ON THE GENERALIZATION OF MODELS TRAINED WITH SGD: INFORMATION-THEORETIC BOUNDS AND IMPLICATIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

This paper follows up on a recent work of Neu (2021) and presents new and tighter information-theoretic upper bounds for the generalization error of machine learning models, such as neural networks, trained with SGD. We apply these bounds to analyzing the generalization behaviour of linear and two-layer ReLU networks. Experimental study based on these bounds provide some insights on the SGD training of neural networks. They also point to a new and simple regularization scheme which we show performs comparably to the current state of the art.

1 INTRODUCTION

The observation that high capacity deep neural networks trained with mini-batched stochastic gradient descent, referred to SGD in this paper, tend to generalize well (Zhang et al., 2017) contradicts the classical wisdom in statistical learning theory (e.g., Vapnik (1998)) and has stimulated intense research interest in understanding the generalization behaviour of modern neural networks.

In this direction, generalization bounds for over-parameterized neural networks are obtained (Du et al., 2018; Allen-Zhu et al., 2019; Bartlett et al., 2017; Neyshabur et al., 2015; 2018a;b; Arora et al., 2018; 2019) and a curious "double descent" phenomenon is observed and analyzed (Belkin et al., 2019; Nakkiran et al., 2019; Yang et al., 2020). Built on a connection between stability and generalization (Bousquet & Elisseeff, 2002), a stability-based bound is first presented in Hardt et al. (2016), followed by a surge of research effort exploiting similar approaches (London, 2017; Chen et al., 2018; Feldman & Vondrak, 2019; Lei & Ying, 2020; Bassily et al., 2020). Information-theoretic bounding techniques established recently (Russo & Zou, 2016; 2019; Xu & Raginsky, 2017; Asadi et al., 2018; Bu et al., 2020; Steinke & Zakynthinou, 2020; Dwork et al., 2015; Bassily et al., 2018; Asadi & Abbe, 2020; Hafez-Kolahi et al., 2020; Zhou et al., 2020) have also demonstrated great power in analyzing SGD-like algorithms. For example, Pensia et al. (2018) is the first to utilize information-theoretic bound in analyzing the generalization ability of SGLD (Gelfand & Mitter, 1991; Welling & Teh, 2011). The bound was subsquently improved by Negrea et al. (2019); Haghifam et al. (2020); Rodríguez-Gálvez et al. (2020); Wang et al. (2021b). Inspired by the work of Pensia et al. (2018), Neu (2021) presents an information- theoretic analysis of the models trained with SGD. The analysis of Neu (2021) constructs an auxiliary weight process parallel to SGD training and upper-bounds the generalization error through this auxiliary process.

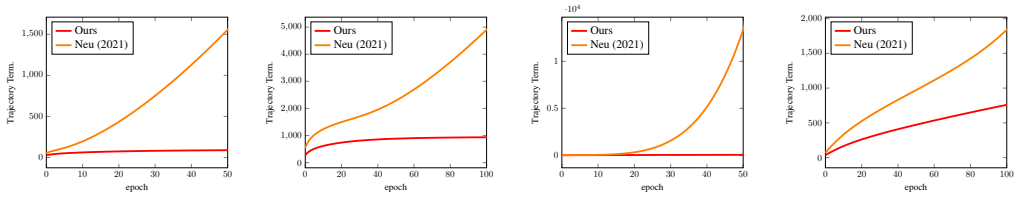
Another line of research connects the generalization of neural networks with the flatness of loss minima (Hochreiter & Schmidhuber, 1997) found by SGD or its variant (Keskar et al., 2017; Dinh et al., 2017; Dziugaite & Roy, 2017; Neyshabur et al., 2017; Chaudhari et al., 2017; Jastrzebski et al., 2017; Jiang et al., 2019; Zheng et al., 2021; Foret et al., 2020). This understanding has led to the discovery of new SGD-based training algorithms for improved generalization. For example, in a concurrent development by Zheng et al. (2021) and Foret et al. (2020), a local "max-pooling" operation is applied to the loss landscape prior to the SGD updates. This approach, referred to as AMP (Zheng et al., 2021) or SAM (Foret et al., 2020), is shown to make SGD favor flatter minima and achieve the state-of-the-art performance among various competitive regularization schemes.

