On the generalization of learning algorithms that do not converge

Nisha Chandramoorthy
Institute for Data, Systems and Society
Massachusetts Institute of Technology
nishac@mit.edu

Andreas Loukas
Prescient Design
Genentech, Roche
andreas.loukas@roche.com

Khashayar Gatmiry
Electrical Engineering and Computer Science
Massachusetts Institute of Technology
gatmiry@mit.edu

Stefanie Jegelka
Electrical Engineering and Computer Science
Massachusetts Institute of Technology
stefje@mit.edu

Abstract

Generalization analyses of deep learning typically assume that the training converges to a fixed point. But, recent results indicate that in practice, the weights of deep neural networks optimized with stochastic gradient descent often oscillate indefinitely. To reduce this discrepancy between theory and practice, this paper focuses on the generalization of neural networks whose training dynamics do not necessarily converge to fixed points. Our main contribution is to propose a notion of statistical algorithmic stability (SAS) that extends classical algorithmic stability to non-convergent algorithms and to study its connection to generalization. This ergodic-theoretic approach leads to new insights when compared to the traditional optimization and learning theory perspectives. We prove that the stability of the time-asymptotic behavior of a learning algorithm relates to its generalization and empirically demonstrate how loss dynamics can provide clues to generalization performance. Our findings provide evidence that networks that "train stably generalize better" even when the training continues indefinitely and the weights do not converge.

1 Introduction

It is common practice that when the training loss of a neural networks converges close to some value, the learning algorithm—typically some variant of Stochastic Gradient Descent (SGD)—is terminated. Perhaps surprisingly, recent works indicate that the network function at termination time is typically not a fixed point of the learning algorithm: if run longer, the gradient norm does not vanish and the learning algorithm outputs functions with significantly different parameters [Cohen et al., 2021, Zhang et al., 2022, Lobacheva et al., 2021]. This observation stands in contrast to common generalization analyses that assume convergence of the training algorithm to a fixed point. It also raises the question of when (and why) non-convergent learning algorithms should be expected to generalize.

Standard approaches for obtaining generalization bounds find ways to bound the complexity measure of the function class expressible by a neural network in a data-independent e.g., [Vapnik, 1999, Bartlett et al., 2017, Neyshabur et al., 2018, Golowich et al., 2018, Bartlett et al., 2019, Arora et al., 2018] or data-dependent manner, e.g., [von Luxburg and Bousquet, 2004, Xu and Mannor, 2012, Sokolic et al., 2017]. Nevertheless, even with recent technical improvements, by and large these

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

approaches do not explicitly account for the relationship between generalization and the dynamics of the learning algorithm.

A different approach to generalization analysis yields algorithm-dependent bounds by connecting the generalization performance of the trained network to the stability of the training to data perturbations [Bousquet and Elisseeff, 2002, Feldman and Vondrak, 2018, Kuzborskij and Lampert, 2018, Bousquet et al., 2020, Zhang et al., 2021b, Rakhlin, 2006]. A key claim of these works has been that networks that are trained fast generalize better. This claim, though intuitively meaningful, hinges on the premise of a convergent algorithm, which deviates from the observed behavior of deep neural network training. Further work has been done for settings where the algorithm provably converges to a fixed point, such as two-layer overparameterized networks, deep linear networks, certain matrix factorizations, etc. [Frei et al., 2019, Allen-Zhu et al., 2019, Soudry et al., 2018, Arora et al., 2019b, Gunasekar et al., 2018, Haeffele and Vidal, 2015]. In a more general setting, Hardt et al. [2016] prove algorithmic stability-based generalization bounds that worsen with increasing training time, whereas Loukas et al. [2021] connect stable training dynamics near convergence under Dropout with good generalization. For algorithms that do not converge (close) to a fixed point, the above analyses result in vacuous bounds.

Contributions. We start with the supposition that robustness of the learning algorithm's fixed points to small changes in the training set is not the mechanism underlying the good generalization properties of deep networks. This is because, as is commonly known and many recent works indirectly remark upon [Cohen et al., 2021, Ahn et al., 2022, Garipov et al., 2018, Zhang et al., 2022], training can be linearly unstable: even if assuming the presence of isolated local minima that the learning algorithm converges to, the weights encountered in training are not robust to small perturbations. Small perturbations can accumulate over time leading to completely different orbits in weight space. Yet, despite unstable dynamics, the generalization properties of two completely different fixed points (e.g., obtained from two different random initializations or due to a different order of SGD updates) are typically comparable.

Our first contribution entails proposing a generalization of stability that applies to non-converging algorithms. To achieve this, we depart from the typical optimization perspective of analyzing the generalization properties of minima [Keskar et al., 2016, Ge et al., 2015, Sagun et al., 2016]. Instead, we study the generalization performance of the learning algorithm on average (see section 2.2). Adopting a statistical viewpoint, in Section 3 we define a new notion of statistical algorithmic stability that measures the stability of this average performance to perturbations of the training data, as opposed to stability of individual orbits in the weight space. Section 3 then proves upper bounds for the average generalization error based on our new notion of statistical stability.

We then move on to consider how statistical stability connects to the behavior of the loss function. To this end, in Section 4 we provide conditions such that the spectral gap of a Markov operator associated with the dynamics of the loss function is predictive of statistical algorithmic stability and hence of generalization. Empirically, we estimate this spectral gap based on the rate of convergence of ergodic averages of the loss function. We show via experiments that our estimate correlates with generalization for image classification under corruption.

1.1 Related work

Our work builds on previous analyses of the stability of learning algorithms that converge [Bousquet and Elisseeff, 2002, Feldman and Vondrak, 2018, Kuzborskij and Lampert, 2018, Bousquet et al., 2020, Zhang et al., 2021b]. Below, we also briefly discuss connections (beyond the literature on stability) with previous analyses on the dynamics of learning algorithms.

Convergence and generalization of SGD. Exponential convergence rates to global minima are known for SGD in the smooth and convex setting, with decaying learning rates and small constant learning rates (e.g., [Moulines and Bach, 2011, Needell et al., 2014]). In the training of deep neural networks, which is the focus of this paper, the loss function is non-convex and is typically not well-approximated locally by convex functions [Ma et al., 2018]. However, in the overparameterized regime, a Polyak-Lojasiewicz-type inequality that is automatically satisfied when the loss is smooth and convex can be shown to hold in the non-convex setting as well [Liu et al., 2022, Ma et al., 2018], allowing convergence to a global minimum. In certain non-convex problems, convergence to local minima and even global minima has been proved, assuming a strict saddle property [Ge et al., 2015]

or local convexity [Kleinberg et al., 2018]. Raginsky et al. [2017] show a favorable convergence for a slightly different version of SGD by approximating the dynamics with a suitable continuous time Langevin diffusion. A number of previous works [Safran and Shamir, 2016, Freeman and Bruna, 2016, Li and Yuan, 2017, Nguyen et al., 2018] have also provided arguments that SGD converges to a solution that generalizes given suitable initialization and significant overparameterization. A few works have studied the nonlinear dynamics of training [Kong and Tao, 2020] and provided insights into generalization in the overparameterized regime [Dogra and Redman, 2020, Saxe et al., 2013, Advani et al., 2020]. Another line of work considers the mean field limit of SGD dynamics to show generalization [Mei et al., 2019, Gabrié et al., 2018, Chen et al., 2020]. The ergodic theoretic perspective we adopt here deviates from the optimization perspectives but relates to the mean field perspective in that we implicitly consider (see section 4) the evolution of probability distributions on weight space.

