscr{D}\underline{w} \prod_{i=1}^{N} \delta\left(-\frac{\partial w_i(t)}{\partial t} - \hat{\nu}(t) w_i(t) - \frac{\partial}{\partial w_i} \frac{1}{\ell} \sum_{\mu=1}^{\alpha N} s_\mu(t) v(h_\mu(t); h_\mu^{(0)}) + \varsigma_i(t) \right)$$
$$= \int \mathscr{D}\underline{w} \mathscr{D}\underline{\hat{w}} \times \exp\left[\int {\rm d}t\, i\underline{\hat{w}}(t) \cdot \left(-\frac{\partial \underline{w}(t)}{\partial t} - \hat{\nu}(t)\underline{w}(t) - \frac{\partial}{\partial \underline{w}} \frac{1}{\ell} \sum_{\mu=1}^{\alpha N} s_\mu(t) v(h_\mu(t); h_\mu^{(0)}) + \underline{\varsigma}(t) \right) \right]. \tag{A.1}$$

Since the dynamical partition function is strictly equal to one, we can safely average its expression on the random patterns $\underline{\xi}^\mu$ and on the Langevin noise $\underline{\varsigma}$. Following [36] we can use a supersymmetric formalism to proceed in a compact way. In this case the dynamical variables $\underline{w}(t)$ are merged with their auxiliary fields $\underline{\hat{w}}(t)$ in a superfield involving a couple of Grassmann variables $\theta_a, \bar{\theta}_a$ [41]:

$$\underline{w}(a) = \underline{w}(t) + i\theta_a \bar{\theta}_a \underline{\hat{w}}(t). \tag{A.2}$$

In this way the dynamical partition function (averaged over the Langevin noise) can be written as:

$$Z_{\rm dyn} = \int \mathscr{D}\underline{w}(a) \exp\left[-\frac{1}{2} \int {\rm d}a{\rm d}b\, \underline{w}(a) \mathscr{K}(a,b) \underline{w}(b) + \alpha N \ln \mathscr{Z} \right], \tag{A.3}$$

where

$$\mathscr{Z} = \int \mathscr{D}h(a) \mathscr{D}\hat{h}(a) \int {\rm d}h_0 {\rm d}\hat{h}_0 \exp\left[\mathscr{S}_{\rm loc}\right], \tag{A.4}$$

and the kinetic kernel $\mathscr{K}(a,b)$ is implicitly defined in such a way that

$$-\frac{1}{2} \int {\rm d}a{\rm d}b\, \underline{w}(a) \mathscr{K}(a,b) \underline{w}(b) = -i \int {\rm d}t\, \underline{\hat{w}}(t) \cdot \left(\frac{\partial \underline{w}(t)}{\partial t} + \hat{\nu}(t)\underline{w}(t) - iT\underline{\hat{w}}(t) \right). \tag{A.5}$$

In particular, we have:

$$\begin{aligned} \mathscr{K}(a,b) &= -2T\delta(t_a - t_b) - \theta_a\bar{\theta}_a \partial_{t_a}\delta(t_b - t_a) - \theta_b\bar{\theta}_b \partial_{t_b}\delta(t_a - t_b) + \hat{\nu}(a)\delta(a,b), \\ \hat{\nu}(a) &= \hat{\nu}(t_a), \\ \delta(a,b) &= \delta(t_a - t_b)(\theta_a\bar{\theta}_a - \theta_b\bar{\theta}_b). \end{aligned} \tag{A.6}$$

The local action $\mathscr{S}_{\rm loc}$ is defined as:

$$\mathscr{S}_{\rm loc} = ih_0\hat{h}_0 + i\int {\rm d}a\, \hat{h}(a)h(a) - \frac{1}{2}\left[\hat{h}_0^2 + 2\,\hat{h}_0 \int {\rm d}a\, \hat{h}(a)m(a) + \int {\rm d}a{\rm d}b\, \hat{h}(a)\hat{h}(b)Q(a,b) \right]$$
$$-\frac{1}{\ell}\int {\rm d}a\, s(a) v(h(a); h_0), \tag{A.7}$$

where $s(a) = s(t_a)$, and we have introduced the dynamical order parameters:

$$m(a) = \frac{1}{N}\underline{w}(a) \cdot \underline{w}^{(0)}, \qquad Q(a,b) = \frac{1}{N}\underline{w}(a) \cdot \underline{w}(b). \tag{A.8}$$

Performing the Gaussian integration over the superfields $\underline{w}(a)$, we obtain:

$$\overline{Z_{\rm dyn}} = \int \mathscr{D}Q(a,b) \mathscr{D}m(a)\, e^{N\mathscr{A}_{\rm dyn}}, \tag{A.9}$$

where

$$\mathscr{A}_{\rm dyn} = -\frac{1}{2}\int {\rm d}a{\rm d}b\, \mathscr{K}(a,b)\left[Q(a,b) + m(a)m(b)\right] + \frac{1}{2}\ln\det Q + \alpha \ln \mathscr{Z}_{\rm loc}, \tag{A.10}$$

where

$$\mathscr{Z}_{\rm loc} = \int \frac{{\rm d}h_0}{\sqrt{2\pi}} e^{-h_0^2/2} \int \mathscr{D}h(a) \exp\left[-\frac{1}{2}\int {\rm d}a{\rm d}b\, h(a) Q^{-1}(a,b) h(b) \right.$$
$$\left. -\frac{1}{\ell}\int {\rm d}a\, s(a) v(h(a) + h_0 m(a); h_0) \right]. \tag{A.11}$$

At this point we can evaluate the integral over Q and m through a saddle point computation. The saddle point equations read:

$$0 = \quad -\mathscr{K}(a,b) + Q^{-1}(a,b) + 2\alpha \frac{\delta \ln \mathscr{Z}_{\rm loc}}{\delta Q(a,b)} \tag{A.12}$$

$$0 = \quad -\int \mathrm{d}b\, \mathscr{K}(a,b) m(b) + \alpha \frac{\delta \ln \mathscr{Z}_{\rm loc}}{\delta m(a)}. \tag{A.13}$$