In this paper, we focus on investigating the generalization of machine learning models trained with SGD. Although we are primarily motivated by the curiosity to understanding neural networks, the results of this paper in fact apply broadly to any model trained with SGD.

Our development builds upon that of Neu (2021). Following the same construction of the auxiliary weight process in Neu (2021), we present upper bounds of generalization error that improve upon Neu (2021) in two ways. This improvement is mainly due to removing an unnecessary term in the bound of Neu (2021), which the author refers to as "local gradient sensitivity", by invoking the HWI inequality (Raginsky & Sason, 2018) (i.e., Lemma 3 of this paper) and Lemma 4. Additionally, the improvement also benefits from the fact that removing "local gradient sensitivity" allows us to have a simple closed form of the optimal bound when the noise variance does not depend on time step. The bounds we obtain can be decomposed into two terms, one measuring the impact of training trajectories ("the trajectory term") and the other measuring the impact of the flatness of the found solution ("the flatness term[1]").

Remarkably, the generalization bound given in Neu (2021) also contains a trajectory term and a flatness term, where the flatness term is identical to ours due to the same construction of the auxillary weight process. However, our trajectory term is much tighter than that in Neu (2021). This is not only because we remove the unnecessary term, and the constant factor of our trajectory term is also half of that in Neu (2021). Figure 1 shows an experimental comparison of the trajectory term in our bound (Theorem 1) with that in the bound in Neu (2021) (re-stated as Lemma 2 in this paper for two neural network models. The trajectory terms are compared for two different settings of σ, which is the variance parameter of the noise used in the construction of the auxiliary weight process.

After the the initial preprint of Neu (2021), a revised version of the paper Neu et al. (2021) is presented, in which the bound is revised by fixing a coefficient error in the bound of Neu (2021) and adopts a slightly different form. It is particularly note-worthy that the development of the bound in Neu et al. (2021) also makes uses of the HWI inequality (or Lemma 3), but the unnecessary local gradient sensitivity term remains in the bound. This is because the development of Neu et al. (2021) requires constructing a ghost auxiliary weight process, which is completely avoided by using Lemma 4 in our paper. We will elaborate on this in an appropriate context.



(a) $\sigma = 10^{-5}$ (MNIST) (b) $\sigma = 10^{-6}$ (MNIST) (c) $\sigma = 10^{-5}$ (CIFAR10) (d) $\sigma = 10^{-6}$ (CIFAR10)

Figure 1: Comparison of the trajectory term between our bound (Theorem 1) and the bound in Neu (2021). (a)(b) MLP trained on MNIST. (c)(d) CNN trained on CIFAR-10.

We also provide an application of our bounds in analyzing the generalization behaviour of linear and two-layer ReLU networks, where we show that the activation state in ReLU networks plays an important role in generalization.

It is remarkable that removing the local gradient sensitivity term not only makes the bound much tighter, it also makes it much easier to evaluate. In this paper, we empirically validate the derived bounds, and provide various insights pertaining to the generalization behavior of models trained with SGD. For example, a particular interesting observation from our experiments is that a key quantity arising in the trajectory term of the bounds, which we refer to as *gradient dispersion*[2], reveals a double descent phenomenon with respect to training epochs. Most intriguingly, the valley in the double descent curve appears to mark the great divide between the "generalization regime" and the "memorization regime" of training. Furthering from this observation, we also show that it is possible to reduce the memorization effect by dynamically clipping the gradient and reducing its dispersion.

Our bounds also inspire a natural and simple solution to alleviate generalization error. Specifically, we propose a new training scheme, referred to as *Gaussian model perturbation* (GMP), aiming at reducing the flatness term of the bounds. This scheme effectively applies a local "average pooling"

[1]We note this term is correlated with flatness rather than precisely measures the flatness.

[2]The quantity is often referred to as gradient variance in the literature, but we prefer "dispersion" to "variance" so as to better comply with the mathematical conventions and avoid possible confusion.

to the empirical risk surface prior to SGD, greatly resembling the "max-pooling" approach adopted in AMP (Zheng et al., 2021). We demonstrate experimentally that GMP achieves a competitive performance with the current art of regularization schemes.

Proofs, additional discussions and experimental results are presented in Appendices.

