Learning Non-Vacuous Generalization
Bounds from Optimization

Chengli Tan*
Department of Mathematics and Statistics, Xi’an Jiaotong University
and
Jiangshe Zhang
Department of Mathematics and Statistics, Xi’an Jiaotong University
and
Junmin Liu
Department of Mathematics and Statistics, Xi’an Jiaotong University

July 23, 2024

Abstract

One of the fundamental challenges in the deep learning community is to theoretically understand how well a deep neural network generalizes to unseen data. However, current approaches often yield generalization bounds that are either too loose to be informative of the true generalization error or only valid to the compressed nets. In this study, we present a simple yet non-vacuous generalization bound from the optimization perspective. We achieve this goal by leveraging that the hypothesis set accessed by stochastic gradient algorithms is essentially fractal-like and thus can derive a tighter bound over the algorithm-dependent Rademacher complexity. The main argument rests on modeling the discrete-time recursion process via a continuous-time stochastic differential equation driven by fractional Brownian motion. Numerical studies demonstrate that our approach is able to yield plausible generalization guarantees for modern neural networks such as ResNet and Vision Transformer, even when they are trained on a large-scale dataset (e.g. ImageNet-1K).

Keywords: fractional Brownian motion, stochastic gradient descent, Rademacher complexity, deep learning

*The authors are supported in part by the National Key Research and Development Program of China under Grant 2020AAA0105601, in part by the National Natural Science Foundation of China under Grants 12371512 and 62276208, and in part by the Natural Science Basic Research Program of Shaanxi Province under Grant 2024JC-JCQN-02.
### 1 Introduction

Deep neural networks (DNNs) have shown remarkable performance in a wide range of tasks over the past decade \(\text{[Bengio et al., 2021]}\). A mystery is that they generalize surprisingly well on unseen data, though having far more trainable parameters than the number of training examples \(\text{[Belkin et al., 2019; Li et al., 2023]}\). This phenomenon of benign overfitting inevitably casts shadows on the classical theory of statistical learning, which posits that models with high complexity tend to overfit the training data, whereas models with low complexity tend to underfit the training data. To reconcile the conflicts, some researchers argue that this is due to the regularization incurred during training, either implicitly imposed via use of stochastic gradient descent (SGD) \(\text{[Advani et al., 2020; Barrett & Dherin, 2021; Smith et al., 2021; Sclocchi & Wyart, 2024]}\) or explicitly via batch normalization \(\text{[Ioffe & Szegedy, 2015]}\), weight decay \(\text{[Krogh & Hertz, 1992]}\), dropout \(\text{[Srivastava et al., 2014]}\), etc. However, Zhang et al. \(\text{(2017)}\) questioned this widely received wisdom because they found that DNNs are still able to achieve zero training error with randomly labeled examples, which apparently cannot generalize.

Prior to our work, there has been extensive study trying to explain the generalization behavior of DNNs and they roughly can be categorized into the following classes. The first class is the so-called norm-based bounds \(\text{[Neyshabur et al., 2015; Bartlett et al., 2017; Neyshabur et al., 2018; Golowich et al., 2018]}\) that are composed of the operator norm of layerwise weight matrices. However, recent studies suggest that these norm-based bounds might be problematic as they abnormally increase with the number of training examples \(\text{[Nagarajan & Kolter, 2019]}\). Moreover, norm-based bounds are numerically vacuous as they are even several orders of magnitude larger than the number of network parameters. The second class connects the generalization to the flatness of the solution \(\text{[Hochreiter, 2020]}\).
showing that flat minima usually generalize well. However, the flat minima alone do not suffice in explaining the generalization behavior of DNNs. For example, Dinh et al. (2017) argued that sharp minima can generalize as well by reparametrizing the function space and Wen et al. (2023) also successfully identified a class of non-generalizing flattest models for two-layer ReLU networks. Another class involves bounding the generalization error via a compression framework (Arora et al. 2018). Empirical results suggest that we can achieve almost non-vacuous bounds on realistic neural networks (Zhou et al. 2019, Lotfi et al. 2022). Nevertheless, this framework only proves the generalization of the compressed net, not of the true net found by the learning algorithm. Lastly, stability-based (Hardt et al. 2016) and information-theoretic (Xu & Raginsky 2017) bounds have also received a lot of attention, but both of them are limited in terms of practical value. Therefore, it remains a great challenge to search for generalization bounds that not only qualitatively but also quantitatively predict how well the model performs on the new-coming data.

Indeed, one critical issue that prevents the generalization bounds from practical usage is that the Rademacher complexity (Bartlett & Mendelson 2002) often is evaluated on a pre-specified hypothesis set (Neyshabur et al. 2015, Bartlett et al. 2017, Arora et al. 2019). But, in practice, we do not want to have a bound that holds uniformly over the pre-specified hypothesis set because we are more interested in a small portion of the hypothesis set that is accessible to the learning algorithm, and our goal is to address this issue. Since most tasks of modern neural networks are attacked by SGD and its variants, we are particularly interested in bounding the Rademacher complexity of the hypothesis set that SGD accesses during training.
Figure 1: Histogram of Hurst exponents for all coordinates of ResNet-20. For each coordinate, we first generate a series of stochastic gradient noise (SGN) and then estimate its Hurst exponent. If the elements of a time series are mutually independent, for example, in the case of the Brownian motion and the Lévy flight, the corresponding Hurst exponent would be $1/2$ (Embrechts 2009, Theorem 8.1.3). Otherwise, it would suggest that the elements are not independent.

To this end, we propose to model the discrete-time SGD recursion through the lens of stochastic differential equations (SDEs), an approach that has been widely used to study the escaping behavior of SGD (Jastrzebski et al. 2018, Nguyen et al. 2019, Xie et al. 2021). An important ingredient to studying SGD from this perspective is stochastic gradient noise (SGN), which is the difference between the stochastic gradient over a mini-batch and the true gradient over the full training set. In early attempts, by invoking the central limit theorem, SGN is assumed to be either Gaussian (Mandt et al. 2017, Li et al. 2017, Hu et al. 2019, Chaudhari & Soatto 2018, Xie et al. 2021) or Lévy stable (Simsekli et al. 2019, Zhang et al. 2020). These assumptions are compliant with an implicit constraint that SGN incurred at different iterations is mutually independent. However, as shown
in Figure [1] the temporal correlation of SGN is significant, suggesting that SGN is more reasonable to be fractional Gaussian noise (FGN) rather than Gaussian noise or from Lévy stable distribution. Recall that FGNs are the increments of fractional Brownian motion (FBM), a self-similar random process, thus allowing us to quantify the roughness of the optimization trajectory in terms of its Hausdorff dimension.

While the FBM-driven SDE representation of the SGD recursion has previously been investigated (Lucchi et al. 2022, Tan et al. 2023), they only focused on why SGD favors flat minima and a rigorous treatment of its relation to generalization is still lacking. At the core of our approach lies the fact that the optimization trajectory accessed by SGD during training is restricted to a small subset of the hypothesis space, which is fractal-like due to the incurred FGNs (Klingenhöfer & Zähle 1999, Lou & Ouyang 2016). We finally note that there already exist some generalization bounds that take the fractal structure into account, for example, see Simsekli et al. (2020), Camuto et al. (2021), Dupuis et al. (2023), Sachs et al. (2023). However, these approaches only present certain complexity measures such as the tail index to compare the generalization performance of one model against that of one another. Both of them are not able to quantitatively give a plausible estimate of the generalization error and their experimental results are restricted to using a constant learning rate, which is unrealistic for real-world applications. More seriously, when a classification model is trained with the cross-entropy loss, Camuto et al. (2021) could not even observe a clear negative or positive correlation between the complexity measure and the generalization error. By contrast, our approach can yield non-vacuous generalization bounds that predict the test loss (accuracy) well. Meanwhile, our bound is also model-agnostic, namely, we can efficiently estimate it for any DNNs with complex architectures such as ResNet (He et al. 2016) and Vision Transformer (Dosovitskiy et al. 2021).
The remainder of the paper is organized as follows. We first review some mathematical notions in Section 2 and then elaborate on the novel generalization bound for SGD in Section 3. Before concluding, we finally present the experimental results in Section 4.