Flat minima. In the generalization literature, flat minima correspond to large connected regions with low empirical error in weight space. Flat minima have been argued to be related to the complexity of the resulting hypothesis and, hence, can imply good generalization Hochreiter and Schmidhuber [1997]. It has also been shown that SGD converges more frequently to flat minima when the batch size is small or the learning rate and the number of iterations are suitably adjusted [Keskar et al., 2016, Hoffer et al., 2017, Jastrzębski et al., 2017, Smith and Le, 2018, Zhang et al., 2018]. Some works consider the flat/sharp dichotomy an oversimplification [Dinh et al., 2017, Sagun et al., 2017, He et al., 2019], e.g., using a reparameterization argument [Sagun et al., 2017]. The eigenvalues of the hessian of the loss affect the local linear stability of optimization orbits. However, given that we adopt the time-asymptotic/statistical picture, we do not explicitly make assumptions on local stability (flatness/sharpness) or the topology of the loss landscape [Montanari and Zhong, 2020, Venturi et al., 2019].

SGD as a Bayesian sampler. The idea of looking at distributions of learners permeates PAC-Bayesian analyses of generalization [McAllester, 1999, Dziugaite and Roy, 2017, Zhou et al., 2019, Pitas, 2020]. In contrast to Bayesian posteriors of the parameters, we study parameter distributions generated by the learning algorithms. Previous empirical work has suggested that SGD operates almost like a Bayesian sampler [Mingard et al., 2021] but the connection remains to be fully understood.

2 Local descent algorithms as dynamical systems: a statistical viewpoint

This section lays out the basic definitions and assumptions of our work. We begin in Section 2.1 by introducing the learning problem and discussing learning algorithms from a dynamical systems perspective. Section 2.2 then puts forth the notions of statistical convergence that give rise to our main results.

2.1 Learning as a dynamical system

We consider the supervised learning setup: a learner is given a training set $S = \{z_1, \dots, z_n\}$ consisting of n pairs $z_i \equiv (x_i, y_i)$ of inputs $x_i \in \mathbb{R}^d$ and their corresponding labels $y_i \in Y \subseteq \mathbb{R}$, drawn i.i.d. from a distribution D. A class of parametric models or hypotheses $h : \mathbb{R}^d \times W \to \mathbb{R}$ on a parameter space W gives possible input-to-output relationships, $h(w, \cdot) : \mathbb{R}^d \to \mathbb{R}$. The learner is given a risk or loss function $\ell : (\mathbb{R}^d \times \mathbb{R}) \times W \to \mathbb{R}^+$ that describes the error of a given hypothesis. Common choices for loss functions are the mean-squared, hinge and cross-entropy loss. The learner attempts to minimize the population risk $R_S(h(\cdot, w)) = E_{z \in D}\, \ell(z, w)$, by minimizing the empirical risk

$$\hat{R}_S(h(\cdot, w)) = L_S(w) = \frac{1}{n} \sum_{z_i \in S} \ell(z_i, w). \tag{1}$$

This minimization is achieved using a local descent algorithm given by iterative updates $\varphi_S : W \to W$ of the form

$$w_{t+1} = \varphi_S(w_t) := w_t - \eta_t \hat{\nabla} L_S(w_t), \tag{2}$$

where η_t is the learning rate or step size at time t.

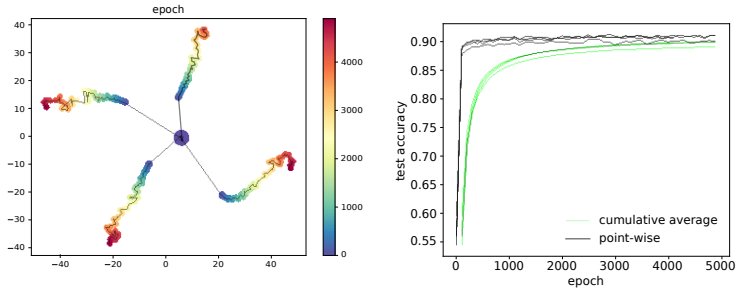

Figure 1: Although the training of neural networks depends on the initialization and does not necessarily converge in the weight space, certain functionals of the learned hypothesis can converge in distribution independently of initial conditions. Left: The orbits of the 2nd layer weights of four different VGG16 models trained on a CIFAR10 dataset using SGD with step size 0.01. We embed the orbits in the plane by performing PCA onto 40 dimensions followed by t-SNE. Colors indicate epoch. Right: Corresponding test accuracy and cumulative average values as a function of epoch.

In the gradient descent (GD) algorithm, $\nabla\hat{L}_S(w) = \nabla L_S(w)$ is the gradient w.r.t. w, and the iterates of φ_S represent a deterministic dynamical system. The critical points (including saddle points, local and global minima) of L_S are fixed points of φ_S (i.e., points w such that $\varphi_S w = w$). For stochastic gradient descent (SGD), $\nabla L_S(w)$ is a random variable given by the sample mean of gradients $\nabla\ell(\cdot, w)$ over a batch of samples in S. In this case, φ_S is a random dynamical system and critical points of L_S are not necessarily fixed points of φ_S.

To unify the definition of φ_S for both GD and SGD, we introduce the random variable Ξ that indicates the choice of batch. Suppose the batch size m is fixed: $m < n$ for SGD and $m = n$ for GD. Denote by [n] the set $\{1, 2, \cdots, n\}$ and let Ξ be a collection of m elements from [n] chosen uniformly at random. Then, we may write φ_S as

$$w_{t+1} = \varphi_S(w_t) = w_t - \eta_t \nabla\hat{L}_S(w_t \mid \Xi_t) = w_t - \frac{\eta_t}{m} \sum_{i \in \Xi} \nabla\ell(z_i, w_t), \tag{3}$$

with Ξ_t being the set [n] at all t for GD.

In our investigation, we opt for simplicity and consider a fixed learning rate $\eta_t = \eta$. More generally, our analysis can be extended to learning rates that asymptotically converge to η. A rapidly vanish-ing learning rate can obscure the learning dynamics by enforcing convergence to arbitrary points irrespective of the points' losses. In contrast, learning rates that do not decay to zero can induce interesting transitions between neighborhoods of critical points and the iterates w_t are generally more exploratory of the loss landscape.

2.2 Statistical convergence of the learning algorithm

In deep learning, any specific choice of w_T, at some time T, is arbitrary as the loss landscape is generally non-convex and stochastic learning algorithms are the norm. Hence, rather than focusing on a single hypothesis, it is attractive to consider the average generalization error induced by the stochastic dynamical system φ_S. We adopt this statistical perspective, rather than focus on a single orbit of φ_S. Thus, the setting below applies to any asymptotic behavior of orbits of the (typically nonlinear) dynamics φ_S including fixed points, periodic, quasiperiodic and chaotic orbits.

To analyze the generalization performance without making assumptions on the global stability properties of individual orbits, we make an assumption about the ergodic properties of φ_S that reasonably matches empirical evidence. To describe the assumption, we give a short primer on the relevant concepts from ergodic theory of dynamical systems.

Invariant measures. Denote the Borel sigma algebra on W by $\mathcal{B}(W)$. A probability measure μ_S is called invariant for φ_S if $\mu_S(A) = \mu_S \circ \varphi_S^{-1}(A)$ for all Borel sets $A \in \mathcal{B}(W)$. Intuitively, for any subset of the weight space, the probability that the dynamical system occupies it does not change. See Liverani [2004] for an introduction to invariant measures.

