Generalization Bounds for Gradient Methods via Discrete and Continuous Prior

Anonymous Author(s)
Affiliation
Address
email

Abstract

Proving algorithm-dependent generalization error bounds for gradient-type optimization methods has attracted significant attention recently in learning theory. However, most existing trajectory-based analyses require either restrictive assumptions on the learning rate (e.g., fast decreasing learning rate), or continuous injected noise (such as the Gaussian noise in Langevin dynamics). In this paper, we introduce a new discrete data-dependent prior to the PAC-Bayesian framework, and prove a high probability generalization bound of order $O(\frac{1}{n} \cdot \sum_{t=1}^{T} (\gamma_t/\varepsilon_t)^2 \left\|\mathbf{g}_t\right\|^2)$ for Floored GD (i.e. a version of gradient descent with precision level ε_t), where n is the number of training samples, γ_t is the learning rate at step t, $\mathbf{g}_t$ is roughly the difference of the gradient computed using all samples and that using only prior samples. $\|\mathbf{g}_t\|$ is upper bounded by and and typical much smaller than the gradient norm $\|\nabla f(W_t)\|$. We remark that our bound holds for nonconvex and nonsmooth scenarios. Moreover, our theoretical results provide numerically favorable upper bounds of testing errors (e.g., 0.037 on MNIST). Using similar technique, we can also obtain new generalization bounds for certain variant of SGD. Furthermore, we study the generalization bounds for gradient Langevin Dynamics (GLD). Using the same framework with a carefully constructed continuous prior, we show a new high probability generalization bound of order $O(\frac{1}{n} + \frac{L^2}{n^2} \sum_{t=1}^{T} (\gamma_t/\sigma_t)^2)$ for GLD. The new $1/n^2$ rate is due to the concentration of the difference between the gradient of training samples and that of the prior.

1 Introduction

Bounding generalization error of learning algorithms is one of the most central problems in machine learning theory. Formally, for a supervised learning problem, the generalization error is defined as the testing error (or population error) minus the training error (or empirical error). In particular, we denote $\mathcal{R}(w, (x, y)) := \mathbb{1}[h_w(x) \neq y]$ as the error of a single data point (x, y), where $h_w(x)$ is the output of a model with parameter $w \in \mathbb{R}^d$. Suppose S is the set of training data, each i.i.d. sampled from the population distribution $\mathcal{D}$, and we use $\mathcal{R}(w, S) := \frac{1}{|S|} \sum_{z \in S} \mathcal{R}(w, z)$ and $\mathcal{R}(w, \mathcal{D}) := \mathbb{E}_{z \sim D}[\mathcal{R}(w, z)]$ to denote the training error and the testing error, respectively. The generalization error of w is formally defined as $\mathrm{err}_{\mathrm{gen}}(w) = \mathcal{R}(w, \mathcal{D}) - \mathcal{R}(w, S)$.

Proving tighter generalization bounds for general nonconvex learning and in particular deep learning has attracted significant attention recently. While the classical learning theory (uniform convergence theory) which bounds the generalization error by various complexity measures (e.g., the VC-dimension and Rademacher complexity) of the hypothesis class has been successful in several classical convex learning models, however, they become vacuous and hence fail to explain the success of modern nonconvex over-parametrized neural networks (i.e., the number of parameters significantly exceeds the number of training data) (see e.g., Zhang et al. [2017a], Nagarajan and

Submitted to 36th Conference on Neural Information Processing Systems (NeurIPS 2022). Do not distribute.

Kolter [2019]). Recently, learning theorists have tried to understand and explain the generalization ability of deep learning from several other perspectives, such as margin theory [Bartlett et al., 2017, Wei et al., 2019], algorithmic stability [Hardt et al., 2016, Mou et al., 2018, Bousquet et al., 2020], PAC-bayeisan [London, 2017, Bartlett et al., 2017, Neyshabur et al., 2018, Zhou et al., 2019, Yang et al., 2019], neural tangent kernel [Arora et al., 2019, Cao and Gu, 2019], information theory [Pensia et al., 2018, Negrea et al., 2019], model compression [Arora et al., 2018], differential privacy [Oneto et al., 2017] and so on.

In this paper, we aim to obtain tighter generalization error bounds that depend on both the training data and the optimization algorithms (a.k.a. gradient-type methods) for general nonconvex learning problems. In particular, we prove algorithm-dependent generalization bounds for close variants of the most commonly used algorithms such as gradient descent (GD), stochastic gradient descent (SGD) and stochastic gradient Langevin dynamics (SGLD). Our proof is based on the classic Catoni's PAC-Bayesian framework [Catoni, 2007] and also has a flavor of algorithmic stability Bousquet and Elisseeff [2002]. Several prior works have obtained generalization bounds for SGD and SGLD by analyzing trajectory through either the PAC-Bayesian or the algorithmic stability framework Mou et al. [2018], Li et al. [2020] (or closely related information theoretic arguments). However, most existing results which are based on analysing the optimization trajectories require either restrictive assumptions on the learning rate (e.g., in Hardt et al. [2016], the learning rate η_t scales with $1/t$), or continuous noise (such as the Gaussian noise in Langevin dynamics) in order to bound the stability or the KL-divergence. In this paper, we resolve the above restrictions by combining the PAC-Bayesian framework with a few simple (yet effective) ideas, so that we can obtain new high probability and non-vacuous generalization bounds for several gradient-based optimization methods with either discrete or continuous noise (in particular certain variants of GD and SGD, either being deterministic or with discrete noise, which cannot be handled by existing techniques).

1.1 Prior work

We first briefly mention some recent work on bounding the generalization error of gradient-based methods. Hardt et al. [2016] first studied the uniform stability (hence the generalization) of stochastic gradient descent (SGD) for both convex and non-convex functions. Their results for non-convex functions requires that the learning rate η_t scales with $1/t$. Their work motivates a long line of subsequent work on generalization error bounds of gradient-based optimization methods: Kuzborskij and Lampert [2018], London [2016], Chaudhari et al. [2019], Raginsky et al. [2017], Mou et al. [2018], Chen et al. [2018].

PAC-Bayesian bounds. The PAC-Bayesian framework [McAllester, 1999] is a powerful method for proving high probability generalization bound [Bartlett et al., 2017, Zhou et al., 2019, Mou et al., 2018]. Roughly speaking, it bounds the generalization error by the KL divergence $\mathrm{KL}\left(Q \,\middle|\middle|\, P\right)$, where Q is the distribution of learned output and P is a prior distribution which is typically independent of dataset S. In this framework, bounding $\mathrm{KL}\left(Q \,\middle|\middle|\, P\right)$ is the most crucial part for obtaining tighter PAC-Bayesian bounds. In order to bound the KL divergence, both the prior P and posterior Q are required to be continuous distributions (typically Gaussians so that KL can be computed in closed form). Hence, most prior work either consider gradient methods with continuous noise (such as gradient Langevin Dynamics) [Mou et al., 2018], or add a Gaussian noise to the final parameter at the end [Neyshabur et al., 2018, Zhou et al., 2019] (so Q is a Gaussian distribution). We also note that designing effective prior P can be also very important. For example, Lever et al. [2013] proposed to use the population distribution to compute the prior. In fact, the prior can even partially depend on the training data [Parrado-Hernández et al., 2012, Negrea et al., 2019], and our Theorem 4.1 is partially inspired by this idea.