2 Preliminaries

In this section, we briefly recap several concepts that we will use throughout this paper.

2.1 Fractional Brownian Motion

In probability theory, fractional Brownian motion (FBM), introduced by Mandelbrot & Van Ness (1968), is an extension of Brownian motion and is defined as follows.

Definition 1. Given a complete probability space \((\Lambda, \mathcal{B}, \mathbb{P})\), FBM is an almost surely continuous centered Gaussian process \(\{\Gamma(t), t \geq 0\}\) with covariance function

\[
\mathbb{E}[\Gamma(t)\Gamma(s)] = \frac{1}{2} \left(t^{2H} + s^{2H} - (t-s)^{2H}\right),
\]

where \(H\) is a real value in \((0, 1)\) and is often referred to as the Hurst exponent.

Unlike Brownian motion and other stochastic processes, the increments of FBM need not be independent. In particular, when \(H \in (0, 1/2)\), the increments of FBM are negatively correlated and exhibit short-range dependence, implying that it is more likely to overturn past changes. By contrast, FBM shows long-range dependence when \(H \in (1/2, 1)\). That is, if it was increasing in the past, it is persistent to keep the trend and vice versa. In particular, when \(H = 1/2\), FBM reduces to the standard Brownian motion. To gain some intuition, we plot several sample paths of FBM in Figure 2 with different Hurst exponents. One can observe that, when the Hurst exponent \(H\) is small, the sample path is seriously ragged.
Figure 2: Sample paths of FBM in two-dimensional space. The colors indicate the evolution over time. The Hurst exponent $H$ corresponds to the raggedness of the sample path, with a higher value leading to a smoother motion.

contrast, it appears dramatically smoother when the Hurst exponent $H$ becomes relatively larger.

2.2 Fractal Dimension

The notion of dimension is central to our analysis. One that we are most familiar with is the ambient dimension. Roughly speaking, a dimension describes how much space a set occupies near each of its points. For instance, $\mathbb{R}^d$ as a vector space has an ambient dimension of $d$ since $d$ different coordinates are required to identify a point in this space.

The fractal dimension, however, extends this notion to the fractional case. While it turns out to be particularly useful in many mathematical fields such as number theory and dynamical systems, there are many different ways to define fractal dimension, and not all the definitions are equivalent to each other. Of the wide variety of fractal dimensions, we focus on probably the most important box-counting and Hausdorff dimensions.

**Box-counting dimension.** Suppose $\mathcal{W}$ is a non-empty subset of $\mathbb{R}^d$, and the diameter of $\mathcal{W}$ is defined as $\text{diam}(\mathcal{W}) = \sup\{|x - y| : x, y \in \mathcal{W}\}$. Let $N_\delta(\mathcal{W})$ be the least number of
subsets \( \{ W_i \} \) of diameter at most \( \delta \) to cover \( W \), that is, \( W \subseteq \bigcup_{i=1}^{N(\delta)} W_i \) and \( \text{diam}(W_i) \leq \delta \) for each \( i \). Then, the lower and upper box-counting dimensions of \( W \), respectively, are defined as

\[
\text{dim}_B W = \lim_{\delta \to 0} \frac{\log N_\delta(W)}{\log(1/\delta)},
\]

and

\[
\overline{\text{dim}}_B W = \lim_{\delta \to 0} \frac{\log N_\delta(W)}{\log(1/\delta)}.
\]

Note that \( \text{dim}_B W \leq \overline{\text{dim}}_B W \) and if the equality holds, the box-counting dimension of \( W \) is then denoted by

\[
\text{dim}_B W = \lim_{\delta \to 0} \frac{\log N_\delta(W)}{\log(1/\delta)}.
\]

The popularity of the box-counting dimension is largely due to its intuitive definition and relative ease of empirical calculation. By contrast, the Hausdorff dimension, which is described below, is in terms of measure theory and is mathematically convenient to work with. Consequently, a disadvantage of the Hausdorff dimension is that it is often difficult to estimate by computational methods. However, for a proper understanding of fractal geometry, familiarity with the Hausdorff dimension is essential.

**Hausdorff dimension.** Let \( \{ W_i \}_{i=1}^\infty \) be a \( \delta \)-cover of a non-empty bounded set \( W \), and for each \( \alpha \geq 0 \), we call

\[
\mathcal{H}^\alpha(W) = \lim_{\delta \to 0} \left\{ \sum_{i=1}^{\infty} \text{diam}(W_i)^\alpha : W \subseteq \bigcup_{i=1}^{\infty} W_i, \text{diam}(W_i) < \delta \right\},
\]

the \( \alpha \)-dimensional Hausdorff measure of \( W \). Usually, it equals 0 or \( \infty \). The critical value of \( \alpha \) at which \( \mathcal{H}^\alpha(W) \) jumps from \( \infty \) to 0 is referred to as the Hausdorff dimension. Rigorously, it is defined as

\[
\dim_H W = \inf \{ \alpha \geq 0 : \mathcal{H}^\alpha(W) = 0 \} = \sup \{ \alpha \geq 0 : \mathcal{H}^\alpha(W) = \infty \}.
\]
While these two kinds of dimensions are the same under some regularity conditions \cite{Mattila1999}*{Theorem 5.7}, they are not equivalent to each other. For example, considering the set of rationals in \([0, 1]\), the Hausdorff dimension is 0, while the box-counting dimension is 1. In general, it holds that \( \dim_H W \leq \dim_B W \).

### 3 Non-Vacuous Generalization Bound for SGD

Assume we have access to a training set of independent and identically distributed (i.i.d.) data points,

\[
S = \{(x_1, y_1), \ldots, (x_m, y_m)\} = \{z_1, \ldots, z_m\},
\]

where \(x \in \mathcal{X}\) denotes the features, \(y \in \mathcal{Y}\) denotes the labels, and \(Z = \mathcal{X} \times \mathcal{Y}\) denotes the data space that follows an unknown data distribution \(\mathcal{D}\). The goal of supervised learning is to choose a suitable hypothesis \(f_w : \mathcal{X} \rightarrow \mathcal{Y}\), parameterized by a vector of network parameters \(w \in \mathbb{R}^d\), so that the generalization error (i.e. the risk on previously unseen data),

\[
R_D(w) = \mathbb{E}_{z \sim \mathcal{D}}[\ell(w, z)] = \mathbb{E}_{(x, y) \sim \mathcal{D}}[\mathcal{L}(f_w(x), y)]
\]

is small. Here, \(\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}_+\) is a non-negative loss function, and \(\ell : \mathbb{R}^d \times Z \rightarrow \mathbb{R}_+\) is the composition of the loss and the hypothesis, which will also referred to as “loss”, with a slight abuse of notation.

However, due to the unknown data distribution \(\mathcal{D}\), we are not able to minimize \(R_D(w)\) directly. Instead, we can only minimize the empirical error over the training set \(S\), namely,

\[
R_S(w) = \frac{1}{m} \sum_{i=1}^{m} \ell(w, z_i).
\]

Notice that the difference \(R_D(w) - R_S(w)\) is referred to as the generalization gap. Particularly, in the realizable case where the empirical error is zero, the generalization gap is
interchangeable with the generalization error.