If φ_S is a deterministic continuous function on the set W and W is compact, classical ergodic theorems (see e.g., Theorem 4.1.1 of Katok and Hasselblatt [1997]) give the existence of at least one invariant, ergodic measure for φ_S. Ergodic measures are a pertinent class of invariant measures in practice, that allow us to understand the statistical properties of a system through evolution of individual orbits. For an ergodic measure, sets that are invariant under the dynamics φ_S (such as a set of periodic points, quasiperiodic or chaotic attractors) are trivial: they have measure 0 or 1.

When φ_S is an SGD update, Dieuleveut et al. [2020] show the existence of invariant distributions for convex loss functions, whereas Fort and Pagès [1999] and Mattingly et al. [2002] show the existence of invariant distributions in more general settings under mild assumptions on the gradient noise (in the estimate $\nabla \hat{L}_S$). Several works have analyzed the asymptotic properties of such invariant distributions, e.g., Chee and Toulis [2018] analyze the oscillation of the iterates w_t about a mean that is a critical point, as a phase separate from the transient phase of convergence to this invariant distribution. Kong and Tao [2020] study the effect of large learning rates (see also [Wang et al., 2021] for matrix factorization problems) and multiscale loss functions to show the existence of Gibbs invariant measures in GD.

Ergodicity. When an invariant measure μ_S is ergodic for φ_S, time averages converge to ensemble averages according to μ_S. That is, for μ_S-almost every initial state w_0, and for all continuous scalar functions h:

$$\frac{1}{T}\sum_{t=0}^{T-1} h(w_t) \to E_{w\sim\mu_S} h(w) \tag{4}$$

and the dependence on initial conditions w_0 is forgotten.

Unfortunately, we cannot generally expect the dynamics of the learning algorithm to have a unique ergodic invariant measure on W. Instead, φ_S-invariant sets could be smaller subsets of W with different ergodic invariant measures supported on them. As a result, starting from two different points chosen Lebesgue almost everywhere (or sampled from any probability density on W), we do not expect time averages to converge to the same limit. Indeed, as shown in Figure 1, the weights visited by φ_S depend on the initial conditions and can vary significantly across different runs. In other words, the limit of infinite time averages described in (4) does depend on w_0.

Dynamics of the hypothesis. To circumvent the above issues, rather than focusing on the weights, we consider the dynamics of the learned hypothesis $h(\cdot, w_t)$. Specifically, suppose that the infinite-time averages of the learned hypothesis are ergodic: that is, the time averages of $h(z, w_t)$ converge to the same function irrespective of the initial function $h(z, w_0)$. Then, functionals that depend only on h and not explicitly on the weights are also ergodic. This is one such scenario that leads to our main assumption: the time averages of loss functions are ergodic.

Assumption 1. Given any $S \in D^n$, there exists a map $z \to \langle \ell_z \rangle_S$, such that for Lebesgue-a.e. w_0 and every $z \in R^d \times R$, the following holds:

$$\lim_{T\to\infty} \frac{1}{T}\sum_{t=0}^{T-1} \ell(z, w_t) = \langle \ell_z \rangle_S \in R^+, \tag{5}$$

where $\{w_t = \varphi_S(w_{t-1})\}$ is an orbit of φ_S.

As an intuitive justification for Assumption 1, symmetries may result in several sets of parameters (weights and biases) that represent the same network function h. Hence, assuming that there exists a unique probability on the space of neural network functions that is ergodic is less restrictive and closer to reality than assuming a unique ergodic measure on the parameter space. Furthermore, ergodicity of the network functions is a stronger condition than our Assumption 1 and serves to illustrate one possible sufficient condition for our assumption to hold.

Figure 1 provides evidence that Assumption 1 can hold in practice. It shows two observables calculated along 4 different orbits of a VGG16 model trained using SGD with momentum (batches of size 128, learning rate of 0.01, momentum of 0.9). On the left, we can see a low-dimensional projection of the second layer weights obtained by projecting the weights onto 40 dimensions using PCA and then employing t-SNE [Van der Maaten and Hinton, 2008], whereas the observable on the right is the test accuracy, which is a functional of the hypothesis function. Here, the test accuracy can

be seen to converge to a distribution independent of the initial conditions, whilst the weight orbits vary significantly across different initializations. This experiment corroborates our hypothesis that time averages of functionals on the loss space converge to distributions independent of the initial condition, even if time averages of a generic observable on the weight space do not. See Remark 1 on the idea that a larger learning rate induces ergodicity and Appendix A, which illustrates the idea with a toy example.

3 Statistical stability implies generalization

In Section 3.1, we generalize algorithmic stability beyond algorithms that converge to fixed points. Section 3.2 then proceeds to examine the implications of our definition to generalization error.

3.1 Statistical algorithmic stability

Classically, the derivations of algorithmic stability-based generalization (see e.g., Hardt et al. [2016], Chapter 14 of Mohri et al. [2018], Bousquet et al. [2020]) utilize an input perturbation of one element in S by replacing it with a different element from the input distribution D:

Definition 1 (Algorithmic stability, adapted from [Bousquet and Elisseeff, 2002]). Consider the weights w_S and $w_{S'}$ obtained by running the learning algorithm on two training sets S, S' sampled from D that differ by exactly one sample. We say that the learning algorithm is algorithmically stable (AS) with a stability coefficient $\beta \geq 0$ if

$$\beta = \sup \{ |\ell(z, w_S) - \ell(z, w_{S'})| : z \in \mathbb{R}^d \times \mathbb{R} \}. \tag{6}$$

We refer to this type of perturbation as a stochastic perturbation. Algorithmic stability does not directly apply to learning algorithms that do not converge to a fixed point. Hence, next, we extend algorithmic stability to loss statistics. The resulting notion of statistical algorithmic stability (SAS) extends algorithmic stability to algorithms whose loss statistics converge as prescribed by Assumption 1, even if the weights do not converge.

Definition 2 (Statistical algorithmic stability). We say that a learning algorithm with loss statistics denoted by $\langle \ell_z \rangle_S$ is statistically algorithmically stable (SAS) with a stability coefficient $\beta \geq 0$ if

$$\beta = \sup \{ |\langle \ell_z \rangle_S - \langle \ell_z \rangle_{S'}| : z \in \mathbb{R}^d \times \mathbb{R} \}, \tag{7}$$

where S, S' differ in exactly one element.

In the above definition, $\langle \ell_z \rangle_{S'}$ refers to the ergodic average of $\ell(z, \cdot)$ observed along almost every orbit of $\varphi_{S'}$. A higher value of β indicates a lower SAS algorithm. The definition of SAS differs from Definition 1 as we do not assume convergence of the learning algorithm. Nevertheless, when every orbit of φ_S converges to a fixed point w_S, for almost every S, then Definition (7) reduces to the standard notion of algorithmic stability since ergodic averages along almost every orbit converge to $\ell(z, w_S)$.

Crucially, and in line with the observations in Figure 1, we quantify stability based on the statistics of the loss function and not the loss function at an ensemble mean or at any one point in the weight space W. In other words, the definition of SAS does not use or provide information about the algorithmic stability of $E_{w \in \mu_S} w$ for any invariant measure μ_S.

3.2 Learning theoretic implications

It is well-known that Definition 1 leads to generalization bounds (see [Hardt et al., 2016, Bousquet et al., 2020] and references therein). Next, we show that the more general statistical algorithmic stability from Definition 2 also can imply a notion of generalization.