Other Related Literature Gradient dispersion is mostly studied from optimization perspectives (Bottou et al., 2018; Roux et al., 2012; Johnson & Zhang, 2013; Wen et al., 2020; Faghri et al., 2020). Prior to this work, only a few works relate gradient dispersion with the generalization behaviour of the networks. In Neu (2021) and Wang et al. (2021b), gradient dispersion also appears in the generalization bounds. and there are limited studies characterizing the generalization performance and the gradient variance. Particularly, in Jiang et al. (2019), gradient dispersion is argued to capture a notion of "flatness" of the local minima of the loss landscape, thereby correlating with generalization. Injecting noise in the training process has been proposed in various regularization schemes (Bishop, 1995; Camuto et al., 2020; 2021; Srivastava et al., 2014; Wei et al., 2020). But unlike GMP derived in this paper, where noise is injected to the model parameters, noise in those schemes is injected either to the training data or to the network activation. Gradient clipping is a common technique for preventing gradient exploding (see, e.g., Merity et al. (2018); Peters et al. (2018)). This technique is also used in Zhang et al. (2019) to accelerate training. In this paper, gradient clipping is used to investigate and control the impact of gradient dispersion on generalization error.

2 PRELIMINARIES

Population Risk, Empirical Risk and Generalization Error Unless otherwise noted, a random variable will be denoted by a capitalized letter (e.g., Z), and its realization denoted by the corresponding lower-case letter (e.g. z). Let $\mathcal{Z}$ be the instance space of interest and μ be an unknown distribution on $\mathcal{Z}$, specifying random variable Z. Let $\mathcal{W} \subseteq \mathbb{R}^d$ be the space of hypotheses. Suppose that a training sample $S = (Z_1, Z_2, \ldots, Z_n)$ is drawn i.i.d. from μ and that a stochastic learning algorithm $\mathcal{A}$ takes S as its input and outputs a hypothesis $W \in \mathcal{W}$ according to some conditional distribution $P_{W|S}$ mapping $\mathcal{Z}^n$ to $\mathcal{W}$. Let $\ell : \mathcal{W} \times \mathcal{Z} \to \mathbb{R}^+$ be a loss function, where $\ell(w, z)$ measures the "unfitness" or "error" of any $z \in \mathcal{Z}$ with respect to a hypothesis $w \in \mathcal{W}$. The population risk, for any $w \in \mathcal{W}$, is defined as

$$L_\mu(w) \triangleq \mathbb{E}_{Z\sim\mu}[\ell(w, Z)].$$

The goal of learning is to find a hypothesis w that minimizes the population risk. But since μ is only partially accessible via the sample S, in practice, we instead turn to the empirical risk, defined as

$$L_S(w) \triangleq \frac{1}{n}\sum_{i=1}^{n} \ell(w, Z_i).$$

The expected generalization error of the learning algorithm $\mathcal{A}$ is then defined as

$$\mathrm{gen}(\mu, P_{W|S}) \triangleq \mathbb{E}_{W,S}[L_\mu(W) - L_S(W)],$$

where the expectation is taken over the joint distribution of (S, W) (i.e., $\mu^n \otimes P_{W|S}$).

Throughout this paper, we take ℓ as a continuous function (adopting the usual notion "surrogate loss" (Shalev-Shwartz & Ben-David, 2014).). Additionally, we assume that ℓ is differentiable almost everywhere with respect to both w and z. Furthermore we assume that $\ell(w, Z)$ is R-subgaussian[3] for any $w \in \mathcal{W}$. Note that a bounded loss is guaranteed to be subgaussian. Let $I(X; Y)$ denote the mutual information (Cover & Thomas, 2012) between any pair of random variables (X, Y). The following result is known.

Lemma 1 (Xu & Raginsky (2017, Theorem 1.)). *Assume the loss $\ell(w, Z)$ is R-subgaussian for any $w \in \mathcal{W}$. The generalization error of $\mathcal{A}$ is bounded by*

$$|\mathrm{gen}(\mu, P_{W|S})| \leq \sqrt{\frac{2R^2}{n} I(W; S)},$$

[3]A random variable X is R-subgaussian if for any ρ, $\log \mathbb{E} \exp\left(\rho\left(X - \mathbb{E}X\right)\right) \leq \rho^2 R^2/2$.