1.2 Our contributions

First, we provide high probability generalization bounds for *discrete* gradient methods. In particular, we study the generalization of Floored Gradient Descent (FGD), which is a variant of GD, and Floored Gradient Descent (FSGD), a variant of SGD. We obtain our bound by an interesting construction of discrete priors. Secondly, we consider well studied gradient methods with continuous noise, (stochastic) gradient Langvin dynamics (GLD and SGLD). We show sharper generalization bounds by carefully bounding the concentration of the sample gradients. Now, we summarize our results.

FGD and FSGD. We first study a close variant of GD called FGD (Algorithm 1). The update rule of FGD is defined as follows:

$$W_t \leftarrow W_{t-1} - \gamma_t \nabla f(W_{t-1}, S_J) - \varepsilon_t \mathrm{floor}\left(\gamma_t \mathbf{g}_t / \varepsilon_t\right), \tag{FGD}$$

where S_J is a subset of training dataset S with size m, J is its index-set chosen before the training, γ_t is the learning rate, ε_t is the precision level, and $\mathbf{g}_t := \nabla f(W_{t-1}, S) - \nabla f(W_{t-1}, S_J)$ is the gradient difference. The flooring operation on a real number is defined by $\mathrm{floor}(x) := \mathrm{sign}(x)\lfloor |x| \rfloor$. It can viewed as gradient descent with given precision limits ε_t. We can see if we ignore the floor operatio or let ε_t be very small, FGD reduces to GD. Empirically, its behavior and optimization capability is very close to GD (See Figure 2a and 2b).

By constructing a discrete data-dependent prior and incorporate it into Catoni's PAC-Bayesian framework, we prove that the following bound (Theorem 5.2) holds with high probability:

$$\mathcal{R}(W_T, \mathcal{D}) \leq c_0 \mathcal{R}(W_T, S_I) + O\left(\frac{1}{n-m} + \frac{\ln(dT)}{n-m} \sum_{t=1}^{T} \frac{\gamma_t^2}{\varepsilon_t^2} \|\mathbf{g}_t\|^2\right),$$

where d is the dimension of parameter space. Now we make a few remarks about our results.

1. Our result holds for nonconvex and nonsmooth learning problems (replacing the gradients with subgradients for nonsmooth cases). There is no additional requirement on the learning rate γ_t.
2. The gradient difference $\mathbf{g}_t$ is typical much smaller than the worst case gradient norm. It usually decreases when $m = |J|$ grows (see Figure 1c). Thus, it enables us to obtain non-vacuous bound on classical datasets. Specifically, our theoretical test error upper bound on MNIST and CIFAR10 are $\mathbf{0.0369}$ and 0.8589, respectively.
3. The prior P is a carefully constructed discrete random processes, so that we can bound the KL between P and the deterministic process of FGD. We hope it may inspire future research on tackling deterministic optimization algorithms or discrete noise.
4. The result can be extended to a variant of SGD, called FSGD (see Theorem 5.3 for details).

GLD, SGLD and CLD. We also provide new a generalization bound for Gradient Langevin Dynamics (GLD) whose update rule is defined below.

$$W_t \leftarrow W_{t-1} + \gamma_t \nabla f(W_{t-1}, S) + \sigma_t \mathcal{N}(0, I_d). \tag{GLD}$$

In this paper, we show that the following generalization bound (Theorem 6.2) holds with high probability over the randomness of $S \sim \mathcal{D}^n$ and random subset J:

$$\mathcal{R}(W_T, \mathcal{D}) \leq c_0 \mathcal{R}(W_T, S_{[n]\setminus J}) + O\left(\frac{1}{n-m} + \frac{1}{(n-m)m} \mathbb{E}\left[\sum_{t=1}^{T} \frac{\gamma_t^2}{\sigma_t^2} L(W_{t-1})^2\right]\right),$$

where $L(W_{t-1}) := \max_{z \in S} \|f(W_{t-1}, z)\|$ is the longest gradient norm of any training sample in S at step t, m is the size of J. Since W_T is independent of the prior index J, the first term $\mathcal{R}(W_T, S_{[n]\setminus J})$ is upper bounded by $\mathcal{R}(W_T, S) + O(\frac{1}{\sqrt{n-m}})$ with high probability. By setting $m = n/2$, our generalization bound has an $O(\frac{1}{\sqrt{n}} + \frac{1}{n} + \frac{T}{n^2})$ rate. We also prove a high probability generalization bound for SGLD (see Theorem 6.3):

$$\mathcal{R}(W_T, \mathcal{D}) \leq c_0 \mathcal{R}(W_T, S_{[n]\setminus J}) + O\left(\frac{1}{n-m} + \frac{1}{n-m}\left(\frac{1}{b} + \frac{1}{m}\right) \sum_{t=1}^{T} \frac{\gamma_t^2}{\sigma_t^2} L(W_{t-1})^2\right).$$

We compare our bounds with existing GLD/SGLD generalization bounds in prior work. For GLD, Mou et al. [2018] provide a generalization bound in expectation based on the uniform stability framework, which is of rate $O(\frac{L\sqrt{T}}{n})$, where L is the global Lipschitz constant (ignoring the factors depending on γ_t and σ_t). Their bound can be tighten to $O(\frac{1}{n}\sqrt{\sum_{t=1}^{T} L_t^2})$ where L_t is the gradient norm at time t (which is always less than L) [Li et al., 2020]. Their bounds can be converted to high probability bound with an additional factor $O(1/\sqrt{n})$ using the technique developed in [Feldman and Vondrak, 2019]. Bounds of similar orders have been also obtained through information theory [Wang et al., 2021, Haghifam et al., 2020] and differential privacy [Wu et al., 2021]. Note that the main term in our bound is of order $O(\frac{1}{n^2}\sum_{t=1}^{T} L_t^2)$, which is quadratically better than theirs if the bound is in $(0, 1)$. For SGLD, Mou et al. [2018] and Li et al. [2020] obtained similar bounds, but requires the assumption that the learning rate should be of order $O(1/L)$. Our bound for SGLD does

not require such assumption and is more favorable for large minibatch size b. For small value of b (say $b = O(1)$, our bound can be worse.