To obtain generalization bounds, we first redefine the empirical and population risk to use the loss statistics rather than loss values at fixed points:

$$\hat{R}_S = \frac{1}{n} \sum_{z \in S} \langle \ell_z \rangle_S \quad \text{and} \quad R_S = E_{z \in D} \langle \ell_z \rangle_S. \tag{8}$$

Although the definitions above appear deceivingly similar to the standard ones introduced in Section 2.1, the two notions of risk are defined on different spaces, with the standard ones being maps from the hypothesis space H to scalar values and (8) being maps from the space of learning algorithms. That is, the latter risks only depend on algorithmic parameters and are not functions of H. Using these definitions, the following generalization bound holds:

Theorem 1. Given a β-statistically algorithmically stable algorithm φ_S (see Definition 2), for any $\delta \in (0, 1)$, with probability greater than $1 - \delta$,

$$R_S \leq \hat{R}_S + \beta + 2(n\beta + L)\sqrt{\frac{\log(2/\delta)}{2n}}, \tag{9}$$

where $L := \sup_z \sup_{w \in W} |\ell(z, w)|$ is an upper bound on the loss.

The proof can be found in Appendix B and is based on applying Mcdiarmid's inequality to the generalization gap. Like Bousquet and Elisseeff [2002]'s bound, the SAS coefficient β has to decay at least as fast as $O(1/\sqrt{n})$ for the generalization bound to be non-vacuous. Note that our proof is also naturally coupled with [Bousquet et al., 2020, Feldman and Vondrak, 2018]'s improved technique, which provides tighter bounds when $\beta \in O(1/n)$. (see Appendix B).

4 Dynamical systems interpretation of SAS

How can we distinguish between two algorithms with different SAS? Can the algorithm dynamics provide clues to its SAS? These questions are of practical importance because a more (statistically) stable algorithm generalizes better (via Theorem 1). Since the SAS coefficient β is defined as a supremum over infinite pairs of training sets and inputs, a numerical estimation of it using its definition only gives a rough lower bound and obtaining this lower bound is also computationally expensive (see Figure 3 and section 5). Moreover, a numerical lower bound on β does not give us a mechanistic understanding of statistical stability, which we seek here.

Here we develop an operator-theoretic explanation for what makes an algorithm statistically stable. We also derive a rough heuristic – albeit not a definitive predictor of generalization – that relates a given training dynamics to its SAS coefficient β.

4.1 Bounding the SAS coefficient

Before we present our bound, we discuss why we must deviate from the trajectory-based approach that is commonly employed to study algorithmic stability. We recall that Hardt et al. [2016] base their proofs of classical algorithmic stability (Theorems 3.7 and 3.8 in Hardt et al. [2016]) on the accumulating differences between two orbits corresponding to S and S', whenever the different element is chosen in the mini-batch. The greater the difference in the weights, the greater the difference in the (upper bound on the) loss functions at those weights, in the case of Lipschitz loss. Thus, Hardt et al. [2016] argue that training faster (or stopping earlier) will lead to stabler algorithms. Since SAS is about robustness of loss statistics or infinitely long time averages, the analysis à la Hardt et al. [2016] generically leads to vacuous bounds for the SAS coefficient.

Our analysis is based on the realization that an algorithm can be SAS even if two orbits corresponding to S and S' diverge from each other (within W), but lead to similar loss statistics. Thus, to bound the SAS coefficient, a perturbation analysis of transition operators (global information), rather than a finite-time perturbation analysis of an orbit (local information), is needed. Since SAS only demands robustness on loss space, we consider Markov transition operators on the loss space ($\in [0, L]$), as opposed to on the weight space (W). In general, at any z and w_0, the loss process, $\{\ell(z, w_t)\}_t$ is not Markovian. But, a family of Markov operators can be associated with the family of loss functions.

Lemma 1. (Markov operators) Let μ_S be an ergodic, invariant measure for φ_S. Assume that a loss function $\ell(z, \cdot) : W \to I_z \in [0, L]$ is such that the pushforward, $\nu_S^z : \mathcal{B}(I_z) \to \mathbb{R}^+$, of μ_S on the loss space is well-defined. Here, $\mathcal{B}(I_z)$ is the Borel sigma algebra on I_z and $\nu_S^z = \ell(z, \cdot)_\# \mu_S$, where $\#$ refers to the pushforward operation. Then, there exists a uniformly ergodic Markov operator $P^z_{\mu_S}$ on the space of probability measures on I_z, with its invariant measure being ν_S^z.

This lemma (see Appendix C for a proof) parallels Chekroun et al. [2014]'s Theorem A, but in a quite different setting, without using existence of a unique physical measure on the phase space (W).

Combined with Assumption 1, perturbation results on the Markov operators, $P_{\mu_S^z}$, defined by Lemma 1 lead us to SAS. Since $P_{\mu_S^z}$ is uniformly ergodic, there exist $\lambda_z \in (0, 1)$ and $C > 0$ such that for all $\xi \in I_z$, in the Wasserstein norm ($\|\cdot\|_W$),

$$\|P_{\mu_S^z}^t \delta_\xi - \nu_S^z\|_W \leq C\lambda_z^t. \tag{10}$$

Let $\lambda := \sup_z \lambda_z$ be the rate of mixing associated with the family of operators $P_{\mu_S^z}$. We leverage the perturbation theory of Markov operators to study the effect of stochastic perturbations on ν_S^z, and subsequently, SAS. To see the connection with SAS, we recall that, by Assumption 1, for any choice of μ_S^z, we have that $\langle \ell \rangle_z = E_{\xi \sim \nu_S^z} \xi$. Thus, given any pair of μ_S and $\mu_{S'}$, we have (see Appendix C for details)

$$|\langle \ell_z \rangle_S - \langle \ell_z \rangle_{S'}| = |E_{\xi \sim \nu_S^z} \xi - E_{\xi \sim \nu_{S'}^z} \xi| \leq \|\nu_S^z - \nu_{S'}^z\|_W. \tag{11}$$

Thus, a uniform upper bound on $\|\nu_S^z - \nu_{S'}^z\|_W$ gives an upper bound for the SAS coefficient, β. Such a bound is obtained from the straightforward use of an appropriate perturbation result on uniformly ergodic Markov chains.

Theorem 2. Given a uniformly ergodic Markov operator $P_{\mu_S^z}$ constructed in Lemma 1, and that the Lipschitz constant of $\|\ell(\cdot, w)\|$ on R^d is bounded above by L_D for all $w \in W$, φ_S is SAS with stability coefficient

$$\beta = O\left(\frac{1}{n}\frac{L_D}{1-\lambda}\right),$$

where $\lambda = \sup_z \lambda_z$ is the supremum over z of the mixing rates of $P_{\mu_S^z}$.

Proof. (Sketch) The proof relies on the perturbation bounds of Rudolf and Schweizer [2018], Corollary 3.2. Note that when (10) is satisfied, and if a stochastic perturbation S' introduces a change $\delta P_{\mu_S^z}$ to $P_{\mu_{S'}^z}$, the perturbation bound from [Rudolf and Schweizer, 2018] gives

$$\|\nu_S^z - \nu_{S'}^z\|_W \leq C\|\delta P_{\mu_S^z}\|/(1 - \lambda_z), \tag{12}$$

where $\|\delta P_{\mu_S^z}\|$ is the operator norm of $\delta P_{\mu_S^z}$ induced by Wasserstein norm. From (11),

$$|\langle \ell_z \rangle_S - \langle \ell_z \rangle_{S'}| \leq C\|\delta P_{\mu_S^z}\|/(1 - \lambda_z). \tag{13}$$

Since this holds for all z, the right hand side of (13) is an upper bound for the stability coefficient β in Definition (2), by taking the supremum over z. An upper bound on $\delta P_{\mu_S^z}$ can be derived when the Lipschitz constant of $\|\ell(\cdot, w)\|$ is uniformly (in w) bounded on R^d, as shown in Appendix C). □

Theorem 2 implies that an algorithm that exhibits faster convergence to the stationary measure on the loss space generalizes better. The bound in Theorem 2 implies that smaller λ (faster convergence of the loss to the ergodic invariant measure) yields smaller upper bounds on β (more statistically stable algorithm) and better generalization.