We can evaluate the functional derivatives:

$$\begin{aligned} 2\alpha \frac{\delta \ln \mathscr{Z}_{\rm loc}}{\delta Q(a,b)} &= \frac{\alpha}{\mathscr{E}^2} \langle s(a)s(b)\partial_1 v(h(a)+h_0 m(a);h_0)\partial_1 v(h(b)+h_0 m(b);h_0)\rangle \\ &\quad - \frac{\alpha}{\mathscr{E}} \delta(a,b)\langle s(a)\partial_1^2 v(h(a)+h_0 m(a);h_0)\rangle \equiv M(a,b) - \delta\nu(a)\delta(a,b), \\ \alpha \frac{\delta \ln \mathscr{Z}_{\rm loc}}{\delta m(a)} &= -\frac{\alpha}{\mathscr{E}} \langle h_0 s(a)\partial_1 v(h(a)+h_0 m(a);h_0)\rangle, \end{aligned} \tag{A.14}$$

where we have defined:

$$\begin{aligned} M(a,b) &= \frac{\alpha}{\mathscr{E}^2} \langle s(a)s(b)\partial_1 v(h(a)+h_0 m(a);h_0)\partial_1 v(h(b)+h_0 m(b);h_0)\rangle, \\ \delta\nu(a) &= \frac{\alpha}{\mathscr{E}} \langle s(a)\partial_1^2 v(h(a)+h_0 m(a);h_0)\rangle. \end{aligned} \tag{A.15}$$

The average in brackets denotes the average with the following measure:

$$\begin{aligned} \langle \bullet \rangle = \int \frac{\mathrm{d}h_0}{\sqrt{2\pi}} \int \mathscr{D}h(a)\, \bullet \\ \times \exp\left[-\frac{h_0^2}{2} - \frac{1}{2}\int \mathrm{d}a\mathrm{d}b\, h(a) Q^{-1}(a,b) h(b) - \frac{1}{\mathscr{E}} \int \mathrm{d}a\, s(a) v(h(a)+h_0 m(a);h_0) \right]. \end{aligned} \tag{A.16}$$

Along the lines of [36], we can rewrite the average in equation (A.16) as an average over $h_0 \sim \mathscr{N}(0,1)$, the variables $s(t)$ defined in equation (7) of the main text, and an effective stochastic process:

$$\partial_t h(t) = -\tilde{\nu}(t) h(t) - \frac{1}{\mathscr{E}} s(t)\partial_1 v(\tilde{h}(t);h_0) + \int_0^t \mathrm{d}t'\, M_R(t,t') h(t') + \chi(t), \tag{A.17}$$

with initial condition $P(h(0)) = e^{-h(0)^2/2(1-m_0^2)}/\sqrt{2\pi(1-m_0^2)}$, where $\chi(t)$ is an effective Gaussian noise:

$$\langle \chi(t) \rangle = 0, \qquad \langle \chi(t)\chi(t') \rangle = 2T\delta(t-t') + M_C(t,t'), \tag{A.18}$$

and we have defined the following auxiliary functions:

$$\begin{aligned} \tilde{h}(t) &\equiv h(t) + h_0 m(t), \\ \delta\nu(t) &= \frac{\alpha}{\mathscr{E}} \langle s(t)\partial_1^2 v(\tilde{h}(t);h_0)\rangle, \\ \hat{\nu}(t) &= -\frac{\alpha}{\mathscr{E}} \langle s(t)\tilde{h}(t)\partial_1 v(\tilde{h}(t);h_0)\rangle + T, \\ \tilde{\nu}(t) &= \hat{\nu}(t) + \delta\nu(t). \end{aligned} \tag{A.19}$$

The expression for the Langrange multiplier $\hat{\nu}(t)$ is obtained enforcing the spherical constraint $\sum_{i=1}^N \mathrm{d}w_i^2/\mathrm{d}t = 0$ by applying Itô's formula to equation (13) of the main text. The kernels $M_C(t,t')$ and $M_R(t,t')$ are obtained expanding $M(a,b)$:

$$\begin{aligned} M(a,b) &= M_C(t_a,t_b) + \theta_a\bar{\theta}_a M_R(t_b,t_a) + \theta_b\bar{\theta}_b M_R(t_a,t_b), \\ M_C(t,t') &= \frac{\alpha}{\mathscr{E}^2} \langle s(t)s(t')\partial_1 v(\tilde{h}(t);h_0)\partial_1 v(\tilde{h}(t');h_0))\rangle, \\ M_R(t,t') &= \frac{\alpha}{\mathscr{E}^2} \langle s(t)s(t')\partial_1 v(\tilde{h}(t);h_0)\partial_1^2 v(\tilde{h}(t');h_0)\, i\hat{h}(t')\rangle \\ &= \frac{\alpha}{\mathscr{E}^2} \frac{\delta}{\delta P(t')} \langle s(t)\partial_1 v(\tilde{h}(t);h_0)\rangle \Big|_{P=0}. \end{aligned} \tag{A.20}$$

The variable $P(t')$ indicates a linear perturbation applied on the gap variable h at time t' and then set to zero. The kernel $M_R(t,t')$ can be also expressed as:

$$M_R(t,t') = \frac{\alpha}{\mathscr{C}^2}\langle s(t)\partial_1^2 v(\tilde{h}(t);h_0)T(t,t')\rangle, \tag{A.21}$$

where $T(t,t') = \delta h(t)/\delta P(t')$ satisfies,

$$\partial_t T(t,t') = -\tilde{\nu}(t)T(t,t') - \frac{1}{\mathscr{C}}s(t)\partial_1^2 v(\tilde{h}(t);h_0)\left(T(t,t') - \delta(t,t')\right) + \int_{t'}^{t} \mathrm{d}s M_R(t,s)T(s,t'). \tag{A.22}$$