Another closely related work is Negrea et al. [2019], which also use a data-dependent prior. They present an in-expectation bound based on information theory. They introduce a quantity called "incoherence" ξ_t that is defined somewhat similar to $\|\mathbf{g}_t\|$ (it is also the norm of the different between two gradients defined by two different subsets of samples). They obtained an $O(\sqrt{\frac{1}{n-m}\sum_{t=1}^{T}\frac{\gamma_t^2}{\sigma_t^2}\mathbb{E}[\|\xi_t\|^2]})$ bound for SGLD. By taking expectation, it can be further bounded by $O(\sqrt{\frac{1}{n}\sum_{t=1}^{T}(\frac{1}{b}+\frac{(n-m)}{nm})\frac{\gamma_t^2}{\sigma_t^2}V_t})$, where V_t is a quantity about the same size as the variance of training gradients. However, directly plugging their worst case $O(\frac{(n-m)^2}{n^2}\sum_{t=1}^{T}\frac{L^2\gamma_t^2}{\sigma_t^2})$ bound for $\mathrm{KL}\left(Q \,\middle|\middle|\, P(S_J)\right)$ into Catoni's PAC-Bayesian bound, one can only obtain an $O(\frac{1}{n-m}+\frac{(n-m)}{n^2}\sum_{t=1}^{T}\frac{\gamma_t^2}{\sigma_t^2}L^2)$ high probability bound. To make the 2nd term in their bound have the same $O(1/n^2)$ rate as ours, one needs to set $n-m=O(1)$ which would result in a large first term $\frac{1}{n-m}=\Omega(1)$. Moreover, that our construction of data-dependent prior $P(S_J)$ is also very different from theirs. Their idea is using the gradients in S_J to cancel out the gradients in S while ours is based on the property that the mean gradient on S_J is concentrated around the mean gradient of the whole dataset S.

Mou et al. [2018] also obtain a high probability PAC-Bayesian bound of rate $O(\sqrt{\frac{1}{n}\sum_{t=1}^{T}e^{-r_t}L_t^2})$ if there is an ℓ_2-regularization in the loss (ignoring other factors depending on γ_t and σ_t). Here $e^{-r_t}<1$ is a decay factor depending on the regularization coefficient. There is a similar decay factor in Wang et al. [2021]'s bound (the fact comes from strong data processing inequalities). Note that without ℓ_2-regularization, there is no such decay factor, and Mou et al. [2018]'s bound becomes $O(\sqrt{T/n})$, which is looser than ours. From technical perspective, they use Fokker Planck equation to track the time derivative of KL and Logarithmic Sobolev inequality to related KL with Fisher information. We also use these tools for our generalization bound of Continuous Langevin dynamics (CLD) (see Appendix E), but the general proof idea is very different.

2 Other Related Work

Stochastic Langevin Dynamics Stochastic Langevin dynamics is popular sampling and optimization method used in machine learning Welling and Teh [2011]. Zhang et al. [2017b], Chen et al. [2020] show a polynomial hitting time (hitting a stationary point) of SGLD in general non-convex setting. Raginsky et al. [2017] study the generalization and excess risk of SGLD in nonconvex settings and their bound depends polynomially on a certain spectral gap parameter which may be exponential in the dimension. Continuous Langevin dynamics (SDE) with various noise structure has also been used extensively as approximations of SGD (see e.g., [Li et al., 2017, 2021]).

Nonvacuous PAC-Bayesian Generalization Bounds. Dziugaite and Roy [2017] first present a non-vacuous PAC-Bayesian generalization bound on MNIST (0.20 for a 3 layer MLP). They use a somewhat different training algorithm that explicitly optimizes the PAC-Bayesian bound and the output distribution is a multivariate normal distribution $\mathcal{N}(w, \mathrm{diag}(s))$. To facilitate computing the closed form of KL, they chose a zero-mean Gaussian distribution as the prior. Zhou et al. [2018] obtain non-vacuous bound for ImageNet via a quite different approach. Their method does not require any continuous noise injected but assumes that the network can be significantly compressed (so that the prior distribution is supported over the set of discrete parameters with finite precisions). To our best knowledge, it is the only work that use a discrete prior for proving generalization bounds of deep neural networks. Our result for Floored gradient descent has a similar flavor in a high level, that is the optimization method has a finite precision. However, our results do not need any assumption on compressibility of the model and can be applied to other nonconvex learning problems.

Generalization bound via Information theory. Raginsky et al. [2017] first shows that the expected generalization error $\mathbb{E}_{S\sim\mathcal{D}^n}[\mathcal{R}(W,\mathcal{D})-\mathcal{R}(W,S)]$ is bounded by $\sqrt{2I(S;W)/n}$, where $I(S;W) := \mathrm{KL}\left(P(S,W) \,\middle|\middle|\, P(S)\otimes P(W)\right)$ is the mutual information. This work motivates many subsequent studies [Pensia et al., 2018, Negrea et al., 2019, Bu et al., 2020, Wang et al., 2021]. The main goal in this line of work is to obtain a tight bound on the mutual information $I(S;W)$. This is again reduced

to bound the KL divergence and thus typically requires continuous injected noise Wang et al. [2021], Negrea et al. [2019].

3 Preliminaries

Notations. We assume the training dataset $S = (z_1, .., z_n)$ is sampled from $\mathcal{D}^n$, where $\mathcal{D}$ is the population distribution. The model parameter w is in $\mathbb{R}^d$. The risk function $\mathcal{R} : \mathbb{R}^d \times \Omega \to [0, 1]$ measures the error of a model on a datapoint. The loss function $f : \mathbb{R}^d \times \Omega \to \mathbb{R}$ is used for training models. The empirical risk is $\mathcal{R}(w, S) = \frac{1}{|S|}\sum_{z\in S}\mathcal{R}(w, z)$ and population risk is $\mathcal{R}(w, \mathcal{D}) = \mathbb{E}_{z\sim\mathcal{D}}[\mathcal{R}(w, z)]$. Similarly, we can define the empirical loss $f(w, S)$ and population loss $f(w, \mathcal{D})$. For any $J = (j_1, .., j_m)$, we use S_J to denote the sequence $(S_{j_1}, ..., S_{j_m})$. The subsequence $(A_i, A_{i+1}, ..., A_j)$ is denoted by A_i^j. We use (A_1^n, B_1^m) to denote the merged sequence $(A_1, A_2, ..., A_n, B_1, ..., B_m)$. When the elements in sequence J are different from each other, we also use itself to represent the set consisting of all of its elements. We also use a random variable itself to denote its distribution. For example, $\mathbb{E}_{x\sim X}[f(x)]$ is equivalent to $\mathbb{E}_{x\sim P_X}[f(x)]$, and $\mathrm{KL}\left(X \,\middle|\middle|\, Y\right)$ means $\mathrm{KL}\left(P_X \,\middle|\middle|\, P_Y\right)$. For a random variable W, we define $\mathcal{R}(W, S) = \mathbb{E}_{w\sim W}[\mathcal{R}(w, S)]$ and $\mathcal{R}(W, \mathcal{D}) = \mathbb{E}_{w\sim W}[\mathcal{R}(w, Q)]$. The index set $\{1, 2, ..., n\}$ is denoted by $[n]$.