Strictly speaking, to determine λ_z, we must consider finite-dimensional approximations of the operator $P_{\mu_S^z}$ (as done in the setting of chaotic systems in e.g., [Crimmins and Froyland, 2020, Chekroun et al., 2014]) and compute its spectral decomposition. However, in our setting, it is difficult in practice to generate samples in the loss space that obey the Markov process defined in Lemma 1. Instead, we argue for using the readily available non-Markovian loss process to estimate convergence to the equilibrium distribution. In particular, we hypothesize below that the auto-correlation function of the test loss can separate algorithms based on their SAS.

Dropping the superscripts z, we define the auto-correlation $C_\ell(\tau)$ in a loss function ℓ (see also Remark 3 in Appendix C) by

$$C_\ell(\tau) := \left|\lim_{T\to\infty} \frac{1}{T}\sum_{t\leq T} \ell(w_t)\ell(w_{t+\tau}) - \langle \ell \rangle^2\right|/\langle \ell \rangle^2. \tag{14}$$

That is, $C_\ell(\tau)$ gives the correlation between the random variables $\ell(w_t)/\langle \ell \rangle$ and $\ell(w_{t+\tau})/\langle \ell \rangle$, where the randomness comes from the batch selection Ξ_t and the initialization of weights in SGD, and only from the latter in GD.

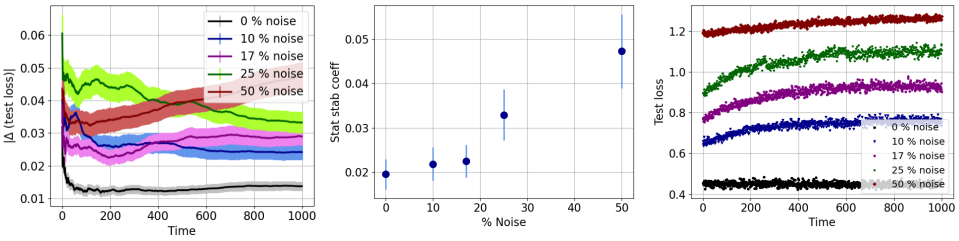

Figure 2: SAS computed from the VGG16 model. Left: The change in cumulative time averages of the test loss upon stochastic perturbation of the CIFAR10 dataset. Center: Lower bound on the statistical stability coefficient β (Definition 2) computed using the test loss. Right: Test error timeseries.

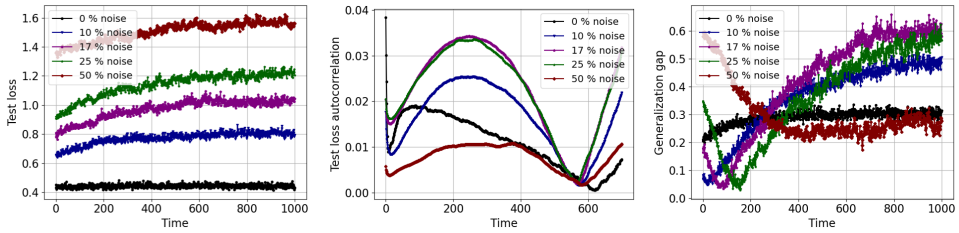

Figure 3: Predictors of generalization gaps on data with corrupted labels computed on the ResNet18 model. Left: Time series of the test loss. Center: Normalized auto-correlation of the test loss timeseries on the left. The magnitude of the auto-correlation in the test loss is suggestive of the generalization gap (right). For large percentage of noise, the gap decreases because the neural network cannot fit the training data well.

Suppose that the loss process is Markov, i.e., $\ell(w_{t+1})$ conditioned on $\ell(w_t)$ is independent of $\{\ell(w_{t'}) : t' \leq t-1\}$. Setting $t = 0$ after a sufficiently long runup time, the function $C_\ell(\tau)$ in (14) is the auto-correlation function of a stationary Markov chain, which typically decays exponentially with τ. This decay rate is lower (slower correlation decay/longer correlation times) when λ is closer to 1. Qualitatively, comparing two different stationary Markov chains, the one with higher values of C_ℓ has longer correlation times and hence, larger λ.

In general, however, the loss process $\{\ell(w_t) : t \geq 0\}$ is non-Markovian. Thus, we do not expect $C_\ell(\tau)$ to decay with τ. This is because the correlation function C_ℓ encodes both the Markovian and non-Markovian components component of the loss dynamics (see Remark 4 in Appendix C; [Zwanzig, 2001, Kondrashov et al., 2015]). Hence, we expect the correlation times in the loss process to not be equal to that of a Markov process generated according to P_{μ_S}. For the purpose of qualitatively comparing the SAS coefficients of two algorithms, however, the loss process can be considered a heuristic for a Markov process generated according to P_{μ_S}. Then, higher values of C_ℓ computed from the loss process indicate larger λ, and hence less statistical stability.

5 Numerical results

We numerically validate the main ideas of section 3 and 4 on VGG16 and ResNet18 models trained on the CIFAR10 dataset (see Appendix D for further numerical results, [Chandramoorthy and Loukas, 2023] for the code). For all our experiments, φ_S is an SGD update with momentum 0.9, fixed learning rate 0.01 and batch size of 128. In all figures, "time" indicates number of epochs. We generate different versions of the training set S_p by corrupting CIFAR10's labels with probability p, with S_0 being the original CIFAR10 dataset. Figures 2 and 3 show results corresponding to p = 0, 0.1, 0.17, 0.25 and 0.5. Each line in Figure 3 is a sample mean over 10 random initializations.

In Figure 2, we numerically estimate a lower bound on the SAS coefficient β using its definition (see Definition 2). On the left, we plot the difference in test loss time-averages between between random orbits of φ_{S_p} and $\varphi_{S'_p}$, with S'_p being a stochastic perturbation of S_p. The mean over 45 pairs of orbits is shown as dots and the error bars indicate the standard error in mean. The cumulative time

average along orbits of length 1200 epochs are used as estimators for statistics. With these estimators for statistics, in Figure 2(center), we compute an estimate of the stability coefficient as the difference of the (estimated) test loss statistics at different values of p. This is hence an estimate on the lower bound of β, and clearly increases with p. This illustrates that cases with worse generalization errors (Figure 2 (right)) have larger lower bounds on β.

In Figure 3 (left), we show the test loss timeseries obtained with the ResNet18 model. Again, greater the noise corruption p, the larger is the generalization error (estimated by test error), consistent with previous studies Zhang et al. [2021a], Loukas et al. [2021]). On the other hand, the generalization gap – difference between training and test error – is bigger for intermediate levels of noise and decreases when $p = 0.5$, as shown in Figure 3 (center). The gap does not always increase with p because, when the labels are close to random, the network cannot fit the training data and thus the training error is also large.