Furthermore, from equation (A.13) we get the behavior of the magnetization $m(t)$ as a function of time:

$$\partial_t m(t) = -\hat{\nu}(t)m(t) - \mu(t), \qquad m(0) = m_0, \tag{A.23}$$

where m_0 is defined in equation (11) of the main text and:

$$\mu(t) = \frac{\alpha}{\mathscr{C}}\langle s(t)h_0\, \partial_1 v(\tilde{h}(t);h_0)\rangle. \tag{A.24}$$

Moreover, setting $m(t) = 0$ one gets a set of equations that coincide with [36]. From the solution $Q(a,b)$ of the saddle point equation (A.12), we can obtain the equations for the dynamical correlation function $C(t,t') = \sum_i w_i(t)w_i(t')/N$ and the response $R(t,t') = \sum_i \delta w_i(t)/\delta H_i(t')/N$ to a linear perturbation of the weights by an infinitesimal local field $H_i(t)$. Indeed, we can write the closure relation:

$$\begin{aligned}\delta(a,b) &= \int \mathrm{d}c\, Q^{-1}(a,c)Q(c,b)\\ &= \int \mathrm{d}c\, \left[\mathscr{K}(a,c) - \mathscr{M}(a,c)\right]Q(c,b) + \delta\nu(a)Q(a,b).\end{aligned} \tag{A.25}$$

Now we can express the overlap explicitly in time and Grassman coordinates:

$$Q(a,b) = \frac{1}{N}\underline{w}(a)\cdot\underline{w}(b) = C(t_a,t_b) - m(t_a)m(t_b) + \theta_a\bar{\theta}_a R(t_b,t_a) + \theta_b\bar{\theta}_b R(t_a,t_b), \tag{A.26}$$

where we remind that in (A.10) we have performed the change of variable $Q(a,b) \to Q(a,b) + m(a)m(b)$. Plugging equation (A.6) and equation (A.20) in equation (A.25), we find:

$$\begin{aligned}\delta(t_a - t_b)(\theta_a\bar{\theta}_a - \theta_b\bar{\theta}_b) =& -2TR(t_b,t_a) + \partial_{t_a}C(t_a,t_b)\\ &- \partial_{t_a}m(t_a)m(t_b) + \hat{\nu}(t_a)\left(C(t_a,t_b) - m(t_a)m(t_b)\right)\\ &- \int \mathrm{d}t_c\, \left[M_C(t_a,t_c)R(t_b,t_c) + M_R(t_a,t_c)\left(C(t_b,t_c) - m(t_b)m(t_c)\right)\right]\\ &+ \theta_a\bar{\theta}_a\left[\partial_{t_a}R(t_b,t_a) + \hat{\nu}(t_a)R(t_b,t_a)\right] - \theta_a\bar{\theta}_a\int \mathrm{d}t_c M_R(t_c,t_a)R(t_b,t_c)\\ &+ \theta_b\bar{\theta}_b\left[\partial_{t_a}R(t_a,t_b) + \hat{\nu}(t_a)R(t_a,t_b) - \int \mathrm{d}t_c M_R(t_a,t_c)R(t_c,t_b)\right]\\ &+ \delta\nu(t_a)\left(C(t_a,t_b) - m(t_a)m(t_b) + \theta_a\bar{\theta}_a R(t_b,t_a) + \theta_b\bar{\theta}_b R(t_a,t_b)\right).\end{aligned} \tag{A.27}$$

We can derive two equations from the scalar and Grassman terms (the terms in $\theta_a\bar{\theta}_a$ and $\theta_b\bar{\theta}_b$ result in the same contribution):

$$\begin{aligned}\partial_t C(t',t) =& -\tilde{\nu}(t)C(t,t') + 2TR(t',t) + \int_0^t \mathrm{d}s M_R(t,s)C(t',s) + \int_0^{t'} \mathrm{d}s M_C(t,s)R(t',s)\\ &- m(t')\left(\int_0^t \mathrm{d}s M_R(t,s)m(s) + \mu(t) - \delta\nu(t)m(t)\right) \quad \text{if } t \neq t',\\ \partial_t R(t,t') =& -\tilde{\nu}(t)R(t,t') + \delta(t-t') + \int_{t'}^t \mathrm{d}s M_R(t,s)R(s,t'),\end{aligned} \tag{A.28}$$

where we have used equation (A.23) in the first of equation (A.28). An alternative expression to equation (A.19) for the Lagrange multiplier $\hat{\nu}(t)$ can be obtained by plugging $C(t,t) = 1$ in the first of equations (A.28):

$$\begin{aligned}\hat{\nu}(t) = &-\delta\nu(t) + T + \int_0^t \mathrm{d}s\,(M_R(t,s)C(t,s) + M_C(t,s)R(t,s)) \\ &- m(t)\left(\mu(t) - \delta\nu(t)m(t) + \int_0^t \mathrm{d}s M_R(t,s)m(s)\right).\end{aligned} \tag{A.29}$$

A.1. Numerical integration of the DMFT equations

In this section, we provide more details on the numerical integration of DMFT equations. Similarly as in [9], we implement an iterative scheme to reach the convergence of the self-consistent process in equation (A.17).

- We start from a simple guess of the kernels in equation (A.20) and the auxiliary functions in (A.19). In particular, we set $m(t) = m_0\ \forall t, M_R(t,t') = 0\ \forall t,t', M_C(t,t) = M_C(0,0)\ \forall t, M_C(t,t') = 0.1 \times M_C(0,0)\ \forall t \neq t'$, and we initialize all the entries of $\tilde{\nu}(t)$ and $\mu(t)$ to their value at $t = 0$.
- We use the previous guess to generate multiple realizations of the curve $h(t)$.
- We update the kernels and auxiliary functions, computing the averages over $h(t)$, h_0, $s(t)$. We introduce a damping in the update to control the oscillations. We integrate equation (A.23) to obtain the magnetization $m(t)$.
- We repeat the above procedure until the kernels and auxiliary functions reach a fixed point.