Stochastic Gradient Descent We now restrict the learning algorithm $\mathcal{A}$ to be the mini-batched SGD algorithm for empirical risk minimization. For each training epoch, the dataset S is randomly split into m disjoint mini-batches, each having size b, namely, $n = mb$. Based on each batch, one parameter update is performed. Specifically, let B_t denote the batch used for the tth update. Define

$$g(w, B_t) \triangleq \frac{1}{b} \sum_{z \in B_t} \nabla_w \ell(w, z),$$

namely, $g(w, B_t)$ is the average gradient computed for the batch B_t with respect to parameter w. The rule for the tth parameter update is then

$$W_t \triangleq W_{t-1} - \lambda_t g(W_{t-1}, B_t),$$

where λ_t is the learning rate at the step t. The initial parameter setting W_0 is assumed to be drawn from the zero-mean spherical Gaussian $\mathcal{N}(0, \sigma_0^2 \mathbf{I}_d)$ with variance σ_0^2 in each dimension. We will assume that the SGD algorithm stops after T updates and outputs W_T as the learned model parameter.

Given the training sample S, let ξ govern the randomness in the sequence $(B_1, B_2, \ldots, B_T)$ of batches. For the simplicity of notion, we will fix the configuration of ξ. That is, we will assume a fixed "batching trajectory", or a fixed way to shuffle the example indices $\{1, \ldots, n\}$ and divide them into m batches in each epoch. The presented generalization bounds of this paper can be extended to the case where the batching trajectory is uniformly random (as we set up above). This merely involves averaging over all batching trajectories or taking expectation over ξ.

Auxiliary Weight Process We now associate with the SGD algorithm an auxiliary weight process $\{\widetilde{W}_t\}$. Let $\sigma_1, \sigma_2, \ldots, \sigma_T$ be a sequence of positive real numbers. Define

$$\widetilde{W}_0 \triangleq W_0, \quad \text{and} \quad \widetilde{W}_t \triangleq \widetilde{W}_{t-1} - \lambda_t g(W_{t-1}, B_t) + N_t, \text{ for } t > 0,$$

where $N_t \sim \mathcal{N}(0, \sigma_t^2 \mathbf{I}_d)$ is a Gaussian noise. The relationship between this auxiliary weight process $\{\widetilde{W}_t\}$ and the weight process $\{W_t\}$ in SGD is shown in the Bayesian network below.

$$\begin{array}{ccccccccccc}
 & & N_1 & & N_2 & & \cdots & & N_{T-1} & & N_T \\
 & & \downarrow & & \downarrow & & & & \downarrow & & \downarrow \\
\widetilde{W}_0 & \rightarrow & \widetilde{W}_1 & \rightarrow & \widetilde{W}_2 & \rightarrow & \cdots & \rightarrow & \widetilde{W}_{T-1} & \rightarrow & \widetilde{W}_T \\
\| & \nearrow & & \nearrow & & \nearrow & & & & \nearrow & \\
W_0 & \rightarrow & W_1 & \rightarrow & W_2 & \rightarrow & \cdots & \rightarrow & W_{T-1} & \rightarrow & W_T
\end{array}$$

Let $\Delta_t = \sum_{\tau=1}^{t} N_\tau$. Noting that the weight updates in $\{\widetilde{W}_t\}$ uses the same gradient signal as that used in $\{W_t\}$ (which depends on W_{t-1} not $\widetilde{W}_{t-1}$), it is immediate that $\widetilde{W}_t = W_t + \Delta_t$. Note that this auxiliary process follows the same construction as Neu (2021), which we will use to study the generalization error of SGD. To that end, let's define *gradient dispersion* by

$$\mathbb{V}_t(w) \triangleq \mathbb{E}\left[||g(w, B_t) - \mathbb{E}\left[\nabla_w \ell(W, Z)\right]||_2^2\right],$$

where the inside expectation is taken over $(W, Z) \sim \mu \otimes P_{W|Z}$. For a given sample $s \in \mathcal{Z}^n$, define

$$\gamma(w, s) \triangleq \mathbb{E}\left[L_s(w + \Delta_T) - L_s(w)\right],$$

where the expectation is taken over Δ_T and $L_s(w)$ is the empirical risk of s at parameter w.

In the remainder of the paper, let S' denote another sample drawn from μ^n, independent of all other random variables. The main generalization bound in Neu et al. (2021) is re-stated below.