In Figure 3 (right), we plot $C_{\ell}(\tau)$ (Section 4) where ℓ is taken as the test loss. The test loss statistic $\langle \ell \rangle$ is estimated as a time average over 1200 epochs. At each p, we show the sample average of the auto-correlations over 10 independent runs. We see that the magnitude of the test loss auto-correlations preserves the same order (across $p's$) as the generalization gap (absolute difference in test and training losses) shown in Figure 3(center). Hence, we empirically observe that the loss process codifies the phenomenological explanation for SAS that is expressed in Theorem 2.

6 Discussion and conclusion

Predicting generalization is an active area of research and has implications for more reliable use of machine learning. In this work, we introduce statistical algorithmic stability (SAS) and show how it implies generalization bounds. Here, we add a few important remarks.

Statistical stability only requires robustness of statistics on loss space. A central challenge in the analyses of non-convergent algorithms is the fact that there may be multiple, very different invariant measures μ_S. For this reason, our stability criterion focuses on statistics of the loss function. This way, an algorithm can be stable even if the measures μ_S and $\mu_{S'}$ are not close in the total variation norm. In fact, an algorithm can be stable even if μ_S and $\mu_{S'}$ are mutually singular, if the loss functions have similar statistics. Since our SAS analysis hinges on a reasonable yet nonrestrictive assumption on ergodic properties of the algorithm, we believe it is broadly applicable.

Algorithms that converge are special cases of the above analysis. In Appendix E, we show that the proposed relationship (section 4) between statistical stability and convergence rates to stationary measures can also be observed in the Neural Tangent Kernel (NTK) regime [Jacot et al., 2018]. In this case, the training dynamics, φ_S, can be approximated by a linear function of the weights, which converges to a fixed point. Thus, this provides an alternative, ergodic theoretic, interpretation of generalization in the NTK regime [Arora et al., 2019b, Montanari and Zhong, 2020, Bartlett et al., 2021].

Broader view. While our focus is on algorithmic stability-based generalization, this work gathers more evidence to support the broader view [Wojtowytsch, 2021, Zhang et al., 2022] that exploiting theoretically and empirically available dynamical information about the training algorithm is a fruitful complement to understanding generalization from the optimization landscape and learning theory perspectives.

Funding: This work was partially funded by the NSF AI Institute TILOS, NSF award 2134108, and ONR grant N00014-20-1-2023 (MURI ML-SCOPE).

Acknolwedgments: N.C. would like to thank Nandhini Chandramoorthy and Derek Lim for helping with the experimental setup and Sven Wang, Benjamin Zhang, Matt Li and Youssef Marzouk for valuable discussions. The authors also thank the reviewers for their constructive suggestions.

References

M. S. Advani, A. M. Saxe, and H. Sompolinsky. High-dimensional dynamics of generalization error in neural networks. Neural Networks, 132:428–446, 2020. ISSN 0893-6080. doi: https://doi.org/10.1016/j.neunet.2020.08.022. URL https://www.sciencedirect.com/science/article/pii/S0893608020303117.

K. Ahn, J. Zhang, and S. Sra. Understanding the unstable convergence of gradient descent, 2022. URL https://arxiv.org/abs/2204.01050.

Z. Allen-Zhu, Y. Li, and Y. Liang. Learning and generalization in overparameterized neural networks, going beyond two layers. 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/62dad6e273d32235ae02b7d321578ee8-Paper.pdf.

H. Arbabi and I. Mezic. Ergodic theory, dynamic mode decomposition, and computation of spectral properties of the koopman operator. SIAM Journal on Applied Dynamical Systems, 16(4):2096–2126, 2017.

J. L. Aron and I. B. Schwartz. Seasonality and period-doubling bifurcations in an epidemic model. Journal of theoretical biology, 110(4):665–679, 1984.

S. Arora, R. Ge, B. Neyshabur, and Y. Zhang. Stronger generalization bounds for deep nets via a compression approach. In International Conference on Machine Learning, pages 254–263. PMLR, 2018.

S. Arora, S. Du, W. Hu, Z. Li, and R. Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In International Conference on Machine Learning, pages 322–332. PMLR, 2019a.

S. Arora, S. S. Du, W. Hu, Z. Li, R. R. Salakhutdinov, and R. Wang. On exact computation with an infinitely wide neural net. 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., 2019b. URL https://proceedings.neurips.cc/paper/2019/file/dbc4d84bfcfe2284ba11beffb853a8c4-Paper.pdf.

P. L. Bartlett, D. J. Foster, and M. Telgarsky. Spectrally-normalized margin bounds for neural networks. In Proceedings of the 31st International Conference on Neural Information Processing Systems, pages 6241–6250, 2017.

P. L. Bartlett, N. Harvey, C. Liaw, and A. Mehrabian. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. Journal of Machine Learning Research, 20(63):1–17, 2019. URL http://jmlr.org/papers/v20/17-612.html.

P. L. Bartlett, A. Montanari, and A. Rakhlin. Deep learning: a statistical viewpoint. Acta numerica, 30:87–201, 2021.

O. Bousquet and A. Elisseeff. Stability and generalization. The Journal of Machine Learning Research, 2:499–526, 2002.

O. Bousquet, Y. Klochkov, and N. Zhivotovskiy. Sharper bounds for uniformly stable algorithms. In J. Abernethy and S. Agarwal, editors, Proceedings of Thirty Third Conference on Learning Theory, volume 125 of Proceedings of Machine Learning Research, pages 610–626. PMLR, 09–12 Jul 2020. URL https://proceedings.mlr.press/v125/bousquet20b.html.

M. Budišić, R. Mohr, and I. Mezić. Applied koopmanism. Chaos: An Interdisciplinary Journal of Nonlinear Science, 22(4):047510, 2012. doi: 10.1063/1.4772195.

N. Chandramoorthy and A. Loukas. ni-sha-c/astability: Paper, Jan. 2023. URL https://doi.org/10.5281/zenodo.7525770.

J. Chee and P. Toulis. Convergence diagnostics for stochastic gradient descent with constant learning rate. In A. Storkey and F. Perez-Cruz, editors, Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics, volume 84 of Proceedings of Machine Learning Research, pages 1476–1485. PMLR, 09–11 Apr 2018.

M. D. Chekroun, J. D. Neelin, D. Kondrashov, J. C. McWilliams, and M. Ghil. Rough parameter dependence in climate models and the role of ruelle-pollicott resonances. Proceedings of the National Academy of Sciences, 111(5):1684–1690, 2014.

Z. Chen, Y. Cao, Q. Gu, and T. Zhang. Mean-field analysis of two-layer neural networks: Non-asymptotic rates and generalization bounds. arXiv preprint arXiv:2002.04026, 2020.

J. M. Cohen, S. Kaur, Y. Li, J. Z. Kolter, and A. Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability, 2021. URL https://arxiv.org/abs/2103.00065.

H. Crimmins and G. Froyland. Fourier approximation of the statistical properties of anosov maps on tori. Nonlinearity, 33(11):6244, 2020.

M. Dellnitz, G. Froyland, and S. Sertl. On the isolated spectrum of the perron-frobenius operator. Nonlinearity, 13(4):1171, 2000.

A. Dieuleveut, A. Durmus, and F. Bach. Bridging the gap between constant step size stochastic gradient descent and Markov chains. The Annals of Statistics, 48(3):1348 – 1382, 2020. doi: 10.1214/19-AOS1850. URL https://doi.org/10.1214/19-AOS1850.

L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio. Sharp minima can generalize for deep nets. In International Conference on Machine Learning, pages 1019–1028. PMLR, 2017.