We use equation (A.29) in order to compute the Lagrange multiplier $\hat{\nu}(t)$ because we find that it is more stable to fluctuations. We integrate equation (A.22) to compute the kernel M_R. We typically use a discrete time step $dt = 10^{-3} - 10^{-2}$.

Appendix B. Generalization error

In this section, we sketch the computation of the average generalization error in the phase retrieval problem under consideration. Given a previously unseen data point $\underline{\xi}_{\text{new}} \sim \mathcal{N}(\underline{0}, \underline{\underline{I}}_N)$, the generalization error can be defined for a generic error function $f: \mathbb{R}^2 \to \mathbb{R}$, taking as first argument the true label and as second argument the estimated one. The average generalization error is then:

$$\varepsilon_{\text{gen}} = \mathbb{E}_{\{\underline{\xi}_\mu\}_{\mu=1}^M, \underline{\xi}_{\text{new}}, \underline{w}^{(0)}}\left[f(y_{\text{new}}, \hat{y}_{\text{new}})\right], \tag{B.1}$$

where $y_{\text{new}} = \left|\frac{1}{\sqrt{N}}\underline{\xi}_{\text{new}} \cdot \underline{w}^{(0)}\right|$ is the true label, $\hat{y}_{\text{new}} = \left|\frac{1}{\sqrt{N}}\underline{\xi}_{\text{new}} \cdot \underline{w}\right|$ is the estimated one, and the weight vector $\underline{w}$ implicitly depends on the training set $\{\underline{\xi}_\mu\}_{\mu=1}^M$ as well as on the hidden signal $\underline{w}^{(0)}$. We can introduce Dirac's δ-functions to rewrite:

$$\begin{aligned}\varepsilon_{\text{gen}} =& \mathbb{E}_{\{\underline{\xi}_\mu\}_{\mu=1}^M, \underline{\xi}_{\text{new}}, \underline{w}^{(0)}} \int_{-\infty}^{+\infty} \mathrm{d}x \int_{-\infty}^{+\infty} \mathrm{d}z f(|x|,|z|) \\ &\times \delta\left(x - \frac{1}{\sqrt{N}}\underline{\xi}_{\text{new}} \cdot \underline{w}^{(0)}\right) \delta\left(z - \frac{1}{\sqrt{N}}\underline{\xi}_{\text{new}} \cdot \underline{w}\right) \\ =& \mathbb{E}_{\{\underline{\xi}_\mu\}_{\mu=1}^M, \underline{\xi}_{\text{new}}, \underline{w}^{(0)}} \int_{-\infty}^{+\infty} \frac{\mathrm{d}x\mathrm{d}\hat{x}}{2\pi} \int_{-\infty}^{+\infty} \frac{\mathrm{d}z\mathrm{d}\hat{z}}{2\pi} f(|x|,|z|) \\ &\times \exp\left(i\hat{x}x + i\hat{z}z - \frac{i}{\sqrt{N}}\underline{\xi}_{\text{new}} \cdot \left(\hat{x}\underline{w}^{(0)} + \hat{z}\underline{w}\right)\right),\end{aligned} \tag{B.2}$$

where we have substituted the δ-functions with their Fourier representation. We first compute the average over the new sample $\underline{\xi}_{\text{new}}$, that is independent both of $\underline{w}^{(0)}$ and $\underline{w}$:

$$\begin{aligned}\varepsilon_{\text{gen}} =& \mathbb{E}_{\{\underline{\xi}_\mu\}_{\mu=1}^M, \underline{w}^{(0)}} \left[\int_{-\infty}^{+\infty} \frac{\mathrm{d}x\mathrm{d}\hat{x}}{2\pi} \int_{-\infty}^{+\infty} \frac{\mathrm{d}z\mathrm{d}\hat{z}}{2\pi} f(|x|,|z|)\right. \\ &\left.\times \exp\left(i\hat{x}x + i\hat{z}z - \frac{1}{2}\hat{x}^2\frac{\underline{w}^{(0)} \cdot \underline{w}^{(0)}}{N} - \frac{1}{2}\hat{z}^2\frac{\underline{w} \cdot \underline{w}}{N} - \hat{x}\hat{z}\frac{\underline{w}^{(0)} \cdot \underline{w}}{N}\right)\right].\end{aligned} \tag{B.3}$$

In the following, we denote:

$$q_0 = \frac{\underline{w}^{(0)} \cdot \underline{w}^{(0)}}{N}, \qquad q = \frac{\underline{w} \cdot \underline{w}}{N}, \qquad m = \frac{\underline{w}^{(0)} \cdot \underline{w}}{N}. \tag{B.4}$$

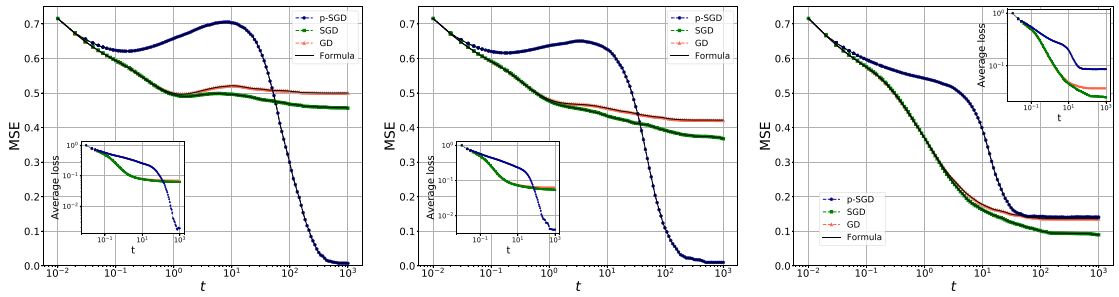

Figure B.1. Mean squared error (main plot) and average training loss (inset) as a function of time, at dimension $N = 1000$, $\alpha = M/N = 3$, warm start $m_0 = 0.2$, learning rate $\eta = 0.01$, for increasing values of regularization $\lambda = 0$ (left), $\lambda = 0.1$ (center) and $\lambda = 1$ (right). We consider full-batch gradient descent (red triangles), multi-pass SGD (green squares) and its persistent version (blue dots). For the last two algorithms, we take a mini-batch size $\mathscr{C} = 0.5$. The symbols mark the average over simulations, while the black line shows the prediction from equation (B.8), where the overlaps q_0, q, and m are obtained from simulations. The simulations are run over 250 seeds and at each time a new landscape is generated. In both panels, the gradient is rescaled by the fraction of samples in the mini-batch at each iteration.