Lemma 2 ((Neu et al., 2021, Theorem 1.)). *The generalization error of SGD is upper bounded by*

$$|\text{gen}(\mu, P_{W_T|S})| \leq \sqrt{\frac{4R^2}{n} \sum_{t=1}^{T} \frac{\lambda_t^2}{\sigma_t^2} \mathbb{E}\left[\Psi(W_{t-1}) + \widetilde{\mathbb{V}}_t(W_{t-1})\right]} + |\mathbb{E}\left[\gamma(W_T, S) - \gamma(W_T, S')\right]|,$$

where $\Psi(w_{t-1}) \triangleq \mathbb{E}\left[||\mathbb{E}\left[\nabla_w(w_{t-1}, Z)\right] - \mathbb{E}\left[\nabla_w(w_{t-1} + \zeta, Z)\right]||_2^2\right]$, $\zeta \sim \mathcal{N}(0, 2\sum_{i=1}^{t-1} \sigma_i^2 \mathrm{I}_d)$ *and* $\widetilde{\mathbb{V}}_t(w) \triangleq \mathbb{E}\left[||g(w, B_t) - \mathbb{E}\left[\nabla_w \ell(w, Z)\right]||_2^2\right]$.

The term $\Psi(w_{t-1})$ in the bound is referred to as "local gradient sensitivity" in Neu et al. (2021).

3 NEW GENERALIZATION BOUNDS FOR SGD

The improvement offered by our new bound upon the bound in Neu (2021) or Lemma 2 is the removal of the local sensitivity term $\Psi(w_{t-1})$. This requires invoking a vector version of the HWI inequality (Raginsky & Sason, 2018, Lemma 3.4.2), which we first state.

Lemma 3. *Let X and Y be two random vectors in $\mathbb{R}^d$, and let $N \sim \mathcal{N}(0, \mathbf{I}_d)$ be independent of (X, Y). Then, for every $t > 0$, $\mathrm{D_{KL}}(P_{X+\sqrt{t}N}||P_{Y+\sqrt{t}N}) \leq \frac{1}{2t}\mathbb{E}\left[||X-Y||_2^2\right]$, where $\mathrm{D_{KL}}$ is the KL divergence.*

Note that Neu et al. (2021) also makes uses of Lemma 3 in their revision. However, the development in Neu et al. (2021) requires constructing a ghost auxiliary weight process in which an independent noise perturbation Δ_t' is introduced. Due to the mismatch between Δ_t and Δ_t', the local gradient sensitivity term $\Psi(W_{t-1})$ appears in their bound. In this paper we waive this Δ_t' by using the following lemma.

Lemma 4. *Let X,Y and Δ be random variables which are all independent of $N \sim \mathcal{N}(0, \mathrm{I})$. Let $Z = Y + \Delta$, then for any σ and any function f, we have*

$$I(f(Z, X) + \sigma N; X|Y) \leq \frac{1}{2\sigma^2}\mathbb{E}\left[||f(Z, X) - \mathbb{E}\left[f(Z, X)\right]||^2\right]$$

The proof of this lemma is inspired by the techniques used in Wang et al. (2021a) and Guo et al. (2005). Exploiting Lemma 4 gives us a crisp way to have the following result, where $\Psi(W_{t-1})$ will not appear.

Lemma 5. *Let $G_t = -\lambda_t g(W_{t-1}, B_t)$. Then, $I(G_t + N_t; S|\widetilde{W}_{t-1}) \leq \frac{\lambda_t^2}{2\sigma_t^2}\mathbb{E}\left[\mathbb{V}_t(W_{t-1})\right]$.*

In this lemma, the mutual information $I(G_t+N_t; S|\widetilde{W}_{t-1})$ roughly indicates the degree by which the SGD's updating signal G_t (smoothed with noise) depends on the training sample S, when B_t is used for computing the gradient. When this dependency is strong (giving rise to a high value of the mutual information), the model conceivably tends to overfit the training sample. This lemma suggests that the strength of this dependency can be upper-bounded by the expected gradient dispersion at the current weight configuration. In our experiments, we will estimate the expected gradient dispersion and validate this intuition.

We are now in a position to state our main theorem.