A. S. Dogra and W. Redman. Optimizing neural networks via koopman operator theory. Advances in Neural Information Processing Systems, 33:2087–2097, 2020.

G. K. Dziugaite and D. M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. arXiv preprint arXiv:1703.11008, 2017.

V. Feldman and J. Vondrak. Generalization bounds for uniformly stable algorithms. 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/05a624166c8eb8273b8464e8d9cb5bd9-Paper.pdf.

J.-C. Fort and G. Pagès. Asymptotic behavior of a markovian stochastic algorithm with constant step. SIAM Journal on Control and Optimization, 37(5):1456–1482, 1999.

C. D. Freeman and J. Bruna. Topology and geometry of half-rectified network optimization. arXiv preprint arXiv:1611.01540, 2016.

S. Frei, Y. Cao, and Q. Gu. Algorithm-dependent generalization bounds for overparameterized deep residual networks. 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/6e2290dbf1e11f39d246e7ce5ac50a1e-Paper.pdf.

M. Gabrié, A. Manoel, C. Luneau, N. Macris, F. Krzakala, L. Zdeborová, et al. Entropy and mutual information in models of deep neural networks. Advances in Neural Information Processing Systems, 31, 2018.

T. Garipov, P. Izmailov, D. Podoprikhin, D. P. Vetrov, and A. G. Wilson. Loss surfaces, mode connectivity, and fast ensembling of dnns. 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/be3087e74e9100d4bc4c6268cdbe8456-Paper.pdf.

R. Ge, F. Huang, C. Jin, and Y. Yuan. Escaping from saddle points—online stochastic gradient for tensor decomposition. In Conference on learning theory, pages 797–842. PMLR, 2015.

N. Golowich, A. Rakhlin, and O. Shamir. Size-independent sample complexity of neural networks. In Conference On Learning Theory, pages 297–299. PMLR, 2018.

S. Gunasekar, J. D. Lee, D. Soudry, and N. Srebro. Implicit bias of gradient descent on linear convolutional 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/0e98aeeb54acf612b9eb4e48a269814c-Paper.pdf.

B. D. Haeffele and R. Vidal. Global optimality in tensor factorization, deep learning, and beyond, 2015. URL https://arxiv.org/abs/1506.07540.

M. Hardt, B. Recht, and Y. Singer. Train faster, generalize better: Stability of stochastic gradient descent. In M. F. Balcan and K. Q. Weinberger, editors, Proceedings of The 33rd International Conference on Machine Learning, volume 48 of Proceedings of Machine Learning Research, pages 1225–1234, New York, New York, USA, 20–22 Jun 2016. PMLR. URL https://proceedings. mlr.press/v48/hardt16.html.

H. He, G. Huang, and Y. Yuan. Asymmetric valleys: Beyond sharp and flat local minima. arXiv preprint arXiv:1902.00744, 2019.

S. Hochreiter and J. Schmidhuber. Flat Minima. Neural Computation, 9(1):1–42, 01 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.1.1. URL https://doi.org/10.1162/neco.1997.9.1.1.

E. Hoffer, I. Hubara, and D. Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In NIPS, 2017.

A. Jacot, F. Gabriel, and C. 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.

S. Jastrzębski, Z. Kenton, D. Arpit, N. Ballas, A. Fischer, Y. Bengio, and A. Storkey. Three factors influencing minima in sgd. arXiv preprint arXiv:1711.04623, 2017.

T. Kato. Perturbation theory for linear operators, volume 132. Springer Science & Business Media, 2013.

A. Katok and B. Hasselblatt. Introduction to the modern theory of dynamical systems. Number 54. Cambridge university press, 1997.

G. Keller and C. Liverani. Stability of the spectrum for transfer operators. Annali della Scuola Normale Superiore di Pisa-Classe di Scienze, 28(1):141–152, 1999.

N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. arXiv preprint arXiv:1609.04836, 2016.

B. Kleinberg, Y. Li, and Y. Yuan. An alternative view: When does SGD escape local minima? In J. Dy and A. Krause, editors, 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. URL https://proceedings.mlr.press/v80/kleinberg18a.html.

D. Kondrashov, M. D. Chekroun, and M. Ghil. Data-driven non-markovian closure models. Physica D: Nonlinear Phenomena, 297:33–55, 2015. ISSN 0167-2789. doi: https://doi.org/10.1016/j.physd.2014.12.005. URL https://www.sciencedirect.com/science/article/pii/S0167278914002413.

L. Kong and M. Tao. Stochasticity of deterministic gradient descent: Large learning rate for multiscale objective function. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, Advances in Neural Information Processing Systems, volume 33, pages 2625–2638. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/1b9a80606d74d3da6db2f1274557e644-Paper.pdf.

B. O. Koopman. Hamiltonian systems and transformation in hilbert space. Proceedings of the national academy of sciences of the united states of america, 17(5):315, 1931.

M. Korda and I. Mezić. On convergence of extended dynamic mode decomposition to the koopman operator. Journal of Nonlinear Science, 28(2):687–710, 2018.

I. Kuzborskij and C. Lampert. Data-dependent stability of stochastic gradient descent. In International Conference on Machine Learning, pages 2815–2824. PMLR, 2018.

A. Lasota and M. C. Mackey. Chaos, fractals, and noise: stochastic aspects of dynamics, volume 97. Springer Science & Business Media, 1998. doi: 10.1007/978-1-4612-4286-4.

Y. Li and Y. Yuan. Convergence analysis of two-layer neural networks with relu activation. Advances in neural information processing systems, 30, 2017.

K. K. Lin and F. Lu. Data-driven model reduction, wiener projections, and the koopman-mori-zwanzig formalism. Journal of Computational Physics, 424:109864, 2021. ISSN 0021-9991. doi: https://doi.org/10.1016/j.jcp.2020.109864. URL https://www.sciencedirect.com/science/article/pii/S0021999120306380.

C. Liu, L. Zhu, and M. Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. Applied and Computational Harmonic Analysis, 2022.

C. Liverani. Invariant measures and their properties. a functional analytic point of view. Dynamical systems. Part II, pages 185–237, 2004.

E. Lobacheva, M. Kodryan, N. Chirkova, A. Malinin, and D. P. Vetrov. On the periodic behavior of neural network training with batch normalization and weight decay. Advances in Neural Information Processing Systems, 34, 2021.

A. Loukas, M. Poiitis, and S. Jegelka. What training reveals about neural network complexity. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. W. Vaughan, editors, Advances in Neural Information Processing Systems, 2021. URL https://openreview.net/forum?id=RcjW7p7z8aJ.

S. Ma, R. Bassily, and M. Belkin. The power of interpolation: Understanding the effectiveness of SGD in modern over-parametrized learning. In J. Dy and A. Krause, editors, Proceedings of the 35th International Conference on Machine Learning, volume 80 of Proceedings of Machine Learning Research, pages 3325–3334. PMLR, 10–15 Jul 2018. URL https://proceedings.mlr.press/v80/ma18a.html.

J. Mattingly, A. Stuart, and D. Higham. Ergodicity for sdes and approximations: locally lipschitz vector fields and degenerate noise. Stochastic Processes and their Applications, 101(2):185–232, 2002. ISSN 0304-4149. doi: https://doi.org/10.1016/S0304-4149(02)00150-3. URL https://www.sciencedirect.com/science/article/pii/S0304414902001503.