KL-divergence. Let P and Q be two probability distributions. The Kullback–Leibler divergence $\mathrm{KL}\left(P \,\middle|\middle|\, Q\right)$ from P to Q is defined only when P is absolute continuous with respect to Q (i.e. for any x, $Q(x) = 0$ implies $P(x) = 0$). In particular, if P and Q are discrete distributions, then $\mathrm{KL}\left(P \,\middle|\middle|\, Q\right) = \sum_x P(x) \ln \frac{P(x)}{Q(x)}$. Otherwise, if P and Q are continuous distributions, it is defined as $\int P(x) \ln \frac{P(x)}{Q(x)}\, \mathrm{d}x$. The following Lemma 3.1 is frequently used in this paper and is a well known property of KL divergence (see Cover [1999, Theorem 2.5.3], Li et al. [2020], Negrea et al. [2019]).

Lemma 3.1 (Chain Rule of KL)**.** *We are given two random sequences $W = (W_0, ..., W_T)$ and $W' = (W'_0, ..., W'_T)$. Then, the following equation holds (given all KLs are well defined):*

$$\mathrm{KL}\left(W \,\middle|\middle|\, W'\right) = \mathrm{KL}\left(W_0 \,\middle|\middle|\, W'_0\right) + \sum_{t=1}^{T} \mathop{\mathbb{E}}_{w\sim W_0^{t-1}} \left[\mathrm{KL}\left(W_t|W_0^{t-1} = w \,\middle|\middle|\, W'_t|{W'}_0^{t-1} = w\right)\right].$$

Here $W_t|W_0^{t-1} = w$ denotes the distribution of W_t conditioning on $W_0^{t-1} = (W_0, \ldots, W_{t-1}) = w$.

PAC-Bayesian. In this paper, we use the PAC-Bayesian bound presented in Catoni [2007] which enjoys a tighter $O(KL/n)$ rate compared to the traditional $O(\sqrt{KL/n})$ bound, but with a slightly larger constant factor on the empirical error. We restate their bound as follows.

Lemma 3.2 (Catoni's Bound)**.** *(see e.g., Lever et al. [2013]) For any prior distribution P independent of S, any $\delta \in (0, 1)$, and any $\eta > 0$, we have the following bound holds w.p. $\geq 1 - \delta$ over $S \sim \mathcal{D}^n$:*

$$\mathop{\mathbb{E}}_{W\sim Q}[\mathcal{R}(W, \mathcal{D})] \leq \eta C_\eta \mathop{\mathbb{E}}_{W\sim Q}[\mathcal{R}(W, \mathcal{D})] + C_\eta \cdot \frac{\mathrm{KL}\left(Q \,\middle|\middle|\, P\right) + \ln(1/\delta)}{n} \quad (\forall Q),$$

where $C_\eta = \frac{1}{1-e^{-\eta}}$ is an absolute constant.

Concentration inequality. We will use the following generalized McDiramid inequality to prove the concentration of cumulative gradient difference in Section 6.

Definition 3.3 (Order-Independent)**.** *A function Φ is said to be order independent if and only if $\Phi(x_1, x_2, ..., x_m) = \Phi(x_{\pi_1}, x_{\pi_2}, ..., x_{\pi_m})$ holds for any input $X = (x_1, x_2, ..., x_m) \in \Omega^m$ and any permutation $\pi \in \mathbb{S}_m$.*

To simplify notation, in the following we use $\Phi(j_1^i, J_{i+1}^m)$ to denote $\Phi(j_1, j_2, ..., j_i, J_{i+1}, ..., J_m)$.

Lemma 3.4. *Suppose $\Phi : [n]^m \to \mathbb{R}^+$ is order-independent and $|\Phi(J) - \Phi(J')| \leq c$ holds for any adjacent J, J' satisfying $|\mathrm{set}(J) \bigcap \mathrm{set}(J')| = m - 1$. Let J be m indices sampled uniformly from $[n]$ without replacement. Then $\Pr_J\left[\Phi(J) - \mathbb{E}_J[\Phi(J)] > \epsilon\right] \leq \exp(\frac{-2\epsilon^2}{mc^2})$.*

4 Data-Dependent PAC-Bayesian Bound

The dominating term in traditional PAC-Bayesian Bound is $\mathrm{KL}\left(Q \,\middle|\middle|\, P\right)/n$, where P is a prior distribution independent of the training dataset S. Intuitively, one can hardly predict the output distribution Q_S without knowing any information from S. Thus the best possible bound for $\mathrm{KL}\left(Q \,\middle|\middle|\, P\right)$ is $\Theta(1)$ w.r.t. n. However, if we are allowed to see m data points from S when constructing our prior, we may produce better prediction on posterior Q_S. The following Theorem 4.1 enables us to use data-dependent prior in PAC-Bayesian bound.

Theorem 4.1 (Data-Dependent PAC-Bayesian)**.** *Suppose J is a random sequence including m indices uniformly sampled from $[n]$ without replacement. For any $\delta \in (0,1)$ and $\eta > 0$, we have w.p. $\geq 1-\delta$ over $S \sim \mathcal{D}^n$ and J:*

$$\mathcal{R}(Q,\mathcal{D}) \leq \eta C_\eta \mathcal{R}(Q,S_I) + C_\eta \cdot \frac{\mathrm{KL}\left(Q \,\middle|\middle|\, P(S_J)\right) + \ln(1/\delta)}{n-m} \quad (\forall Q),$$

where $I = [n] \backslash J$ are indices not in J, $P(S_J)$ is the prior distribution only depending on the information of S_J, and $C_\eta := \frac{1}{1-e^{-\eta}}$ is a constant.

Remarks. Note that the above bound holds for arbitrary Q. Usually, we apply this bound to the posterior Q_S defined by our learning algorithm such as GD or GLD. We remark that for most of our learning algorithms are independent of J, that is, changing J does not change the output Q_S. In this case, by standard Chernoff-Hoeffding inequality, the first term $\mathcal{R}(Q,S_I)$ in the RHS can be bounded by $\mathcal{R}(Q,S) + O(1/\sqrt{n-m})$ with high probability over the randomness of J. However, we point out a subtle point that FGD (Algorithm 1) studied in this paper depends on J. It may be the case that by knowing J, the algorithm extracts more information from S_J but not much from S_I, unintendedly making $\mathcal{R}(Q,S_I)$ an validation error, rather than training error as it should be. However, for the algorithm FGD we study, it does not really discriminate samples from J and from I. From our experiment (Figure 2a and 2d) on CIFAR10, we can see that the S_I error $\mathcal{R}(W_T,S_I)$ is close to the training error $\mathcal{R}(W_T,S)$ and both are significantly smaller than the testing error $\mathcal{R}(W_T,\mathcal{D})$. So $\mathcal{R}(Q,S_I)$ can be considered as a genuine training error in our study of FGD. Finally, note that GLD, SGLD and CLD we study are the standard versions and independent of J, hence $\mathcal{R}(Q,S_I)$ can be replaced by $\mathcal{R}(Q,S) + O(1/\sqrt{n-m})$.