By integrating over the conjugate variables $\hat{x}$ and $\hat{z}$, we obtain:

$$\varepsilon_{\text{gen}} = \mathbb{E}_{\{\underline{\xi}_\mu\}_{\mu=1}^M, \underline{w}^{(0)}, z, x} \left[f(|x|, |z|) \right], \tag{B.5}$$

where $z \sim \mathcal{N}(0, q)$ and $x \sim \mathcal{N}(my/q, q_0 - m^2/q)$. In the infinite dimensional limit, q_0, q, and m concentrate to their average value, therefore we simply obtain:

$$\varepsilon_{\text{gen}} = \mathbb{E}_{z,x} \left[f(|x|, |z|) \right], \tag{B.6}$$

where now the quantities q_0, q, m are intended in the infinite dimensional limit. This computation shows that the generalization error depends on the signal and the training set only through q_0, q, m. In particular, in the spherical case $q_0 = q = 1$ and the performance depends only on m.

Mean squared error

As a measure of the error, we can consider for instance the commonly-used mean squared error, here defined as:

$$\text{MSE}(y, \hat{y}) = \mathbb{E}(y - \hat{y})^2. \tag{B.7}$$

From equation (B.5), we obtain that the mean squared error between the true label of a new sample and its estimate—in the infinite dimensional limit—is:

$$\text{MSE} = q + q_0 - \frac{4}{\pi} \left[\sqrt{q q_0 - m^2} + m \arctan \left(\frac{m}{\sqrt{q q_0 - m^2}} \right) \right], \tag{B.8}$$

which in the spherical case is a monotonically decreasing function of m. Figure B.1 shows the mean squared error for GD, multi-pass SGD and its persistent version in the case of ridge regularization. The dots mark the average from simulations, while the black line displays the prediction obtained from equation (B.8), computed in the average values of q_0, q and m from simulations at dimension $N = 1000$. We find a very good agreement between theory and simulations.

Appendix C. Ridge regularization

In this section, we consider a variant of the training algorithm presented in equations (5), (8), and (9) of the main text, where instead of projecting the weights on the hyper sphere $|\underline{w}(t)|^2 = N$ at each iteration, we apply a ridge regularization of strength λ. The parameter λ is fixed during training and can be tuned *a posteriori*, e.g. by cross-validation. The flow dynamics given by equation (13) of the main text is modified as follows:

$$\frac{\partial w_i(t)}{\partial t} = -\lambda w_i(t) + \varsigma_i(t) - \frac{1}{\mathscr{C}} \sum_{\mu=1}^{\alpha N} s_\mu(t) \partial_1 v(h_\mu(t); h_\mu^{(0)}) \frac{1}{\sqrt{N}} \xi_i^\mu. \tag{C.1}$$

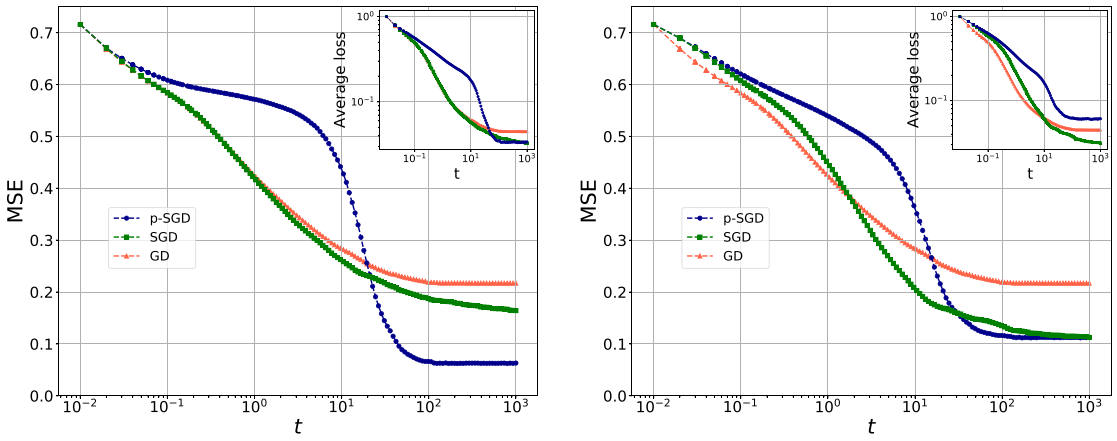

Figure B.2. Mean squared error (main plot) and average training loss (inset) as a function of time, at dimension $N = 1000$, $\alpha = M/N = 3$, warm start $m_0 = 0.2$, learning rate $\eta = 0.01$, fixed regularization $\lambda = 0.5$. We consider full-batch gradient descent (red triangles), multi-pass SGD (green squares) and its persistent version (blue dots). For the last two algorithms, we take a mini-batch size $\mathscr{E} = 0.5$. The simulations are run over 250 seeds and at each time a new landscape is generated. The left panel depicts the performance of the three algorithms when the gradient is rescaled by the fraction of training samples in the mini-batch $\mathscr{E}$, as defined in equation (C.1). The right panel illustrates the case in which the gradient is not rescaled by the fraction of samples used to compute it.

Note that this change simply amounts to substituting the time-dependent Lagrange multiplier $\hat{\nu}(t)$ with the constant λ. All the other variables are defined in the main text and stay the same. We modify accordingly the initial condition (defined by equation (11) of the main text) as follows:

$$\underline{w}(t=0) = m_0\,\underline{w}^{(0)} + \underline{z} \in \mathbb{R}^N, \tag{C.2}$$

where m_0 is the average initial magnetization and $\underline{z} \sim \mathcal{N}(\underline{0}, \underline{I}_N)$. This change is reflected in the initial condition for the effective stochastic process in equation (A.17), that becomes: $P(h(0)) = e^{-h(0)^2/2}/\sqrt{2\pi}$. We still consider a teacher on the hyper sphere $|\underline{w}^{(0)}|^2 = N$.