### 3.1 Problem Setup

Starting from an initialization point \( w_0 \in \mathbb{R}^d \), the SGD algorithm recursively updates the weights of the neural network as follows,

\[
w_{k+1} = w_k - \eta \nabla \hat{\ell}_{\Omega_k}(w_k),
\]

where \( \eta \) is the learning rate and \( \nabla \hat{\ell}_{\Omega_k}(w_k) \) is an unbiased estimate of the true gradient, which is computed by

\[
\nabla \hat{\ell}_{\Omega_k}(w_k) = \frac{1}{|\Omega_k|} \sum_{z \in \Omega_k} \nabla \ell(w_k, z),
\]

where \( \Omega_k \) is a set of examples (i.e. mini-batch) that are i.i.d. drawn from \( S \) and \( b = |\Omega_k| \) is the mini-batch size. Particularly, when \( \Omega_k = S \), SGD becomes the full-batch gradient descent (GD). While the SGD algorithm is random, once the training set \( S \), the initialization point \( w_0 \), and the training steps \( K \) are fixed, the total number of optimization trajectories (i.e. the collection of weights throughout training) \( M(m, b, K) \) is indeed finite (though very large). To see this, notice that there are only finitely many subsets that \( \Omega_k \) can take. For example, in the case of with-replacement sampling, there are in total \( m^b \) mini-batches to choose from at every step. By contrast, in the case of without-replacement sampling, this number can be further reduced to \( \binom{m}{b} \). Of course, here we require that there are no other sources of stochasticity during training such as perturbing the weights with random noise.

Many studies [Zhu et al., 2019; Amir et al., 2021; Wu & Su, 2023] have shown that training neural networks with the stochastic gradient \( \nabla \hat{\ell}_{\Omega_k}(w_k) \) generally outperforms with the true gradient \( \nabla \hat{\ell}_S(w_k) \) because of the incurred stochastic gradient noise (SGN), which is defined as

\[
\zeta_k = \nabla \hat{\ell}_{\Omega_k}(w_k) - \nabla \hat{\ell}_S(w_k).
\]
If one assumes that the learning rate $\eta$ is sufficiently small and $\zeta_k$ follows a zero-mean distribution, the SGD recursion (1) can be seen as a first-order discretization of a continuous-time SDE (Li et al. 2017).

Recently, perspectives from SDEs have provided many insights on studying the generalization behavior of DNNs through the asymptotic convergence rate and local dynamic behavior of SGD (Mandt et al. 2017, Simsekli et al. 2019, Xie et al. 2021, Tan et al. 2023, Gess et al. 2024). In our analysis, we will consider the case where SGD is viewed as the Euler-Maruyama discretization of the following SDE,

$$dw_t = -\mu(w_t, t)dt + \sigma(w_t, t)d\Gamma(t),$$

(2)

where $\mu(w_t, t) \in \mathbb{R}^d$ is the drift coefficient, $\sigma(w_t, t) \in \mathbb{R}^d$ is the diffusion coefficient, and $\Gamma(t)$ represents a $d$-dimensional FBM with Hurst exponents $H = (H_1, \ldots, H_d)$. For simplicity, we also assume that the random noise of different coordinates is mutually independent. Such class of SDEs admits SGN produced at different iterations to be mutually interdependent, which significantly varies from previous studies where SGN is assumed either to be Gaussian (Mandt et al. 2017, Li et al. 2019) or follow a Lévy stable distribution (Simsekli et al. 2020, Dupuis & Simsekli 2024).

A pairwise correspondence between discrete-time SGD recursion (1) and continuous-time SDE driven by FBM (2) can be easily established. For a finite number $K$ of training steps, let $W^{0{:}K}_{\xi|S,w_0} = \{w_0, w_1, \ldots, w_K\}$ be the optimization trajectory that achieved by a specific run indexed by $\xi \in \{1, \ldots, M(m, b, K)\}$ of SGD. When the learning rate $\eta$ is small enough, for a given $t \in [k\eta, (k + 1)\eta)$, we can always define a stochastic process $\hat{w}_t$ as the interpolation of two successive iterates $w_k$ and $w_{k+1}$ such that $\hat{w}_{k\eta} = w_k$ for all $k$. This approach is frequently adopted in SDE literature (Mishura & Shevchenko 2008) and allows the trajectory to be continuous to represent the SGD recursion. Therefore, $W^{0{:}K}_{\xi|S,w_0}$ always
can be viewed as a sample path of the solution to SDE (2) in a time frame, say, without loss of generality, \( W_{\xi,S,w_0}([0, 1]) = \{ w_t, t \in [0, 1] \} \). Consequently, for a training set \( S \) and an initialization point \( w_0 \), the hypothesis set that SGD accesses is essentially a tiny space and can be defined as

\[
\mathcal{W}_{S,w_0} = \bigcup_{\xi=1}^{M(m,b,K)} W_{\xi,S,w_0}([0, 1]).
\]

While \( w_0 \) is randomly drawn from a probability distribution, unless otherwise specified, our discussion below always assumes that \( w_0 \) is fixed so that our analysis can be greatly simplified. This is because most SGD solutions trained from different initialization points belong to the same basin in the loss landscape after proper permutation (Entezari et al. 2022, Ainsworth et al. 2023). As a result, any generalization bounds conditioned on \( w_0 \) can also be applied to predict the generalization performance of SGD solutions that are trained from another initialization point. For simplicity of notation, we will omit the dependence on \( w_0 \) and simply write \( \mathcal{W}_S \) instead. Further, we write \( \mathcal{G}_S \) to denote the loss functions associated with \( \mathcal{W}_S \) mapping from \( Z = \mathcal{X} \times \mathcal{Y} \) to \( \mathbb{R}_+ \),

\[
\mathcal{G}_S = \{ g_w = \ell(w, z) \mid w \in \mathcal{W}_S \}.
\]

To remove the dependence on \( S \), we can take a union over \( S \in Z^m \), yielding \( \mathcal{W} = \bigcup_{S \in Z^m} \mathcal{W}_S \) and \( \mathcal{G} = \bigcup_{S \in Z^m} \mathcal{G}_S \) to represent the set of all possible parameters and loss functions. For any \( \varepsilon > 0 \), our goal is to bound the following term

\[
P \left[ \sup_{w \in \mathcal{W}} | \hat{R}_S(w) - R(w) | \geq \varepsilon \right],
\]

which is algorithm-dependent and differs from what is usually studied where \( \mathcal{W} \) is replaced by a pre-specified hypothesis set. In the sequel, we will present the main result in terms of the empirical Rademacher complexity \( \mathcal{R}_S(\mathcal{G}) \) (Bartlett & Mendelson 2002), which is
defined as
\[ R_S(G) = \frac{1}{m} \mathbb{E} \sigma \sim \{\pm 1\}^m \left[ \sup_{g_w \in G} \sum_{i=1}^m \sigma_i g_w(z_i) \right] = \frac{1}{m} \mathbb{E} \sigma \sim \{\pm 1\}^m \left[ \sup_{w \in W} \sum_{i=1}^m \sigma_i \ell(w, z_i) \right], \]
where the Rademacher variables \( \sigma \) are i.i.d. with \( \mathbb{P}(\sigma = \pm 1) = 1/2 \). Let \( \mathcal{H} = G \circ S \) be the set of all possible loss evaluations that a loss function \( g_w \in G \) can achieve over the training set \( S \), namely,
\[ \mathcal{H} = G \circ S = \{ h_w = (g_w(z_1), \ldots, g_w(z_m)) | g_w \in G \}. \]
We can further observe that the value of \( R_S(G) \) is the same as the Rademacher complexity \( R(\mathcal{H}) \) of the set \( \mathcal{H} \subset \mathbb{R}_+^m \).