Theorem 1. *The generalization error of SGD is upper bounded by*

$$|\mathrm{gen}(\mu, P_{W_T|S})| \leq \sqrt{\frac{R^2}{n}\sum_{t=1}^{T}\frac{\lambda_t^2}{\sigma_t^2}\mathbb{E}\left[\mathbb{V}_t(W_{t-1})\right]} + \left|\mathbb{E}\left[\gamma(W_T, S) - \gamma(W_T, S')\right]\right|. \tag{1}$$

Further, assume $L_\mu(w_T) \leq \mathbb{E}_\Delta\left[L_\mu(w_T + \Delta_T)\right]$, ℓ is twice differentiable, and σ_t^2 is independent of t. Denote by H_{W_T} the Hessian of the loss with respect to W_T and let $\mathrm{Tr}(\cdot)$ denote trace. Then

$$\mathrm{gen}(\mu, P_{W_T|S}) \leq \frac{3}{2}\left(\sum_{t=1}^{T}\frac{R^2\lambda_t^2 T}{n}\mathbb{E}\left[\mathbb{V}_t(W_{t-1})\right]\mathbb{E}\left[\mathrm{Tr}\left(\mathrm{H}_{W_T}(Z)\right)\right]\right)^{\frac{1}{3}} \tag{2}$$

Remark 1. *Removing $\Psi(W_{t-1})$ will make the bound much tighter as shown in Figure 1. This remarkable improvement should come at no surprise, since $\Psi(W_{t-1})$ monotonically increases with training epochs (noting that $\Psi(W_{t-1})$ has the cumulative variance $2\sum_{i=1}^{t-1}\sigma_i^2\mathrm{I}_d$) while $\mathbb{V}_t(W_{t-1})$ appears decreasing (see Figure 4a). Figure 1 also indicates that if the noise variance σ_t is made small, one expects the gradient dispersion to dominate the trajectory term in Lemma 2. However, the factor $1/\sigma_t^2$ will become too large, make the bound loose. Additionally, the constant factor of the trajectory term in Eq. 1 is half of that in Lemma 2.*

Remark 2. *The condition $L_\mu(w_T) \leq \mathbb{E}_{\Delta_T}\left[L_\mu(w_T + \Delta_T)\right]$ indicates that the perturbation does not decrease the population risk. This is also assumed in Foret et al. (2020) in the derivation of a PAC-Bayesian generalization bound.*

In the bound of Eq.1, the first term captures the impact of the training trajectory ("trajectory term"), and the second term captures the impact of the final solution, which in fact measures the flatness for the loss landscape at the found solution ("flatness term"). It is worth noting that using both the gradient dispersion (or the trace of the gradient's covariance matrix), and the trace of Hessian to characterize generalization is not new. For example, the Takeuchi information criterion used by Thomas et al. (2020) consists of Hessian and gradient dispersion (referred to as "curvature" and "noise" by the authors), which is shown relevant to estimating the generalization gap.

The learning rate in SGD has explicitly appeared in the trajectory term of Eq.2 in Theorem 1. From the way it appears in the bound, one may be tempted to assert that a small learning rate will improve generalization. This would then contradict some previous observations (Jastrzebski et al., 2017; Wu et al., 2018; He et al., 2019), in which large learning rate will benefit generalization. In addition, the bound also suggests that the batch size has some direct impact on the trajectory term through gradient dispersion. We investigate this by performing experiments with varying learning rates and batch sizes (see Figure 6 in Appendix E.2). As seen in the experiments, increasing the learning rate impacts the trajectory term and the flatness term in opposite ways, i.e. making one increase and the other decrease. A similar (but reverted) behaviour is also observed with batch sizes. This makes the generalization bound in Eq.2, have a rather complex relationship with the settings of learning rate and batch size.

Eq.2 follows from Eq.1 by minimizing the bound over σ. The bound in Eq.2 is thus independent of a choice of σ_t's and can be computed easily and efficiently.

To summarize, we remark that these bounds suggest that in order for the model to generalize well, both the trajectory term and the flatness term need to be small — the former involves the interaction of the learning rate and batch size with the gradient dispersion along the training trajectory, whereas the latter depends on the flatness of the empirical risk surface at the found solution.