C.1. Results

In this section, we discuss the results obtained for ridge regularization. The behavior of the GD-based algorithms is qualitatively the same as what we have observed for the spherical case in the main text: stochasticity is beneficial for generalization also in the case of ridge regularization and without any regularization.

Figure B.1 illustrates the performance of GD, multi-pass SGD and persistent SGD for three different values of regularization strength: $\lambda = 0$ (left panel), $\lambda = 0.1$ (central panel), $\lambda = 1$ (right panel), fixing the values of all the other control parameters. The generalization performance is evaluated by measuring the MSE and the average training loss is shown in the inset. We observe that the effect of ridge regularization is different on SGD and persistent-SGD: while the former benefits from a finite regularization $\lambda = 1$, the latter generalizes better at low values of λ. Evaluating the optimal regularization is beyond the scope of this work. Furthermore, the left and central panels of figure B.1 display a peculiar phenomenon of *double descent* of the generalization error as a function of time that has also been observed in real data [42].

Figure B.2 depicts the MSE as a function of time for the three algorithms under consideration at fixed regularization ($\lambda = 0.5$) and for two different dynamics. In particular, we illustrate the effect of rescaling the gradient by the fraction of samples in the mini batch ($\mathscr{E}$) on the dynamics. In the left panel, the gradient is rescaled by $\mathscr{E}$, while in the right panel we do not rescale it. We observe that, while the rescaling is beneficial for persistent-SGD, SGD performs better without it. At variance with the spherical case considered in the main text, in the case of ridge regularization of fixed strength λ rescaling the gradient by $\mathscr{E}$ does not result in a simple rescaling of the learning rate. Instead, the regularization is also affected.

Appendix D. Additional figures

In this section we provide additional figures in support to our observations in sections 3 and 5 of the main text. All the figures illustrate the spherical case treated in the main text. Therefore, the generalization performance is entirely captured by the magnetization.

Figure B.3 compares the average magnetization (left panel) and loss (right panel) as a function of training time for GD, SGD and persistent-SGD for decreasing values of the learning rate. We observe that, in

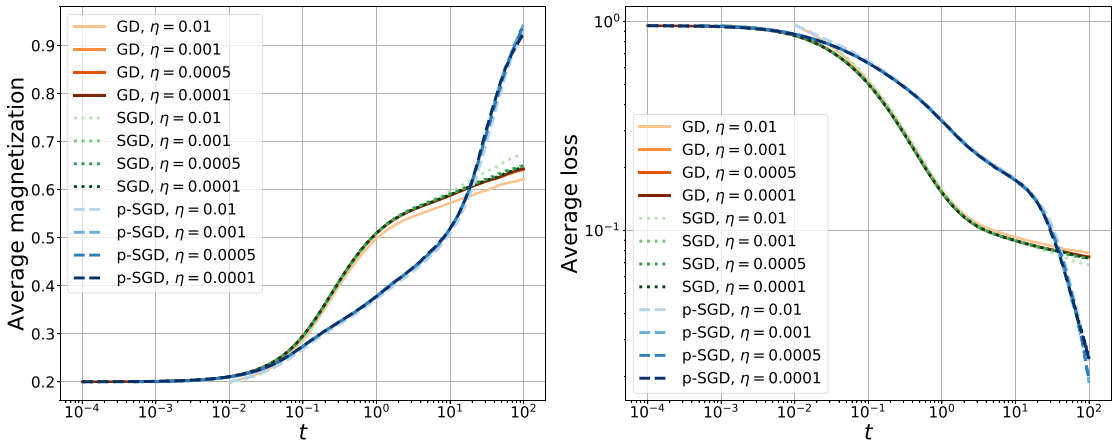

Figure B.3. Average magnetization (right) and average loss (less) as a function of training time for the three algorithms: GD (full red lines), SGD (dotted green lines) and persistent-SGD (dashed blue lines). The numerical simulations are run at fixed $\alpha = M/N = 3$, warm start $m_0 = 0.2$ and input dimension $N = 1000$, over 250 seeds. The stochastic algorithms are run at fixed batch size $\mathscr{b} = 0.5$. We consider decreasing values of learning rate $\eta = 0.01, 0.001, 0.0005, 0.0001$, depicted with increasing color intensity. For visibility purposes, we plot $t + \eta$ on the x-axes.