In the following section, we aim to control \( R(\mathcal{H}) \) by taking into account the Hausdorff dimension of the sample paths of the solution to SDE (2). The Hausdorff dimension determines the raggedness of the sample path and characterizes the dynamic behavior of SGD around the local minimum.

### 3.2 Main Assumptions

We will first present several assumptions used in our theoretical analysis.

**Assumption 1.** The loss function \( \ell : \mathbb{R}^d \times \mathcal{Z} \mapsto \mathbb{R}_+ \) is bounded in \([0, 1]\) and \( L \)-Lipschitz \((L \geq 1)\) continuous with respect to its first argument.

The boundedness assumption is standard in the literature, for example, see Shalev-Shwartz & Ben-David (2014) and Mohri et al. (2018). Furthermore, if a mapping satisfies the Lipschitz continuity, then the Hausdorff dimension of the image is no greater than the Hausdorff dimension of the preimage (Falconer 2004, Proposition 3.3). This Lipschitz assumption can be easily satisfied, if the gradient of the loss function is uniformly bounded for any \( w \in \mathbb{R}^d \), for example, by gradient clipping.
Assumption 2. The drift coefficient $\mu(w_t, t)$ and diffusion coefficient $\sigma(w_t, t)$ in SDE (2) are both bounded vector fields on $\mathbb{R}^d$.

This assumption is reasonable due to the existence of batch normalization (Ioffe & Szegedy 2015), weight decay (Krogh & Hertz 1992), and other popular tricks. Under this assumption, the existence and uniqueness of solutions to SDE (2) are guaranteed if the Hurst exponent $H$ is larger than $1/4$ (Lyons & Qian 2002). However, the current study on the Hausdorff dimension of the sample paths of the solution to SDE (2) is only limited to the case where the Hurst exponent $H$ is the same for all coordinates (Lou & Ouyang 2016). This obviously is not true for real-world neural networks that have millions (even billions) number of parameters (cf. Figure 1). Luckily, when the mini-batch size is small, the norm of SGN is always much larger than the norm of the true gradient (cf. Figure 3), suggesting that the training process is dominated by the diffusion term so that we can instead use the known results of multi-dimensional FBM. In light of this, we can further impose the assumption below.

Figure 3: Norm of the true gradient and the stochastic gradient as a function of training epoch, where the mini-batch size is 128.
Assumption 3. For each specific run indexed by $\xi \in \{1, \ldots, M(m, b, K)\}$ of SGD, the Hausdorff dimension of the sample path of the solution to SDE (2), $\dim_H W_\xi S([0, 1])$, is upper bounded by the Hausdorff dimension of the sample path of the driven FBM, which is explicitly given by

$$\dim_H \Gamma_\xi S([0, 1]) = 1 + \frac{1}{H_k} \sum_{i=1}^{k} (H_k - H_i),$$

(3)

where the Hurst exponents are sorted such that $0 < H_1 \leq H_2 \leq \cdots \leq H_d < 1$ and $k$ is determined by the inequality $\sum_{i=1}^{k-1} H_i \leq 1 \leq \sum_{i=1}^{k} H_i$ (Xiao 1995, Theorem 2.1).

Furthermore, we assume the data distribution $\mathcal{D}$ is supported on a countable set so that $\dim_H \mathcal{W} = \dim_H \bigcup_{S \in \mathbb{Z}^m} W_S \leq \sup_{S \in \mathbb{Z}^m} \max_{\xi \in \{1, \ldots, M(m, b, K)\}} \dim_H \Gamma_\xi S([0, 1])$.

We note that the countability assumption is crucial to our results. Thanks to this condition, we are able to invoke the countable stability (Falconer 2004, Section 3.2) of the Hausdorff dimension to control the upper bound of $\dim_H \mathcal{W}$. This assumption generally holds for image-based datasets, where each pixel is an integer from 0 to 255. Moreover, we can further require that the Hausdorff dimension $\dim_H \Gamma_\xi S([0, 1])$ corresponding to the driven FBM does not depend on the order of the mini-batches. Namely, for any specific run of SGD, it remains the same. This can be easily checked by shuffling the order of mini-batches (cf. Table 1). Furthermore, we can also observe that $\dim_H \Gamma_\xi S([0, 1])$ remains approximately the same even when the model is trained with different training sets and initialization points. Therefore, the Hausdorff dimension $\dim_H \Gamma_\xi S([0, 1])$ estimated under any specific run of SGD essentially provides a plausible upper bound over $\dim_H \mathcal{W}$, which is particularly useful in practice.

Assumption 4. Let $\mathcal{W}$ be a non-empty bounded subset of $\mathbb{R}^d$ and there exists a Borel measure $\nu$ on $\mathbb{R}^d$ and positive numbers $a$, $b$, $r_0$ and $\kappa$ such that $0 < \nu(\mathcal{W}) \leq \nu(\mathbb{R}^d) < \infty$.
Table 1: Effects of different sources of stochasticity on Hausdorff dimension $\dim_H \Gamma_{\xi|S([0,1])}$. The first row quantifies how $\dim_H \Gamma_{\xi|S([0,1])}$ is affected by the different initialization points of the neural network (ResNet-20) under the same training set. When the neural network is initialized with the same weights, the second row describes how $\dim_H \Gamma_{\xi|S([0,1])}$ changes with the training set (i.e. random subsets of CIFAR-10). Finally, when both the initialization point and the training set are the same, the last row further studies the effect of the order of the mini-batches.

| Number of training examples (per class) | 1000 | 2000 | 3000 | 4000 |
|----------------------------------------|------|------|------|------|
| Initialization point                    | 3.12 ± 0.07 | 2.84 ± 0.09 | 2.79 ± 0.07 | 2.70 ± 0.06 |
| Training set                           | 3.02 ± 0.09 | 2.84 ± 0.06 | 2.78 ± 0.04 | 2.71 ± 0.04 |
| Mini-batch order                       | 3.03 ± 0.09 | 2.83 ± 0.04 | 2.77 ± 0.02 | 2.71 ± 0.04 |

and for $w \in \mathcal{W}$

$$0 < ar^\kappa \leq \nu(B(w, r)) \leq br^\kappa < \infty, \quad 0 < r < r_0,$$

where

$$B(w, r) = \{w' \in \mathcal{W} ||w - w'|| < r\}.$$

This so-called Ahlfors regularity is often used in fractal geometry to ensure the set is regular enough so that the Hausdorff dimension is equivalent to the box-counting dimension [Mattila 1999, Theorem 5.7]. That is, under this assumption, we have $\dim_B \mathcal{W} = \dim_H \mathcal{W} = \kappa$. As a result, we can use the covering number techniques. Recall that $\mathcal{W}$ is a collection of sample paths of the solution to SDE (2) and thus we have $\kappa \geq 1$ as well.
3.3 Upper Bound

Based on these assumptions, we are ready to present an upper bound over $R(\mathcal{H})$.

**Theorem 1.** Let Assumptions 1-4 hold. For any i.i.d. sample $S \in Z^m$, there always exist a constant $c \geq 1$ such that the following inequality holds:

$$R(\mathcal{H}) \leq \frac{6\text{diam}(\mathcal{H})}{m} \sqrt{2 \dim_{\mathcal{H}} \mathcal{W}} \left[ \left( \log \frac{\sqrt{mL}}{\text{diam}(\mathcal{H})} + \log c \right)^{-1/2} + \left( \log \frac{\sqrt{mL}}{\text{diam}(\mathcal{H})} + \log c \right)^{1/2} \right].$$

**Proof.** Fix $r_k = \text{diam}(\mathcal{H})/2^k$ and $\tilde{r}_k = \text{diam}(\mathcal{H})/2^k \sqrt{mL}$. Then, for any $w, w' \in \mathcal{W}$ satisfying $\|w - w'\| \leq \tilde{r}_k$, we always have for the corresponding $h_w, h_{w'} \in \mathcal{H}$ the following inequality

$$\|h_w - h_{w'}\| = \left( \sum_{i=1}^{m} |g_w(z_i) - g_{w'}(z_i)|^2 \right)^{1/2} \leq \left( mL^2\|w - w'\|^2 \right)^{1/2} \leq r_k,$$

implying that $N_{r_k}(\mathcal{H}) \leq N_{\tilde{r}_k}(\mathcal{W})$.