5 Floored Gradient Descent

In this section, we study the generalization of a variant of gradient descent: Floored Gradient Descent (FGD). First we need to define the "flooring" operation which will be used in our algorithm.

Definition 5.1 (Flooring)**.** *For any vector $X \in \mathbb{R}^d$, let $Y = \mathrm{floor}(X)$ defined as:*

$$Y_i = \mathrm{floor}(X_i) = \lfloor X_i \rfloor \text{ if } X_i \geq 0, \quad = -\lfloor -X_i \rfloor \text{ if } X_i < 0, \text{ for all } i \in [d].$$

The Floored Gradient Descent algorithm is defined in Algorithm 1, where $(\gamma_t)_{t\geq 0}$ and $(\varepsilon_t)_{t\geq 0}$ are the step size and precision sequences, respectively.

Algorithm 1: Floored Gradient Descent (FGD)

Input: Training dataset $S = (z_1, .., z_n)$. Index set J. Momentum coefficient α.
Result: Parameter $W_T \in \mathbb{R}^d$.

```
Initialize W_0 ← w_0;
for t : 1 → T do
    g_1 ← γ_t ∇f(W_{t-1}, S);
    g_2 ← γ_t ∇f(W_{t-1}, S_J);
    W_t ← W_{t-1} + α · (W_{t-1} − W_{t-2}) − g_2 − ε_t · floor((g_1 − g_2)/ε_t);
end
```

One can turn off the momentum by setting α to 0. We remark that the FGD is a deterministic algorithm. The following Theorem 5.2 gives the generalization error bound for FGD.

Theorem 5.2. *Suppose J is a random sequence consisting of m indices uniformly sampled from $[n]$ without replacement. Then for any $\delta \in (0,1)$, the FGD (Algorithm 1) satisfies the following generalization bound w.p. at least $1-\delta$ over $S \sim \mathcal{D}^n$ and J:*

$$\mathcal{R}(W_T, \mathcal{D}) \leq \eta C_\eta \mathcal{R}(W_T, S_I) + C_\eta \cdot \frac{\ln(1/\delta)+3}{n-m} + \frac{C_\eta \ln(dT)}{n-m} \sum_{t=1}^{T} \left(\frac{\gamma_t^2}{\varepsilon_t^2} \left\| \mathbf{g}_t \right\|^2 \right),$$

where d is the dimension of parameter space, $I = [n] \backslash J$ are indices that are not in J, $C_\eta := \frac{1}{1-e^{-\eta}}$ is a constant, and $\mathbf{g}_t := \nabla f(W_{t-1}, S) - \nabla f(W_{t-1}, S_J)$ is the gradient difference.

Proof. We'll use Theorem 4.1 to prove our theorem. The key is to construct the prior distribution $P(S_J)$ such that $\mathrm{KL}\left(W_T \,\middle|\middle|\, P(S_J)\right)$ is tractable. We define $P(S_J)$ as the distribution of W'_T obtained by the following update rule ($W'_0 := w_0$):

$$W'_t \leftarrow W'_{t-1} + \alpha \cdot \left(W'_{t-1} - W'_{t-2}\right) - \nabla f(W'_{t-1}, S_J) - \varepsilon_t \cdot \xi_t,$$

where ξ_t is a discrete random variable such that for all $(a_1, .., a_d) \in \mathbb{Z}^d$:

$$\Pr[\xi_t = (a_1, ..., a_d)^\top] := \left(\sum_{i=-\infty}^{\infty} p^{i^2} \right)^{-d} \exp\left(-\sum_{k=1}^{d} \ln(1/p) a_k^2 \right).$$

It is easy to verify that the sum of the probability equals to 1. Recall that $W_0^t = (W_0, ..., W_t)$ is the parameter sequence of FGD (Algorithm 1). Applying the chain rule of KL-divergence, we have:

$$\begin{aligned}
\mathrm{KL}\left(W_T \,\middle|\middle|\, P(S_J)\right) = \mathrm{KL}\left(W_T \,\middle|\middle|\, W'_T\right) &\leq \mathrm{KL}\left(W_0^T \,\middle|\middle|\, {W'}_0^T\right) \\
&= \sum_{t=1}^{T} \mathop{\mathbb{E}}_{w \sim W_0^{t-1}} \left[\mathrm{KL}\left(W_t | W_0^{t-1} = w \,\middle|\middle|\, W'_t | {W'}_0^{t-1} = w \right) \right] \\
&= \sum_{t=1}^{T} \mathrm{KL}\left(W_t | W_0^{t-1} = W_0^{t-1} \,\middle|\middle|\, W'_t | {W'}_0^{t-1} = W_0^{t-1} \right).
\end{aligned} \tag{1}$$

The last equation is because FGD is deterministic. Let $w = W_0^{t-1}$, the distribution of $W_t | W_0^{t-1} = w$ is a point mass on

$$w_{t-1} + \alpha \cdot (w_{t-1} - w_{t-2}) - \gamma_t \nabla f(w_{t-1}, S_J) - \varepsilon_t \cdot \mathrm{floor}\left(\frac{\gamma_t (\nabla f(w_{t-1}, S) - \nabla f(w_{t-1}, S_J))}{\varepsilon_t}\right).$$

Let $a = \mathrm{floor}(\frac{\gamma_t}{\varepsilon_t}(\nabla f(w_{t-1}, S) - \nabla f(w_{t-1}, S_J)))$. By definition of W'_t, we have

$$\begin{aligned}
&\mathrm{KL}\left(W_t | W_0^{t-1} = w \,\middle|\middle|\, W'_t | {W'}_0^{t-1} = w \right) = 1 \cdot \ln\left(1/\Pr\left[\xi_t = a\right]\right) \\
&= \ln\left(\left(\sum_{i=-\infty}^{\infty} p^{i^2} \right)^d \right) + \sum_{k=1}^{d} \ln(1/p) \cdot a_k^2.
\end{aligned}$$