4 Application in Linear Networks and Two-Layer ReLU Networks.

We now apply Theorem 1 to two neural network models in a regression setting. Let $Z = [X, Y] \in \mathbb{R}^{d_0+1}$. Assume $||X|| = 1$. We use SGD to train a model $f(W, \cdot) : \mathbb{R}^{d_0} \to \mathbb{R}$ and define the loss as $\ell(W, Z) = 1/2(Y - f(W, X))^2$. The bounds below are both in the form of the product of the flatness term and the trajectory term.

Theorem 2 (Linear Networks)**.** *Let $f(W, X) = W^T X$. Then,*

$$\mathrm{gen}(\mu, P_{W_T|S}) \leq 3 \left(\sum_{t=1}^{T} \frac{R^2 \lambda_t^2 T}{4n} \mathbb{E}\left[\ell(W_{t-1}, Z)\right] \right)^{\frac{1}{3}}.$$

Theorem 3 (Two-Layer ReLU Networks)**.** *Following Arora et al. (2019), consider $f(W, X) = \frac{1}{\sqrt{m}} \sum_{r=1}^{m} A_r \mathrm{ReLU}(W_r^T X)$ where m is the width of the neural network, $A_r \sim \mathrm{unif}(\{+1, -1\})$ and $\mathrm{ReLU}(\cdot)$ is the ReLU activation. We only train the first layer parameters $W = [W_1, W_2, \ldots, W_m] \in \mathbb{R}^{d_0 \times m}$ and fix the second layer parameters $A = [A_1, A_2, \ldots, A_m] \in \mathbb{R}^m$ during training. Then,*

$$\mathrm{gen}(\mu, P_{W_T|S}) \leq 3 \left(\sum_{r=1}^{m} \mathbb{E}\left[\frac{\mathbb{I}_{r,i,T}}{m}\right] \sum_{t=1}^{T} \frac{R^2 \lambda_t^2 T}{4n} \mathbb{E}\left[\sum_{r=1}^{m} \frac{\mathbb{I}_{r,i,t}}{m} \ell(W_{t-1}, Z)\right] \right)^{\frac{1}{3}},$$

where $\mathbb{I}_{r,i,t} = \mathbb{I}\{W_{t-1,r}^T X_i \geq 0\}$ and $\mathbb{I}$ is the indicator function.

Remark 3. *Compared with Theorem 2, we notice that in the two-layer ReLU network, the ReLU activation state along the training trajectory plays a key role in the bound of Theorem 3. Specifically, the weights of the deactivated neurons do not contribute to the bound of Theorem 3, making the bound not explicitly depend on the model dimension d. This result also suggests that sparsely activated ReLU networks are expected to generalize better. Despite various empirical evidence pointing to this behaviour (see, e.g., Glorot et al. (2011)), to the best of our knowledge, this theorem provides the first theoretical justification in this regard.*

Figure 2 shows that this bound can characterize the dynamics of generalization gap of the ReLU network. We defer the implementation details including the linear network's plot to Appendix.

In the remainder of this paper, we will study more complex network architectures beyond the linear and two-layer ReLU networks and implications of our bounds in those settings.

5 Experimental Study

Bound Verification We first verify our bound of Eq.2 in Theorem 1 by training an MLP (with one hidden layer) and an AlexNet (Krizhevsky et al., 2012) on MNIST and CIFAR10 (Krizhevsky, 2009), respectively. To simplify estimation, we fix the weight initialization and set σ_t and λ_t to be constants σ and λ, respectively. To compute $\sum_{t=1}^{T} \mathbb{E}\left[\mathbb{V}_t(W_{t-1})\right]$, we compute the gradient dispersion as its empirical estimate from a batch, utilizing a PyTorch (Paszke et al., 2019) library BackPack (Dangel et al., 2020). To compute $\mathrm{Tr}\left(\mathbb{E}\left[\mathrm{H}_{W_T}(Z)\right]\right)$, we use the PyHessian library (Yao et al., 2020) to compute the Hessian. To choose the variance proxy R, we first collected all the per-instance losses $\ell(W_{t-1}, Z_i)$ that were observed during training, then we let $R = (max_{i,t}\ell(W_{t-1}, Z_i) - min_{i,t}\ell(W_{t-1}, Z_i))/2$ in our experiments.