According to Assumption 4 we know that $\mathcal{W}$ is regular enough so that $\dim_{\mathcal{H}} \mathcal{W} = \dim_{\mathcal{H}} \mathcal{W}$. This means that, when $\tilde{r}_k$ approaches to zero, we have

$$\dim_{\mathcal{H}} \mathcal{W} = \lim_{\tilde{r}_k \to 0} \frac{N_{\tilde{r}_k}(\mathcal{W})}{- \log \tilde{r}_k}.$$

Therefore, for any $\varepsilon > 0$, there always exists an integer $k_{\varepsilon}$ such that for any $k \geq k_{\varepsilon}$

$$\log N_{\tilde{r}_k}(\mathcal{W}) \leq (\dim_{\mathcal{H}} \mathcal{W} + \varepsilon)(- \log \tilde{r}_k).$$

Choosing $\varepsilon = \dim_{\mathcal{H}} \mathcal{W}$ and $c = \max\left( \frac{N_{r_1}(\mathcal{W})}{\tilde{r}_1}, \ldots, \frac{N_{r_{k_{\varepsilon}}}(\mathcal{W})}{\tilde{r}_{k_{\varepsilon}}}, 1 \right)$, then we have for all $k \in \mathbb{N}_+$

$$N_{\tilde{r}_k}(\mathcal{W}) \leq c(\tilde{r}_k)^{-2\dim_{\mathcal{H}} \mathcal{W}}.$$  

Substituting $\tilde{r}_k$ in, yielding

$$\log N_{\tilde{r}_k}(\mathcal{W}) \leq \log c + 2 \dim_{\mathcal{H}} \mathcal{W} (k + \log \frac{\sqrt{mL}}{\text{diam}(\mathcal{H})}) \leq 2 \dim_{\mathcal{H}} \mathcal{W} (k + \log \frac{\sqrt{mLc}}{\text{diam}(\mathcal{H})}).$$
Write $\beta = \log \sqrt{mL}/\text{diam}(\mathcal{H})$, we have

$$\sqrt{\log \mathcal{N}_r(\mathcal{H})} \leq \sqrt{\log \mathcal{N}_r^{\sigma}(\mathcal{W})} \leq \sqrt{2 \dim \mathcal{W}(k + \beta + \log c)}$$

$$\leq \sqrt{2 \dim \mathcal{W}(\frac{k}{2\sqrt{\beta} + \log c} + \sqrt{\beta + \log c}),}$$

where the last inequality is due to the fact that $\sqrt{k + x} \leq \sqrt{x} + k/2\sqrt{x}$ for all $x > 0$.

By appealing to Dudley’s lemma (Shalev-Shwartz & Ben-David 2014, Lemma 27.5), the following inequality holds

$$\mathcal{R}(\mathcal{H}) \leq \frac{6 \text{diam}(\mathcal{H})}{m} \sqrt{2 \dim \mathcal{W}} \left[ \left( \log \frac{\sqrt{mL}}{\text{diam}(\mathcal{H})} + \log c \right)^{-1/2} + \left( \log \frac{\sqrt{mL}}{\text{diam}(\mathcal{H})} + \log c \right)^{1/2} \right],$$

thus completing the proof.

Based on the Rademacher complexity $\mathcal{R}(\mathcal{H})$, we are now ready to present the bound over the maximal generalization gap.

**Theorem 2.** Let Assumptions 1-4 hold. Then, for any $\tau > 0$, with probability at least $1 - \tau$ over the draw of an i.i.d. sample $S \in \mathcal{Z}^m$, there always exists a constant $c \geq 1$ such that the following inequality holds for all $w \in \mathcal{W}$,

$$R_D(w) - R_S(w) \leq \frac{12 \text{diam}(\mathcal{H})}{m} \sqrt{2 \dim \mathcal{W}} \left[ \left( \log \frac{c\sqrt{mL}}{\text{diam}(\mathcal{H})} \right)^{-1/2} + \left( \log \frac{c\sqrt{mL}}{\text{diam}(\mathcal{H})} \right)^{1/2} \right] + 3\sqrt{\frac{1}{2m} \log \frac{2}{\tau}}.$$

**Proof.** This is a direct consequence of Mohri et al. (2018, Theorem 3.3).

**Remark 1.** In the classical literature where the fractal structure of the learned hypothesis set is not taken into consideration, the Rademacher complexity $\mathcal{R}(\mathcal{H})$ scales as $O(\sqrt{\log m})$ if we assume $\text{diam}(\mathcal{H}) \propto \sqrt{m}$, see Shalev-Shwartz & Ben-David (2014, Example 27.2). As a result, this suggests that the generalization bound would increase with the number of training examples, which is obviously contradictory to the empirical results. By contrast, our result suggests that the above bound can decrease with the number of training examples in a sublinear rate, namely, $O(1/\sqrt{m})$. 

18
Remark 2. For the simplest case where \( \log c \sqrt{mL/\text{diam}(\mathcal{H})} = 1 \), the above bound reduces to

\[
O\left(\frac{\text{diam}(\mathcal{H}) \sqrt{\text{dim}_H \mathcal{W}}}{m}\right),
\]

which implies that the generalization gap continues to increase until the training process saturates. In addition, it also suggests that optimizing in the flat regions of the loss landscape indeed decreases the generalization gap. This is because the optimization trajectories generated in the flat regions are smoother in terms of lower values of \( \text{dim}_H \mathcal{W} \) (e.g. in the case of small vs. large mini-batch size). However, it should be emphasized that a small generalization gap does not necessarily dictate a small generalization error (requiring the training loss to be small as well). For example, for an untrained neural network, the generalization gap between the training set and the test set is small, whereas the generalization error on the test set could be very large.

Remark 3. Note that our bound does not explicitly depend on the number of trainable parameters \( d \). Instead, the Hausdorff dimension \( \text{dim}_H \mathcal{W} \) plays a similar role and quantifies the “effective” complexity of the hypothesis set because \( \text{dim}_H \mathcal{W} \) in general is much smaller than \( d \). Moreover, the effects of other important ingredients such as the network architecture and the initialization method are implicitly absorbed in \( \text{diam}(\mathcal{H}) \) as well.

3.4 Estimation

The generalization bound of Theorem 2 can be easily computed in practice, and we estimate it by the formula below:

\[
\varrho_{\text{bound}} = \frac{12\text{diam}(\mathcal{H})}{m} \sqrt{2 \text{dim}_H \mathcal{W}} \left[ \left( \log \frac{\sqrt{mL}}{\text{diam}(\mathcal{H})} \right)^{-1/2} + \left( \log \frac{\sqrt{mL}}{\text{diam}(\mathcal{H})} \right)^{1/2} \right].
\]

Compared to Theorem 2, notice that we have omitted the nuisance factor \( \log c \) because it is essentially an artifact due to the proof and its influence is limited even though the value
of \(c\) is very large. Indeed, if \(\mathcal{W}\) is a self-similar set or generated from an iterated function system \(\text{[Falconer 2004, Camuto et al. 2021]}\), the value of \(c\) approximately equals to one. Apart from the already known number of training examples \(m\), there are three remaining terms to be calculated.