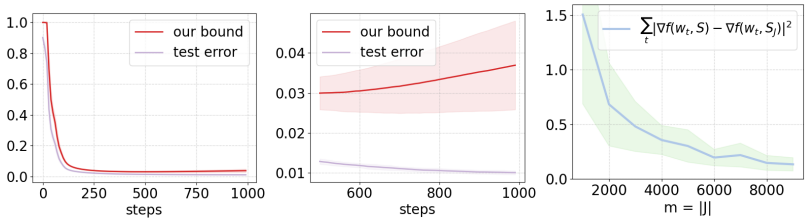
Since $|i| \leq i^2$ and $p \in (0, 1/3)$, we have $\ln\left((\sum_{i=-\infty}^{\infty} p^{i^2})^d \right)$ is at most $d \ln\left(1 + 2\sum_{i=1}^{\infty} p^i\right)$. It can be further bounded by $d \ln(1+3p)$. Moreover, it can be bounded by $3dp$ as $\ln(1+x) \leq x$. Thus, the above KL-divergence can be bounded by $3dp + \sum_{k=1}^{d} \ln(1/p) a_k^2$. Recall that $a_k := \lfloor \frac{\gamma_t}{\varepsilon_t} \cdot (\nabla_k f(w_{t-1}, S) - \nabla_k f(w_{t-1}, S_J)) \rfloor$ which is less than or equal to $\frac{\gamma_t}{\varepsilon_t} \cdot (\nabla_k f(w_{t-1}, S) - \nabla_k f(w_{t-1}, S_J))$. Therefore we have

$$\mathrm{KL}\left(W_t | W_0^{t-1} = w \,\middle|\middle|\, W'_t | {W'}_0^{t-1} = w \right) \leq 3dp + \frac{\ln(1/p)\gamma_t^2}{\varepsilon_t^2} \left\| \nabla f(w_{t-1}, S) - \nabla f(w_{t-1}, S_J) \right\|_2^2.$$

Plugging the above inequality into (1), we have

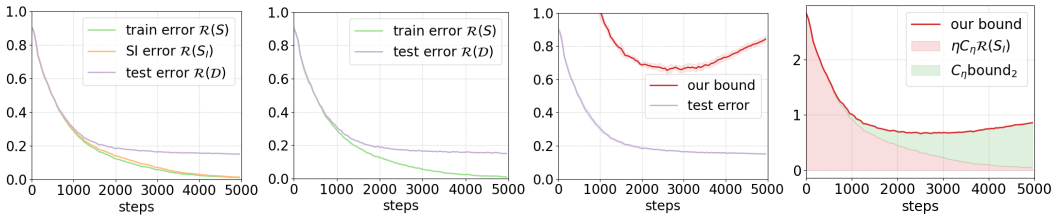
$$\mathrm{KL}\left(W_T \,\middle|\middle|\, P(S_J)\right) \leq \sum_{t=1}^{T} \left(3dp + \frac{\ln(1/p)\gamma_t^2}{\varepsilon_t^2} \left\| \nabla f(W_{t-1}, S) - \nabla f(W_{t-1}, S_J) \right\|_2^2 \right).$$

We conclude our proof by plugging it into Theorem 4.1 (setting $p = 1/(Td)$). □



(a) entire training path (b) later stage (c) different m

Figure 1: MNIST + CNN + FGD. In (a) and (b), our bound is defined by the RHS of Theorem 5.2 with $\eta = 1, \delta = 0.1$, and the test error is $\mathcal{R}(W_T, \mathcal{D})$. In (c), we show how cumulative gradient difference decreases as prior size m increases.



(a) errors (FSGD) (b) errors (SGD) (c) bound (d) details

Figure 2: MNIST + SimpleNet + FSGD/SGD. In (a), the yellow curve $\mathcal{R}(W_T, S_I)$ is very close to the train error. In (b), we show that the learning curve of SGD is similar to FSGD. In (c), our bound is defined by the RHS of Theorem 5.3 with $\eta = 3, \delta = 0.1$. In (d), we plot the composition of our bound, where $\text{bound}_2 := \frac{\ln(1/\delta)+3}{n-m} + \frac{\ln(dT)}{n-m} \mathbb{E}[\sum_{t=1}^{T} \frac{\gamma_t^2}{\varepsilon_t^2} \|\mathbf{g}_t\|^2]$.

FSGD We can use a similar approach to prove a generalization bound for FSGD (a variant of SGD). It is identical to Algorithm 1 except for the definitions of g_1 and g_2 replaced with:

$$g_1 \leftarrow \nabla f(W_{t-1}, S_{B_t}), \quad g_2 \leftarrow \nabla f(W_{t-1}, S_{B_t \cap J}),$$

where $B_t \subseteq [n]$ is a random batch index set independent of S, J and W_0^{t-1}. The following Theorem 5.3 gives the generalization bound for FSGD.

Theorem 5.3. *Suppose J is a random sequence consisting of m indices uniformly sampled from $[n]$ without replacement. Then for any $\delta \in (0,1), \varepsilon \in (0,1)$, FSGD satisfies the following generalization bound: w.p. at least $1-\delta$ over $S \sim \mathcal{D}^n$ and J:*

$$\mathcal{R}(W_T, \mathcal{D}) \leq \eta C_\eta \mathcal{R}(W_T, S_I) + C_\eta \cdot \frac{\ln(1/\delta)+3}{n-m} + \frac{C_\eta \ln(dT)}{n-m} \mathop{\mathbb{E}}_{W_0^T, B_0^T} \left[\sum_{t=1}^{T} \frac{\gamma_t^2}{\varepsilon_t^2} \|\mathbf{g}_t\|^2 \right],$$

where d is the dimension of parameter space, $I = [n] \backslash J$ includes indices out of J, $C_\eta := \frac{1}{1-e^{-\eta}}$ is a constant, and $\mathbf{g}_t := f(W_{t-1}, S_{B_t}) - \nabla f(W_{t-1}, S_{J \cap B_t})$ is the gradient difference.

5.1 Experiment

We run experiments for FGD and FSGD on MNIST [LeCun et al., 1998] and CIFAR10 [Krizhevsky et al., 2009] to investigate the performance of our bound as well as the properties of the algorithms.

Non-vacuous bound. For MNIST, we train a CNN ($d = 1.4 \cdot 10^6$) by FGD with $\gamma_t = \varepsilon_t = 0.01$ and momentum $\alpha = 0.9$). The size $m = |J|$ is set to $n/2 = 30000$. As shown in Figure 1a and 1b, our bound (Theorem 5.2 with $\eta = 1, \delta = 0.1$) tracks the testing error closely. At step $T = 990$, our bound is $\mathbf{0.0369}$ while the testing error is 0.0101. For CIFAR10, we train a SimpleNet [Hasanpour et al., 2016] without BatchNorm and Dropout. The number of parameters d is nearly $18 \cdot 10^6$. We use FSGD to train our model. The learning rate γ_t is set to 0.001, the precision ε_t is set to 0.004, and the momentum α is set to 0.99. The batch size is 2000. $m = |J|$ is set to $n/5 = 10000$. The result is shown in Figure 2c. We stop training at step $t = 5000$ when the training and testing error is 0.0133 and 0.1542, respectively. At that time, our testing error bound is $0.8589 < 1$ which is non-vacuous.