D. A. McAllester. Some pac-bayesian theorems. Machine Learning, 37(3):355–363, 1999.

S. Mei, T. Misiakiewicz, and A. Montanari. Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. In Conference on Learning Theory, pages 2388–2464. PMLR, 2019.

C. Mingard, G. Valle-Pérez, J. Skalse, and A. A. Louis. Is sgd a bayesian sampler? well, almost. Journal of Machine Learning Research, 22, 2021.

M. Mohri, A. Rostamizadeh, and A. Talwalkar. Foundations of machine learning. MIT press, 2018.

A. Montanari and Y. Zhong. The interpolation phase transition in neural networks: Memorization and generalization under lazy training. arXiv preprint arXiv:2007.12826, 2020.

E. Moulines and F. Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. In J. Shawe-Taylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, editors, Advances in Neural Information Processing Systems, volume 24. Curran Associates, Inc., 2011. URL https://proceedings.neurips.cc/paper/2011/file/40008b9a5380fcacce3976bf7c08af5b-Paper.pdf.

D. Needell, R. Ward, and N. Srebro. Stochastic gradient descent, weighted sampling, and the randomized kaczmarz algorithm. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Weinberger, editors, Advances in Neural Information Processing Systems, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/f29c21d4897f78948b91f03172341b7b-Paper.pdf.

Y. Nesterov. Introductory lectures on convex optimization: A basic course, volume 87. Springer Science & Business Media, 2003.

B. Neyshabur, S. Bhojanapalli, and N. Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. In International Conference on Learning Representations, 2018.

Q. Nguyen, M. C. Mukkamala, and M. Hein. On the loss landscape of a class of deep neural networks with no bad local valleys. In International Conference on Learning Representations, 2018.

K. Pitas. Dissecting non-vacuous generalization bounds based on the mean-field approximation. In International Conference on Machine Learning, pages 7739–7749. PMLR, 2020.

T. Quail, A. Shrier, and L. Glass. Predicting the onset of period-doubling bifurcations in noisy cardiac systems. Proceedings of the National Academy of Sciences, 112(30):9358–9363, 2015. doi: 10.1073/pnas.1424320112. URL https://www.pnas.org/doi/abs/10.1073/pnas.1424320112.

M. Raginsky, A. Rakhlin, and M. Telgarsky. Non-convex learning via stochastic gradient langevin dynamics: a nonasymptotic analysis. In Conference on Learning Theory, pages 1674–1703. PMLR, 2017.

A. Rakhlin. Applications of empirical processes in learning theory: algorithmic stability and generalization bounds. PhD thesis, Massachusetts Institute of Technology, 2006.

D. Rudolf and N. Schweizer. Perturbation theory for Markov chains via Wasserstein distance. Bernoulli, 24(4A):2610 – 2639, 2018. doi: 10.3150/17-BEJ938. URL https://doi.org/10.3150/17-BEJ938.

I. Safran and O. Shamir. On the quality of the initial basin in overspecified neural networks. In International Conference on Machine Learning, pages 774–782. PMLR, 2016.

L. Sagun, L. Bottou, and Y. LeCun. Eigenvalues of the hessian in deep learning: Singularity and beyond. arXiv preprint arXiv:1611.07476, 2016.

L. Sagun, U. Evci, V. U. Guney, Y. Dauphin, and L. Bottou. Empirical analysis of the hessian of over-parametrized neural networks. arXiv preprint arXiv:1706.04454, 2017.

A. M. Saxe, J. L. McClelland, and S. Ganguli. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks, 2013. URL https://arxiv.org/abs/1312.6120.

J. Sirignano and K. Spiliopoulos. Mean field analysis of deep neural networks. Mathematics of Operations Research, 47(1):120–152, 2022.

S. L. Smith and Q. V. Le. A bayesian perspective on generalization and stochastic gradient descent. In International Conference on Learning Representations, 2018.

J. Sokolić, R. Giryes, G. Sapiro, and M. R. Rodrigues. Robust large margin deep neural networks. IEEE Transactions on Signal Processing, 65(16):4265–4280, 2017.

D. Soudry, E. Hoffer, M. S. Nacson, S. Gunasekar, and N. Srebro. The implicit bias of gradient descent on separable data. The Journal of Machine Learning Research, 19(1):2822–2878, 2018.

L. Van der Maaten and G. Hinton. Visualizing data using t-sne. Journal of machine learning research, 9(11), 2008.

V. N. Vapnik. An overview of statistical learning theory. IEEE transactions on neural networks, 10 (5):988–999, 1999.

L. Venturi, A. S. Bandeira, and J. Bruna. Spurious valleys in one-hidden-layer neural network optimization landscapes. Journal of Machine Learning Research, 20:133, 2019.

U. von Luxburg and O. Bousquet. Distance-based classification with lipschitz functions. J. Mach. Learn. Res., 5:669–695, 2004.

Y. Wang, M. Chen, T. Zhao, and M. Tao. Large learning rate tames homogeneity: Convergence and balancing effect. 2021. doi: 10.48550/ARXIV.2110.03677. URL https://arxiv.org/abs/2110.03677.

A. Wilkinson. What are lyapunov exponents, and why are they interesting? Bulletin of the American Mathematical Society, 54(1):79–105, 2017.

M. O. Williams, C. W. Rowley, and I. G. Kevrekidis. A kernel-based approach to data-driven koopman spectral analysis. arXiv preprint arXiv:1411.2260, 2014.

S. Wojtowytsch. Stochastic gradient descent with noise of machine learning type. part i: Discrete time analysis, 2021. URL https://arxiv.org/abs/2105.01650.

H. Xu and S. Mannor. Robustness and generalization. Machine learning, 86(3):391–423, 2012.

C. Zhang, Q. Liao, A. Rakhlin, B. Miranda, N. Golowich, and T. Poggio. Theory of deep learning iib: Optimization properties of sgd. arXiv preprint arXiv:1801.02254, 2018.

C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning (still) requires rethinking generalization. Communications of the ACM, 64(3):107–115, 2021a.

J. Zhang, H. Li, S. Sra, and A. Jadbabaie. Neural network weights do not converge to stationary points: An invariant measure perspective. In K. Chaudhuri, S. Jegelka, L. Song, C. Szepesvari, G. Niu, and S. Sabato, editors, Proceedings of the 39th International Conference on Machine Learning, volume 162 of Proceedings of Machine Learning Research, pages 26330–26346. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/zhang22q.html.

Y. Zhang, W. Zhang, S. Bald, V. Pingali, C. Chen, and M. Goswami. Stability of sgd: Tightness analysis and improved bounds. arXiv preprint arXiv:2102.05274, 2021b.

L. Zhao, D. Tang, F. Lin, and B. Zhao. Observation of period-doubling bifurcations in a femtosecond fiber soliton laser with dispersion management cavity. Optics express, 12(19):4573–4578, 2004.

W. Zhou, V. Veitch, M. Austern, R. P. Adams, and P. Orbanz. Non-vacuous generalization bounds at the imagenet scale: a PAC-bayesian compression approach. In International Conference on Learning Representations, 2019. URL https://openreview.net/forum?id=BJgqqsAct7.

R. Zwanzig. Nonequilibrium statistical mechanics. Oxford university press, 2001.

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]
   (b) Did you describe the limitations of your work? [Yes]
   (c) Did you discuss any potential negative societal impacts of your work? [Yes]
   (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]

(b) Did you include complete proofs of all theoretical results? [Yes]

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] In the supplementary material

(b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] see section 5 and Appendix D

(c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes] See section 5 and Appendix D

(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)? [No]

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? [Yes]

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]