We start with the Lipschitz constant \(L\). Although we have assumed \(L\) as a constant that universally holds for any \(w, w' \in \mathbb{R}^d\), in practice, it should be restricted to the space of \(\mathcal{W}\) and therefore corresponds to a much smaller value. Recall that the Lipschitz continuity can be guaranteed if the gradient of the loss function is bounded, namely, \(\|\nabla \ell_w(w, z)\| \leq L\) for any \(w \in \mathcal{W}\) and \(z \in \mathcal{Z}\). Moreover, we have at each iteration \(\text{Var}(\nabla \ell_{w_k}(w_k, z)) = |\Omega_k| \text{Var}(\nabla \hat{\ell}_{\Omega_k}(w_k))\). Therefore, we can approximate \(L\) with the maximum value of \(|\Omega_k|^{1/2}\|\nabla \hat{\ell}_{\Omega_k}(w_k)\|\) throughout training.

Next, we are going to estimate \(\text{diam}(\mathcal{H})\). To this end, we need to calculate the per-example loss on the full training set until the training is finished. Subsequently, we can estimate the diameter of \(\mathcal{H} \subset \mathbb{R}_+^m\) by computing the smallest bounding ball \(\mathbf{1}\). However, this approach is computationally prohibitive when \(m\) is large. To circumvent this issue, we can alternatively approximate \(\text{diam}(\mathcal{H})\) with

\[
\left[ (\ell(w_K, z_1) - \ell(w_0, z_1))^2 + \cdots + (\ell(w_K, z_m) - \ell(w_0, z_m))^2 \right]^{1/2},
\]

where \(w_0\) and \(w_K\) are the vectors of network parameters at initialization and the end of training. This is because that the loss is always non-negative and generally tends to decrease during training.

We now continue to compute \(\text{dim}_H \Gamma_{\xi \parallel S([0, 1])}\) according to Equation \(\text{[3]}\) to give an estimation of \(\text{dim}_H \mathcal{W}\), for which we first need to estimate the Hurst exponent \(\text{[2]}\) for each coordinate of the neural network. To produce a series of SGN for a neural network, we run

\[1\] The code is available at [https://github.com/hirsch-lab/cyminiball](https://github.com/hirsch-lab/cyminiball).

\[2\] The code is available at [https://github.com/CSchoel/nolds](https://github.com/CSchoel/nolds).
through the full training set to calculate the full-batch gradient. Then, we feed a number of mini-batches into the neural network, and as a result, we can obtain a series of SGN by subtracting the full-batch gradient from the mini-batch gradient. Notice that for very large neural networks that contain millions (even billions) of trainable parameters, due to limited memory, we are not able to generate a series of SGN for each coordinate. In this case, we can randomly sample a small portion of coordinates and we find that the estimation is robust to the number of used coordinates (see Supplementary Material, Figure 1).

Finally, we want to emphasize that these terms, theoretically, should be better estimated using the union of multiple runs with different seeds. In practice, however, we find that they often lead to similar results. Therefore, we choose to estimate $\varrho_{\text{bound}}$ using a single run, which is particularly useful in scenarios such as neural architecture search where an instant measure is required to compare against different runs.

4 Numerical Studies

In this section, we present the experimental results to demonstrate the efficacy of the proposed generalization bound.

4.1 Implementation Details

We consider three publicly available datasets—CIFAR-10, CIFAR-100 [Krizhevsky et al. 2009], and ImageNet-1K [Deng et al. 2009]. CIFAR-10 and CIFAR-100 are composed of 50,000 training examples and 10,000 test examples that are equally divided into 10 and 100 classes. By contrast, ImageNet-1K is a large-scale dataset that consists of 1000 classes and contains approximately one million training images and 50,000 validation images. We do not use data augmentation in all experiments, since doing so will prevent the model
from consistently reaching low cross-entropy loss and impose uncontrollable effects on SGN as the training examples are no longer i.i.d. distributed (Dziugaite et al. 2020, Jiang et al. 2020).

Unless otherwise specified, optimization uses SGD with momentum of 0.9 and weight decay of $5.0 \times 10^{-4}$. By default, we use a mini-batch size of 128, a learning rate of 0.05, and a cosine learning rate scheduler to ensure that the models can fit the training set completely. Determining when to stop the training process is important to quantitatively assess the generalization bounds, especially for those that can only be calculated after the training is finished. Stopping too early or too late may produce different results. Slightly different from Jiang et al. (2020), Dziugaite et al. (2020), we terminate the training process when the training accuracy reaches the threshold of 99.5%. This is because decreasing the cross-entropy loss to a very low value will result in severe overfitting.

### 4.2 Number of Training Examples

Increasing the number of training examples generally will promote the generalization performance of DNNs (Kaplan et al. 2020). While this observation is obvious, a non-negligible fact is that there are still a large number of generalization bounds that fail to (correctly) reveal this correlation (Nagarajan & Kolter 2019).

In the following, we aim to investigate how the proposed bound $\varrho_{\text{bound}}$ changes with the number of training examples. First, we need to generate a bunch of subsets as follows: for CIFAR-10, we gradually increase the number of training examples (per class) from 500 to 5000 with a step size of 500; and for CIFAR-100, the number is increased from 100 to 500 with a step size of 50. We then proceed to train two modern neural networks—ResNet-56 (He et al. 2016) and WideResNet-28-10 (Zagoruyko & Komodakis 2016)—on them for 50
Figure 4: Upper bound $\varrho_{\text{bound}}$ and true generalization gap as a function of the number of training examples.

Figure 5: Negative correlation between the upper bound $\varrho_{\text{bound}}$ and the ratio of learning rate to mini-batch size.

As depicted in Figure 4 (and Supplementary Material, Figure 2), the generalization gap (test loss - training loss) indeed decreases as more training examples are used and our bound $\varrho_{\text{bound}}$ correctly captures this trend. More importantly, we observe that $\varrho_{\text{bound}}$ is non-vacuous and almost can recover the generalization gap when the full training set is used.
4.3 Effects of Learning Rate and Mini-batch Size

Another issue that hinders previous generalization bounds from wide usage is that they often anti-correlate with the generalization error when changing the commonly used training hyperparameters (Jiang et al. 2020). In this part, we aim to probe the effects of learning rate and mini-batch size, which typically dominate the generalization performance of DNNs. To this end, we varied the learning rate from 0.02 to 0.1 with a step size of 0.02 and simultaneously doubled the mini-batch size from 64 to 1024.

As shown in Figure 5 (and Supplementary Material, Figure 3), we can observe that the upper bound $\varrho_{\text{bound}}$ indeed decreases with the ratio of the learning rate to the mini-batch size. These results align with the observation that a larger ratio of learning rate to mini-batch size usually leads to a better generalization (Jastrzebski et al. 2018, He et al. 2019).

4.4 Results on ImageNet-1K

In this section, we continue to investigate how the proposed bound $\varrho_{\text{bound}}$ evolves with the training epoch. Particularly, we evaluate it on the large-scale ImageNet-1K dataset. For this purpose, we trained on two popular neural networks—ResNet-18 and ViT-S-32 (Dosovitskiy et al. 2021)—with basic data augmentation, namely, resizing and cropping images to 224-pixel resolution and then normalizing them. For ResNet-18, we trained it for 100 epochs with a mini-batch size of 256 and optimization uses SGD with an initial learning rate of 0.1 and a weight decay of 1.0e-4. For ViT-S-32, we trained it for 300 epochs with a mini-batch size of 1024 and the optimizer is AdamW with an initial learning rate of 3.0e-3 and a weight decay of 0.1. For both models, a cosine schedule is used to adjust the learning rate.
Figure 6: Predicted accuracy as a function of the training epoch on the ImageNet-1K validation set. The predicted accuracy on the validation set is obtained by first estimating the validation loss (i.e. $\varrho_{\text{bound}} + \text{training loss}$) and then retrieving the closest accuracy from the training curve (i.e. pairs of training loss and training accuracy).