Decline of gradient difference. Intuitively, the cumulative squared norm of gradient difference $\mathbf{g}_t := \nabla f(W_t, S) - \nabla f(W_t, S_J)$ should decrease when increasing the size of J. Although we cannot prove a concentration like Lemma 6.1 that $\|\mathbf{g}_t\|^2$ is $O(1/m)$ since W_t of FGD depends on J, we still observe a decreasing phenomena when increasing m. The result is depicted in Figure 1c.

6 Gradient Langevin Dynamics

In this section, we present a $O(\frac{1}{n} + \frac{T}{n^2})$ generalization bound for the gradient Langevin Dynamics (GLD). The GLD learning algorithm can be viewed as gradient descent plus a gaussian noise. Formally, for a given $S \sim \mathcal{D}^n$, the update rule of GLD is defined as follows:

$$W_{t+1} \leftarrow W_t - \frac{\gamma_{t+1}}{n} \sum_{i=1}^{n} \nabla f(W_t, z_i) + \sigma_{t+1} \mathcal{N}(0, I_d), \tag{GLD}$$

Here the gradient $\nabla f(W_t, z_i)$ can be replaced with any gradient-like vector such as a clipped gradient. The output of GLD is the last step parameter W_T or any function of the whole training trajectory W_0^T.

We still use the data-dependent PAC-Bayesian framework (Theorem 4.1) to prove the generalization bound for GLD. Unlike FGD (Algorithm 1), GLD doesn't depend on the prior indices J, which enables us to prove the following concentration bound (Lemma 6.1) for the gradient difference. The proof is based on Lemma 3.4.

Lemma 6.1. *Let $S = (z_1, ...z_n)$ be any fixed training set. J is a random sequence including m indices uniformly sampled from $[n]$ without replacement, and $W = (W_0, ..., W_T)$ be any random sequence independent of J. Then the following bound holds with probability at least $1 - \delta$ over the randomness of J:*

$$\mathop{\mathbb{E}}_{W}\left[\sum_{t=1}^{T} \frac{\gamma_t^2}{\sigma_t^2} \|\nabla f(W_{t-1}, S) - \nabla f(W_{t-1}, S_J)\|^2\right] \leq \frac{C_\delta}{m} \mathop{\mathbb{E}}_{W}\left[\sum_{t=1}^{T} \frac{\gamma_t^2}{\sigma_t^2} L(W_{t-1})^2\right],$$

where $C_\delta = 4 + 2\ln(1/\delta) + 5.66\sqrt{\ln(1/\delta)}$, and $L(w) = \max_{i\in[n]} \|\nabla f(w, z_i)\|$.

Now we are ready to present our main result.

Theorem 6.2. *Suppose J is a random sequence consisting for m indices uniformly sampled from $[n]$ without replacement. Let W_T be the output of GLD. Then for any $\delta \in (0, \frac{1}{2})$ and $\eta > 0$, we have w.p. $\geq 1 - 2\delta$ over $S \sim \mathcal{D}^n$ and J, the following holds:*

$$\mathcal{R}(W_T, \mathcal{D}) \leq \eta C_\eta \mathcal{R}(W_T, S_{[n]\setminus J}) + \frac{C_\eta \ln(1/\delta)}{n-m} + \frac{C_\eta C_\delta}{2(n-m)m} \mathop{\mathbb{E}}_{W}\left[\sum_{t=1}^{T} \frac{\gamma_t^2}{\sigma_t^2} L(W_{t-1})^2\right],$$

where $L(w) := \max_{z\in S} \|f(w, z)\|$, $C_\delta = 4 + 2\ln(1/\delta) + 5.66\sqrt{\ln(1/\delta)}$, and $C_\eta = \frac{1}{1-e^{-\eta}}$.

Stochastic Gradient Langevin Dynamics For a given training data set S, the the update rule of SGLD is defined as:

$$W_{t+1} \leftarrow W_t - \gamma_{t+1} \nabla f(W_t, B_t) + \sigma_{t+1} \mathcal{N}(0, I_d), \tag{SGLD}$$

where B_t is a mini-batch with batch size b. Each batch is i.i.d. sampled from $\mathrm{uniform}(S)^b$. Note that the B_t is a sequence instead of a set, thus it may include duplicate elements. The output of SGLD is the whole training sequence $W = (W_0, ..., W_T)$. Similar to the analysis of GLD, we can prove the following bounds for SGLD.

Theorem 6.3. *Let Q_S be the distribution of SGLD's output when the training set is S. For any $\delta \in (0, 1)$ and $m \geq 1$, we have w.p. $\geq 1 - 2\delta$ over $S \sim \mathcal{D}^n$ and $J \sim Uniform([n])^m$, the following holds:*

$$\mathcal{R}(Q_S, \mathcal{D}) \leq \eta C_\eta \mathcal{R}(Q_S, S_I) + \frac{C_\eta \ln(1/\delta)}{n-m} + \frac{C_\eta}{n-m}\left(\frac{4}{b} + \frac{C_\delta}{2m}\right) \sum_{t=1}^{T} \frac{\gamma_t^2}{\sigma_t^2} L(W_{t-1})^2,$$

where $L(w) := \max_{z\in S} \|f(w, z)\|$, $C_\delta = 4 + 2\ln(1/\delta) + 5.66\sqrt{\ln(1/\delta)}$, b is the batch size, and $I = [n]\setminus J$.

References

Sanjeev Arora, Rong Ge, Behnam Neyshabur, and Yi Zhang. Stronger generalization bounds for deep nets via a compression approach. In *International Conference on Machine Learning*, pages 254–263. PMLR, 2018.

Sanjeev Arora, Simon Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, pages 322–332. PMLR, 2019.

Peter L Bartlett, Dylan J Foster, and Matus J Telgarsky. Spectrally-normalized margin bounds for neural networks. *Advances in neural information processing systems*, 30, 2017.

Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

Olivier Bousquet, Yegor Klochkov, and Nikita Zhivotovskiy. Sharper bounds for uniformly stable algorithms. In *Conference on Learning Theory*, pages 610–626. PMLR, 2020.

Yuheng Bu, Shaofeng Zou, and Venugopal V Veeravalli. Tightening mutual information-based bounds on generalization error. *IEEE Journal on Selected Areas in Information Theory*, 1(1): 121–130, 2020.

Yuan Cao and Quanquan Gu. Generalization bounds of stochastic gradient descent for wide and deep neural networks. *Advances in neural information processing systems*, 32, 2019.

Olivier Catoni. Pac-bayesian supervised classification: the thermodynamics of statistical learning. *arXiv preprint arXiv:0712.0248*, 2007.

Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12):124018, 2019.

Xi Chen, Simon S Du, and Xin T Tong. On stationary-point hitting time and ergodicity of stochastic gradient langevin dynamics. *Journal of Machine Learning Research*, 2020.