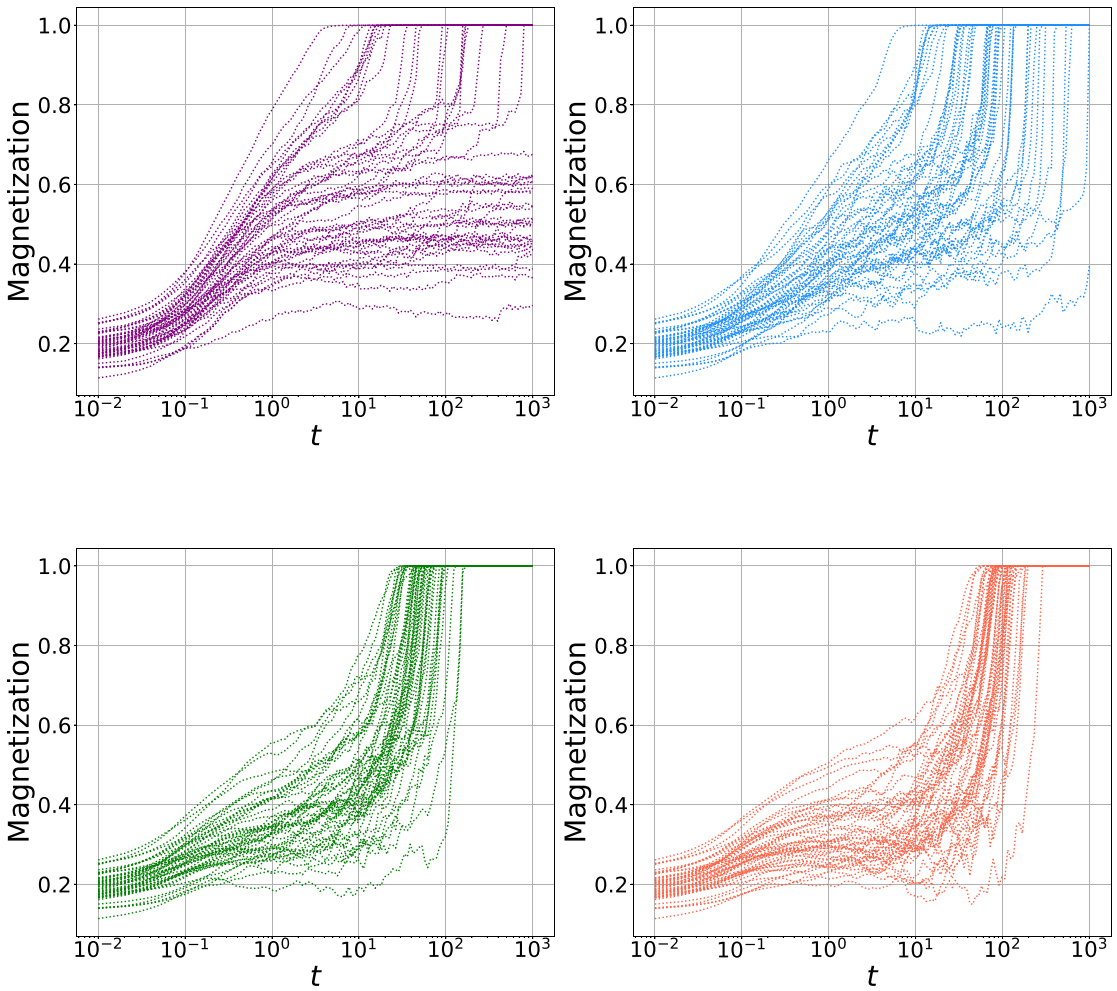

Figure B.4. Instances of the magnetization as a function of time from numerical simulations for the persistent SGD algorithm at fixed $\alpha = M/N = 3$, batch size $\mathscr{b} = 0.5$ and warm initialization $m_0 = 0.2$. We consider the model with spherical constraint defined in section 2 of the main text. We consider four different values of the persistence time: $\tau = 0.05$ (upper left), $\tau = 0.5$ (upper right), $\tau = 2$ (lower left), $\tau = 5$ (lower right). For each panel, we show 50 different seeds, corresponding to different realizations of the landscape and initial weights. The simulations are run at dimension $N = 1000$ and learning rate $\eta = 0.01$.

the limit of small learning rate, the learning curves of SGD collapse to the ones of GD. On the contrary, the persistent-SGD algorithm has a well-defined continuous time limit that is different than the one of full batch GD.

Figure B.4 summarizes the effect of increasing the persistence time on the performance of the persistent-SGD algorithm. We show the instances of the magnetization as a function of time—corresponding to 50 different realizations of the problem landscape and initializations of the weight vector. We consider increasing values of the parameter $\tau = 0.05$ (upper left panel), $\tau = 0.5$ (upper right panel), $\tau = 2$ (lower left panel), and $\tau = 5$ (lower right panel), at a fixed ratio $\alpha = 3$ of training samples over input dimensions, batch size $\ell = 0.5$ and warm initialization $m_0 = 0.2$. On the one hand, we observe that increasing the persistence time gradually diminishes the number of seeds that get stuck at intermediate plateau, resulting in an improved generalization performance. On the other hand, until time $t \sim \tau$ the samples in the mini-batch have not been reshuffled yet (on average). Therefore, for large values of τ the plateaus disappear but the magnetization is stuck at the beginning of the training and only at training time $t > \tau$ it has a sudden increase.

ORCID iDs

Francesca Mignacco https://orcid.org/0000-0001-9944-2498
Pierfrancesco Urbani https://orcid.org/0000-0002-4722-6811