As shown in Figure 6, we can observe that the predicted accuracy on the validation set monotonically increases as a function of the training epoch, which is consistent with the true validation accuracy. More importantly, our approach is able to produce non-vacuous predictions at the end of training on the validation accuracy (76.5% of predicted accuracy vs. 65.2% of validation accuracy for ViT-S-32 and 57.8% of predicted accuracy vs. 67.6% of validation accuracy for ResNet-18). To the best of our knowledge, these results are the tightest generalization bounds on ImageNet-1K up to date.

4.5 Comparison with Existing Estimators

In this section, we quantitatively compare the Hausdorff dimension $\dim H W$ estimated according to Equation (3) against other methods such as through the upper Blumenthal-Getoor index (Simsekli et al. 2020) and the persistent homology dimension (Birdal et al. 2021, Dupuis et al. 2023). Theoretically, these measures would be smaller if the corresponding neural network enjoys a better generalization performance. For convenience, we
still probe how they change with the number of training examples.

As illustrated in Figure 7 (and Supplementary Material, Figure 4), the persistent homology dimension increases with the training set size, which is undesirable because training with more examples generally yields better generalization. Meanwhile, the upper Blumenthal-Getoor index stays around 1.0 and fails to convey any information about the training set size. By contrast, our method suggests that the Hausdorff dimension decreases with the number of training examples, which is more consistent with the true generalization error.

Figure 7: Comparison between different Hausdorff dimension estimators.
5 Conclusion

In this study, we developed a non-vacuous and tractable generalization bound for SGD from the perspective of fractal geometry, which is different from the classical generalization bounds. Empirical results further demonstrated its efficacy by altering the training set size and the ratio of the learning rate to the mini-batch size. Following this line, it is natural to extend our results to encompass the adaptive optimizers such as Adam and RMSprop, which we leave for future study.

Supplementary Materials and Conflict of Interest

The supplement includes the additional figures and the source code to reproduce all experimental results. In addition, the authors report there are no competing interests to declare.

References

Advani, M. S., Saxe, A. M. & Sompolinsky, H. (2020), ‘High-dimensional dynamics of generalization error in neural networks’, Neural Networks 132, 428–446.

Ainsworth, S., Hayase, J. & Srinivasa, S. (2023), Git re-basin: Merging models modulo permutation symmetries, in ‘ICLR’, pp. 1–29.

Amir, I., Koren, T. & Livni, R. (2021), SGD generalizes better than GD (and regularization doesn’t help), in ‘COLT’, pp. 63–92.

Arora, S., Du, S. S., Hu, W., Li, Z. & Wang, R. (2019), Fine-grained analysis of optimization
and generalization for overparameterized two-layer neural networks, in ‘ICML’, pp. 322–332.

Arora, S., Ge, R., Neyshabur, B. & Zhang, Y. (2018), Stronger generalization bounds for deep nets via a compression approach, in ‘ICML’, pp. 254–263.

Barrett, D. G. T. & Dherin, B. (2021), Implicit gradient regularization, in ‘ICLR’, pp. 1–25.

Bartlett, P. L., Foster, D. J. & Telgarsky, M. (2017), Spectrally-normalized margin bounds for neural networks, in ‘NeurIPS’, pp. 6240–6249.

Bartlett, P. L. & Mendelson, S. (2002), ‘Rademacher and Gaussian complexities: Risk bounds and structural results’, Journal of Machine Learning Research 3, 463–482.

Belkin, M., Hsu, D., Ma, S. & Mandal, S. (2019), ‘Reconciling modern machine-learning practice and the classical bias–variance trade-off’, Proceedings of the National Academy of Sciences 116(32), 15849–15854.

Bengio, Y., Lecun, Y. & Hinton, G. (2021), ‘Deep learning for AI’, Communications of the ACM 64(7), 58–65.

Birdal, T., Lou, A., Guibas, L. J. & Simsekli, U. (2021), Intrinsic dimension, persistent homology and generalization in neural networks, in ‘NeurIPS’, pp. 6776–6789.

Camuto, A., Deligiannidis, G., Erdogdu, M. A., Gurbuzbalaban, M., Simsekli, U. & Zhu, L. (2021), Fractal structure and generalization properties of stochastic optimization algorithms, in ‘NeurIPS’, pp. 18774–18788.

Chaudhari, P. & Soatto, S. (2018), Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks, in ‘ICLR’, pp. 1–20.
Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K. & Fei-Fei, L. (2009), Imagenet: A large-scale hierarchical image database, in ‘CVPR’, pp. 248–255.

Dinh, L., Pascanu, R., Bengio, S. & Bengio, Y. (2017), Sharp minima can generalize for deep nets, in ‘ICML’, pp. 1019–1028.

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S. et al. (2021), An image is worth 16x16 words: Transformers for image recognition at scale, in ‘ICLR’, pp. 1–22.

Dupuis, B., Deligiannidis, G. & Simsekli, U. (2023), Generalization bounds using data-dependent fractal dimensions, in ‘ICML’, pp. 8922–8968.

Dupuis, B. & Simşekli, U. (2024), Generalization bounds for heavy-tailed SDEs through the fractional Fokker-Planck equation, in ‘ICML’, pp. 1–13.

Dziugaite, G. K., Drouin, A., Neal, B., Rajkumar, N., Caballero, E., Wang, L., Mitliagkas, I. & Roy, D. M. (2020), In search of robust measures of generalization, in ‘NeurIPS’, pp. 1–28.

Dziugaite, G. K. & Roy, D. M. (2017), Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data, in ‘UAI’, pp. 1–14.

Embrechts, P. (2009), Selfsimilar Processes, Princeton University Press.

Entezari, R., Sedghi, H., Saukh, O. & Neyshabur, B. (2022), The role of permutation invariance in linear mode connectivity of neural networks, in ‘ICLR’, pp. 1–27.

Falconer, K. (2004), Fractal Geometry: Mathematical Foundations and Applications, John Wiley & Sons.
Gess, B., Kassing, S. & Konarovskyi, V. (2024), ‘Stochastic modified flows, mean-field limits and dynamics of stochastic gradient descent’, Journal of Machine Learning Research 25(30), 1–27.

Golowich, N., Rakhlin, A. & Shamir, O. (2018), Size-independent sample complexity of neural networks, in ‘COLT’, pp. 297–299.

Hardt, M., Recht, B. & Singer, Y. (2016), Train faster, generalize better: Stability of stochastic gradient descent, in ‘ICML’, pp. 1225–1234.

He, F., Liu, T. & Tao, D. (2019), Control batch size and learning rate to generalize well: Theoretical and empirical evidence, in ‘NeurIPS’, pp. 1143–1152.

He, K., Zhang, X., Ren, S. & Sun, J. (2016), Deep residual learning for image recognition, in ‘CVPR’, pp. 770–778.

Hochreiter, S. & Schmidhuber, J. (1997), ‘Flat minima’, Neural Computation 9(1), 1–42.

Hu, W., Li, C. J., Li, L. & Liu, J.-G. (2019), ‘On the diffusion approximation of nonconvex stochastic gradient descent’, Annals of Mathematical Sciences and Applications 4(1), 3–32.

Ioffe, S. & Szegedy, C. (2015), Batch normalization: Accelerating deep network training by reducing internal covariate shift, in ‘ICML’, pp. 448–456.

Jastrzebski, S., Kenton, Z., Arpit, D., Ballas, N., Fischer, A., Bengio, Y. & Storkey, A. (2018), Three factors influencing minima in SGD, in ‘ICANN’, pp. 1–14.