Yuansi Chen, Chi Jin, and Bin Yu. Stability and convergence trade-off of iterative optimization algorithms. *arXiv preprint arXiv:1804.01619*, 2018.

Thomas M Cover. *Elements of information theory*. John Wiley & Sons, 1999.

Devdatt P Dubhashi and Alessandro Panconesi. *Concentration of measure for the analysis of randomized algorithms*. Cambridge University Press, 2009.

John Duchi. Derivations for linear algebra and optimization. *Berkeley, California*, 3(1):2325–5870, 2007.

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

Vitaly Feldman and Jan Vondrak. High probability generalization bounds for uniformly stable algorithms with nearly optimal rate. In *Conference on Learning Theory*, pages 1270–1279. PMLR, 2019.

Mahdi Haghifam, Jeffrey Negrea, Ashish Khisti, Daniel M Roy, and Gintare Karolina Dziugaite. Sharpened generalization bounds based on conditional mutual information and an application to noisy, iterative algorithms. *Advances in Neural Information Processing Systems*, 33:9925–9935, 2020.

Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International conference on machine learning*, pages 1225–1234. PMLR, 2016.

Seyyed Hossein Hasanpour, Mohammad Rouhani, Mohsen Fayyaz, and Mohammad Sabokrou. Lets keep it simple, using simple architectures to outperform deeper and more complex architectures. *arXiv preprint arXiv:1608.06037*, 2016.

Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

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

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Guy Lever, François Laviolette, and John Shawe-Taylor. Tighter pac-bayes bounds through distribution-dependent priors. *Theoretical Computer Science*, 473:4–28, 2013.

Jian Li, Xuanyuan Luo, and Mingda Qiao. On generalization error bounds of noisy gradient methods for non-convex learning. In *International Conference on Learning Representations*, 2020.

Qianxiao Li, Cheng Tai, and E Weinan. Stochastic modified equations and adaptive stochastic gradient algorithms. In *International Conference on Machine Learning*, pages 2101–2110. PMLR, 2017.

Zhiyuan Li, Sadhika Malladi, and Sanjeev Arora. On the validity of modeling sgd with stochastic differential equations (sdes). *Advances in Neural Information Processing Systems*, 34, 2021.

Ben London. Generalization bounds for randomized learning with application to stochastic gradient descent. In *NIPS Workshop on Optimizing the Optimizers*, 2016.

Ben London. A pac-bayesian analysis of randomized learning with application to stochastic gradient descent. *Advances in Neural Information Processing Systems*, 30, 2017.

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

Wenlong Mou, Liwei Wang, Xiyu Zhai, and Kai Zheng. Generalization bounds of sgld for non-convex learning: Two theoretical viewpoints. In *Conference on Learning Theory*, pages 605–638. PMLR, 2018.

Vaishnavh Nagarajan and J Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Jeffrey Negrea, Mahdi Haghifam, Gintare Karolina Dziugaite, Ashish Khisti, and Daniel M Roy. Information-theoretic generalization bounds for sgld via data-dependent estimates. *Advances in Neural Information Processing Systems*, 32, 2019.

Behnam Neyshabur, Srinadh Bhojanapalli, and Nathan Srebro. A pac-bayesian approach to spectrally-normalized margin bounds for neural networks. In *International Conference on Learning Representations*, 2018.

Luca Oneto, Sandro Ridella, and Davide Anguita. Differential privacy and generalization: Sharper bounds with applications. *Pattern Recognition Letters*, 89:31–38, 2017.

Emilio Parrado-Hernández, Amiran Ambroladze, John Shawe-Taylor, and Shiliang Sun. Pac-bayes bounds with data dependent priors. *The Journal of Machine Learning Research*, 13(1):3507–3531, 2012.

Ankit Pensia, Varun Jog, and Po-Ling Loh. Generalization error bounds for noisy, iterative algorithms. In *2018 IEEE International Symposium on Information Theory (ISIT)*, pages 546–550. IEEE, 2018.

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

Hao Wang, Yizhe Huang, Rui Gao, and Flavio Calmon. Analyzing the generalization capability of sgld using properties of gaussian channels. *Advances in Neural Information Processing Systems*, 34, 2021.

Colin Wei, Jason D Lee, Qiang Liu, and Tengyu Ma. Regularization matters: Generalization and optimization of neural nets vs their induced kernel. *Advances in Neural Information Processing Systems*, 32, 2019.

Max Welling and Yee W Teh. Bayesian learning via stochastic gradient langevin dynamics. In *Proceedings of the 28th international conference on machine learning (ICML-11)*, pages 681–688. Citeseer, 2011.

Bingzhe Wu, Zhicong Liang, Yatao Bian, ChaoChao Chen, Junzhou Huang, and Yuan Yao. Generalization bounds for stochastic gradient langevin dynamics: A unified view via information leakage analysis. *arXiv preprint arXiv:2112.08439*, 2021.

Jun Yang, Shengyang Sun, and Daniel M Roy. Fast-rate pac-bayes generalization bounds via shifted rademacher processes. *Advances in Neural Information Processing Systems*, 32, 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017a.

Yuchen Zhang, Percy Liang, and Moses Charikar. A hitting time analysis of stochastic gradient langevin dynamics. In *Conference on Learning Theory*, pages 1980–2022. PMLR, 2017b.

Wenda Zhou, Victor Veitch, Morgane Austern, Ryan P Adams, and Peter Orbanz. Non-vacuous generalization bounds at the imagenet scale: a pac-bayesian compression approach. In *International Conference on Learning Representations*, 2018.

Wenda Zhou, Victor Veitch, Morgane Austern, Ryan P. Adams, and Peter Orbanz. Non-vacuous generalization bounds at the imagenet scale: a pac-bayesian compression approach. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Checklist

The checklist follows the references. Please read the checklist guidelines carefully for information on how to answer these questions. For each question, change the default **[TODO]** to [Yes] , [No] , or [N/A] . You are strongly encouraged to include a **justification to your answer**, either by referencing the appropriate section of your paper or providing a brief inline description. For example:

- Did you include the license to the code and datasets? [Yes]
- Did you include the license to the code and datasets? [No] The code and the data are proprietary.
- Did you include the license to the code and datasets? [N/A]

Please do not modify the questions and only use the provided macros for your answers. Note that the Checklist section does not count towards the page limit. In your paper, please delete this instructions block and only keep the Checklist section heading above along with the questions/answers below.

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] See the conclusion section.
   (c) Did you discuss any potential negative societal impacts of your work? [N/A]
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [Yes]
   (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]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (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)? [Yes] It can be found in our supplemental material.
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? [Yes]
   (b) Did you mention the license of the assets? [N/A]
   (c) Did you include any new assets either in the supplemental material or as a URL? [N/A]
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [N/A]
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [N/A] We only use the standard MNIST and CIFAR10 datasets.
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]