References

[1] Liu S, Papailiopoulos D and Achlioptas D 2019 Bad global minima exist and SGD can reach them (arXiv:1906.02613)
[2] Abbe E and Sandon C 2020 Poly-time universality and limitations of deep learning (arXiv:2001.02992)
[3] HaoChen J Z, Wei C, Lee J D, and Ma T 2020 Shape matters: understanding the implicit bias of the noise covariance (arXiv:2006.08680)
[4] Cheng X, Yin D, Bartlett P and Jordan M 2020 Stochastic gradient and Langevin processes *Int. Conf. Machine Learning* (PMLR) pp 1810–19
[5] Li Q, Tai C and Weinan E 2017 Stochastic modified equations and adaptive stochastic gradient algorithms *Int. Conf. Machine Learning* (PMLR) pp 2101–10
[6] Jastrzebski S, Kenton Z, Arpit D, Ballas N, Fischer A, Bengio Y and Storkey A 2017 Three factors influencing minima in SGD *Artificial Neural Networks and Machine Learning, ICANN* (arXiv:1711.04623)
[7] Hu W, Li C J, Li L and Liu J-G 2019 On the diffusion approximation of nonconvex stochastic gradient descent *Ann. Math. Sci. Appl.* **4** 3–32
[8] Simsekli U, Sagun L and Gurbuzbalaban M 2019 A tail-index analysis of stochastic gradient noise in deep neural networks *Int. Conf. Machine Learning* (PMLR) pp 5827–37
[9] Mignacco F, Krzakala F, Urbani P and Zdeborová L 2020 Dynamical mean-field theory for stochastic gradient descent in Gaussian mixture classification *NeurIPS 2020*
[10] Walther A 1963 The question of phase retrieval in optics *Opt. Acta: Int. J. Opt.* **10** 41–9
[11] Millane R P 1990 Phase retrieval in crystallography and optics *J. Opt. Soc. Am.* A **7** 394–411
[12] Balan R, Casazza P and Edidin D 2006 On signal reconstruction without phase *Appl. Comput. Harmon. Anal.* **20** 345–56
[13] Corbett J 2006 The Pauli problem, state reconstruction and quantum-real numbers *Rep. Math. Phys.* **57** 53–68
[14] Maillard A, Arous G B and Biroli G 2020 Landscape complexity for the empirical risk of generalized linear models *Mathematical and Scientific Machine Learning* (PMLR) pp 287–327
[15] Barbier J, Krzakala F, Macris N, Miolane L and Zdeborová L 2019 Optimal errors and phase transitions in high-dimensional generalized linear models *Proc. Natl Acad. Sci.* **116** 5451–60
[16] Mondelli M and Montanari A 2018 Fundamental limits of weak recovery with applications to phase retrieval *Conf. Learning Theory* (PMLR) pp 1445–50
[17] Ma J, Xu J and Maleki A 2019 Optimization-based amp for phase retrieval: the impact of initialization and l2 regularization *IEEE Trans. Inf. Theory* **65** 3600–29
[18] Dong J, Krzakala F and Gigan S 2019 Spectral method for multiplexed phase retrieval and application in optical imaging in complex media *ICASSP 2019-2019 IEEE Int. Conf. Acoustics, Speech and Signal Processing (ICASSP)* pp 4963–7
[19] Luo W, Alghamdi W and Lu Y M 2019 Optimal spectral initialization for signal recovery with applications to phase retrieval *IEEE Trans. Signal Process.* **67** 2347–56
[20] Chen Y, Chi Y, Fan J and Ma C 2019 Gradient descent with random initialization: fast global convergence for nonconvex phase retrieval *Math. Program.* **176** 5–37
[21] Mannelli S S, Biroli G, Cammarota C, Krzakala F, Urbani P, and Zdeborová L 2020a Complex dynamics in simple neural networks: understanding gradient flow in phase retrieval *NeurIPS 2020*
[22] Gerace F, Loureiro B, Krzakala F, Mezard M and Zdeborova L 2020 Generalisation error in learning with random features and the hidden manifold model *Int. Conf. Machine Learning* (PMLR) pp 3452–62
[23] Cai J, Huang M, Li D, and Wang Y 2021 Solving phase retrieval with random initial guess is nearly as good as by spectral initialization (arXiv:2101.03540)
[24] Mannelli S S, Vanden-Eijnden E and Zdeborová L 2020b Optimization and generalization of shallow neural networks with quadratic activation functions *NeurIPS*
[25] Ma C, Wang K, Chi Y and Chen Y 2018 Implicit regularization in nonconvex statistical estimation: gradient descent converges linearly for phase retrieval and matrix completion *Int. Conf. Machine Learning* (PMLR) pp 3345–54
[26] Tan Y S and Vershynin R 2019 Online stochastic gradient descent with arbitrary initialization solves non-smooth, non-convex phase retrieval (arXiv:1910.12837)

[27] Mondelli M, Thrampoulidis C and Venkataramanan R 2020 Optimal combination of linear and spectral estimators for generalized linear models (arXiv:2008.03326)
[28] Ben Arous G, Gheissari R, and Jagannath A 2020 A classification for the performance of online SGD for high-dimensional inference (arXiv e-prints 2003)
[29] Mézard M, Parisi G and Virasoro M-A 1986 *World Sci. Lecture Notes Phys.* vol 9 (Singapore: World Scientific Publishing Company) p 476
[30] Georges A, Kotliar G, Krauth W and Rozenberg M J 1996 Dynamical mean-field theory of strongly correlated fermion systems and the limit of infinite dimensions *Rev. Mod. Phys.* **68** 13
[31] Parisi G, Urbani P and Zamponi F 2020 *Theory of Simple Glasses: Exact Solutions in Infinite Dimensions* (Cambridge: Cambridge University Press)
[32] Sompolinsky H and Zippelius A 1981 Dynamic theory of the spin-glass phase *Phys. Rev. Lett.* **47** 359–62
[33] Cugliandolo L F and Kurchan J 1993 Analytical solution of the off-equilibrium dynamics of a long-range spin-glass model *Phys. Rev. Lett.* **71** 173–6
[34] Sompolinsky H, Crisanti A and Sommers H J 1988 Chaos in random neural networks *Phys. Rev. Lett.* **61** 259–62
[35] Arous G B *et al* 1997 Symmetric Langevin spin glass dynamics *Ann. Probab.* **25** 1367–422
[36] Agoritsas E, Biroli G, Urbani P and Zamponi F 2018 Out-of-equilibrium dynamical mean-field equations for the perceptron model *J. Phys. A: Math. Theor.* **51** 085002
[37] Eissfeller H and Opper M 1992 New method for studying the dynamics of disordered spin systems without finite-size effects *Phys. Rev. Lett.* **68** 2094
[38] Eissfeller H and Mean-field Monte O, M 1994 Carlo approach to the Sherrington-Kirkpatrick model with asymmetric couplings *Phys. Rev.* E **50** 709
[39] Zhu Z, Jingfeng W, Bing Y, Lei W and Jinwen M 2019 The anisotropic noise in stochastic gradient descent: its behavior of escaping from sharp minima and regularization effects *Int. Conf. Machine Learning*
[40] Keskar N S, Mudigere D, Nocedal J, Smelyanskiy M, and Tang P T P 2017 On large-batch training for deep learning: generalization gap and sharp minima (arXiv:1609.04836)
[41] Zinn-Justin J 1996 *Quantum Field Theory and Critical Phenomena* (Oxford: Clarendon)
[42] Nakkiran P, Kaplun G, Bansal Y, Yang T, Barak B and Sutskever I 2020 Deep double descent: where bigger models and more data hurt *Int. Conf. Learning Representations*