Jiang, Y., Neyshabur, B., Mobahi, H., Krishnan, D. & Bengio, S. (2020), Fantastic generalization measures and where to find them, in ‘ICLR’, pp. 1–33.
Kaplan, J., McCandlish, S., Henighan, T., Brown, T. B., Chess, B., Child, R., Gray, S., Radford, A., Wu, J. & Amodei, D. (2020), ‘Scaling laws for neural language models’, *ArXiv preprint abs/2001.08361.*

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M. & Tang, P. T. P. (2017), On large-batch training for deep learning: Generalization gap and sharp minima, *in ‘ICLR’*, pp. 1–16.

Klingenhöfer, F. & Zähle, M. (1999), ‘Ordinary differential equations with fractal noise’, *Proceedings of the American Mathematical Society* **127**(4), 1021–1028.

Krizhevsky, A., Hinton, G. et al. (2009), Learning multiple layers of features from tiny images, Technical report, University of Toronto.

Krogh, A. & Hertz, J. A. (1992), A simple weight decay can improve generalization, *in ‘NeurIPS’*, pp. 950–957.

Li, Q., Tai, C. & E, W. (2017), Stochastic modified equations and adaptive stochastic gradient algorithms, *in ‘ICML’*, pp. 2101–2110.

Li, Q., Tai, C. & Weinan, E. (2019), ‘Stochastic modified equations and dynamics of stochastic gradient algorithms i: Mathematical foundations’, *Journal of Machine Learning Research* **20**, 40–1.

Li, Z., Su, W. J. & Sejdinovic, D. (2023), ‘Benign overfitting and noisy features’, *Journal of the American Statistical Association* **118**(544), 2876–2888.

Lotfi, S., Finzi, M., Kapoor, S., Potapczynski, A., Goldblum, M. & Wilson, A. G. (2022), PAC-Bayes compression bounds so tight that they can explain generalization, *in ‘NeurIPS’*, pp. 31459–31473.
Lou, S. & Ouyang, C. (2016), ‘Fractal dimensions of rough differential equations driven by fractional Brownian motions’, *Stochastic Processes and Their Applications* **126**(8), 2410–2429.

Lucchi, A., Proske, F., Orvieto, A., Bach, F. & Kersting, H. (2022), On the theoretical properties of noise correlation in stochastic optimization, *in ‘NeurIPS’*, pp. 14261–14273.

Lyons, T. & Qian, Z. (2002), *System Control and Rough Paths*, Oxford University Press.

Mandelbrot, B. B. & Van Ness, J. W. (1968), ‘Fractional Brownian motions, fractional noises and applications’, *SIAM Review* **10**(4), 422–437.

Mandt, S., Hoffman, M. D. & Blei, D. M. (2017), ‘Stochastic gradient descent as approximate Bayesian inference’, *Journal of Machine Learning Research* **18**, 1–35.

Mattila, P. (1999), *Geometry of Sets and Measures in Euclidean Spaces: Fractals and Rectifiability*, Cambridge University Press.

Mishura, Y. & Shevchenko, G. (2008), ‘The rate of convergence for Euler approximations of solutions of stochastic differential equations driven by fractional Brownian motion’, *Stochastics* **80**(5), 489–511.

Mohri, M., Rostamizadeh, A. & Talwalkar, A. (2018), *Foundations of Machine Learning*, MIT Press.

Nagarajan, V. & Kolter, J. Z. (2019), Uniform convergence may be unable to explain generalization in deep learning, *in ‘NeurIPS’*, pp. 11611–11622.

Neyshabur, B., Bhojanapalli, S. & Srebro, N. (2018), A PAC-Bayesian approach to spectrally-normalized margin bounds for neural networks, *in ‘ICLR’*, pp. 1–9.
Neyshabur, B., Tomioka, R. & Srebro, N. (2015), Norm-based capacity control in neural networks, *in* ‘COLT’, pp. 1376–1401.

Nguyen, T. H., Simsekli, U., Gürbüzbalaban, M. & Richard, G. (2019), First exit time analysis of stochastic gradient descent under heavy-tailed gradient noise, *in* ‘NeurIPS’, pp. 273–283.

Nguyen, V.-A., Vuong, T.-L., Phan, H., Do, T.-T., Phung, D. & Le, T. (2024), Flat seeking Bayesian neural networks, *in* ‘NeurIPS’, pp. 1–11.

Pérez-Ortiz, M., Rivasplata, O., Shawe-Taylor, J. & Szepesvári, C. (2021), ‘Tighter risk certificates for neural networks’, *Journal of Machine Learning Research* **22**, 1–40.

Sachs, S., van Erven, T., Hodgkinson, L., Khanna, R. & Şimşekli, U. (2023), Generalization guarantees via algorithm-dependent Rademacher complexity, *in* ‘COLT’, PMLR, pp. 4863–4880.

Sclocchi, A. & Wyart, M. (2024), ‘On the different regimes of stochastic gradient descent’, *Proceedings of the National Academy of Sciences* **121**(9), e2316301121.

Shalev-Shwartz, S. & Ben-David, S. (2014), *Understanding Machine Learning: From Theory to Algorithms*, Cambridge University Press.

Simsekli, U., Sagun, L. & Gürbüzbalaban, M. (2019), A tail-index analysis of stochastic gradient noise in deep neural networks, *in* ‘ICML’, pp. 5827–5837.

Simsekli, U., Sener, O., Deligiannidis, G. & Erdogdu, M. A. (2020), Hausdorff dimension, heavy tails, and generalization in neural networks, *in* ‘NeurIPS’, pp. 1–14.

Smith, S. L., Dherin, B., Barrett, D. G. T. & De, S. (2021), On the origin of implicit regularization in stochastic gradient descent, *in* ‘ICLR’, pp. 1–14.
Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. (2014), ‘Dropout: a simple way to prevent neural networks from overfitting’, *Journal of Machine Learning Research* **15**(1), 1929–1958.

Tan, C., Zhang, J. & Liu, J. (2023), ‘Understanding short-range memory effects in deep neural networks’, *IEEE Transactions on Neural Networks and Learning Systems* pp. 1–14.

Wen, K., Li, Z. & Ma, T. (2023), Sharpness minimization algorithms do not only minimize sharpness to achieve better generalization, in ‘NeurIPS’, pp. 1–12.

Wu, L. & Su, W. J. (2023), The implicit regularization of dynamical stability in stochastic gradient descent, in ‘ICML’, pp. 37656–37684.

Xiao, Y. (1995), ‘Dimension results for Gaussian vector fields and index-α stable fields’, *The Annals of Probability* pp. 273–291.

Xie, Z., Sato, I. & Sugiyama, M. (2021), A diffusion theory for deep learning dynamics: Stochastic gradient descent escapes from sharp minima exponentially fast, in ‘ICLR’, pp. 1–28.

Xu, A. & Raginsky, M. (2017), Information-theoretic analysis of generalization capability of learning algorithms, in ‘NeurIPS’, pp. 1–15.

Zagoruyko, S. & Komodakis, N. (2016), ‘Wide residual networks’, *arXiv preprint arXiv:1605.07146*.

Zhang, C., Bengio, S., Hardt, M., Recht, B. & Vinyals, O. (2017), Understanding deep learning requires rethinking generalization, in ‘ICLR’, pp. 1–15.
Zhang, J., Karimireddy, S. P., Veit, A., Kim, S., Reddi, S., Kumar, S. & Sra, S. (2020), Why are adaptive methods good for attention models?, in 'NeurIPS', pp. 1–23.

Zhou, W., Veitch, V., Austern, M., Adams, R. P. & Orbanz, P. (2019), Non-vacuous generalization bounds at the Imagenet scale: a PAC-Bayesian compression approach, in 'ICLR', pp. 1–16.

Zhu, Z., Wu, J., Yu, B., Wu, L. & Ma, J. (2019), The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects, in 'ICML', pp. 7654–7663.