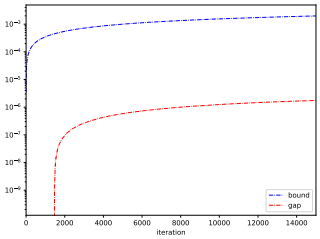
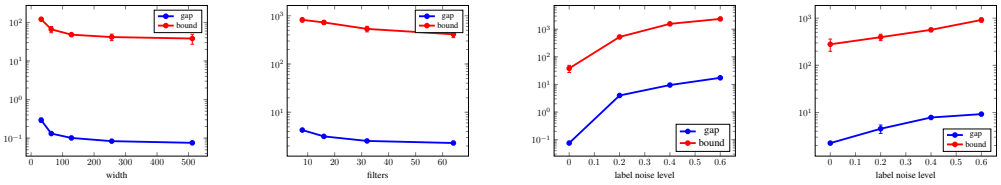


Figure 2: Generalization gap of a two-layer ReLU network v.s. Theorem 3.

We perform experiments with varying network width and varying levels of label noise. Specifically, noise level ϵ refers to the setting where we replace the labels of ϵ fraction of the training and testing instances with random labels. The estimated bound is compared against the true generalization gap, namely, the difference between the training loss and testing loss, and is shown in Figure 3.



(a) MLP on MNIST (b) AlexNet on CIFAR10 (c) MLP on MNIST (d) AlexNet on CIFAR10

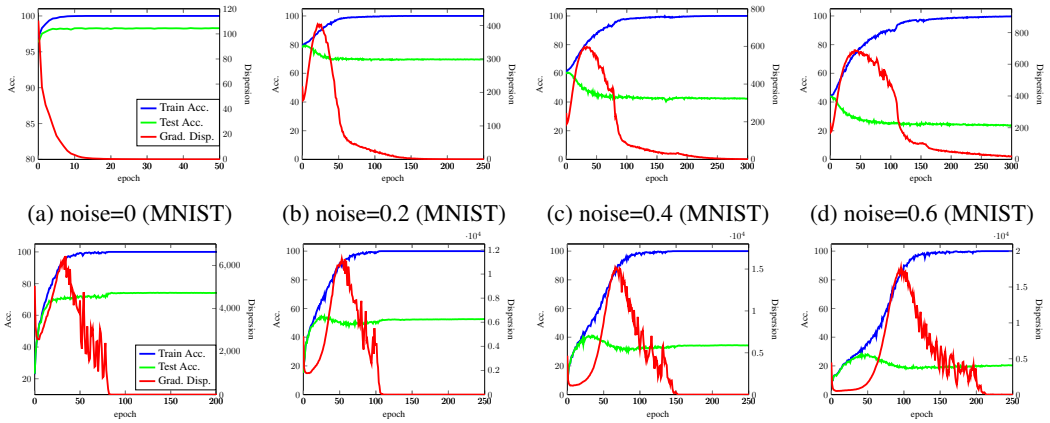
Figure 3: Estimated bound and empirical generalization gap ("gap") as functions of network width ((a) and (b)) and label noise level ((c) and (d)). Note that Y-axis is in log scale.

In Figure 3, we see that in all cases the estimated bound follows closely the trend of the true generalization gap. The fact that the bound curve consistently tracks the gap curve under various label noise levels indicates that our bound very well captures the changes of the data distribution. Note that in Figure 3 (a) and (b), our bound decays with the increase of the model size, showing a trend as opposite to the bounds obtained in classical learning theory. But such a trend clearly better explains the generalization behaviour of modern neural networks.

Epoch-wise Double Descent of Gradient Dispersion We experimentally investigate the impact of gradient dispersion on the training of the neural networks by fixing the learning rate, batch size and weight initialization for the each model (MLP for MNIST, AlexNet for CIFAR10). For each model and various label noise levels, we plot in Figure 4 the evolution of the (empirical) gradient dispersion $\widehat{\mathbb{V}}_t(w_{t-1})$, training accuracy and testing accuracy across training epochs.

An intriguing epoch-wise "double descent" phenomenon is observed, particularly when the labels are noisy. According to the double descent curve, the training may be split into three phases (e.g., Figure 4 (h)). In the first phase, the gradient dispersion rapidly descends and maintains a very low level. In this phase, both training and test accuracies increase while maintaining a very small generalization gap. This suggests that the network in this phase is extracting useful patterns and generalizes well. In the second phase, the gradient dispersion starts increasing until it reaches a peak value. In this phase, the training and testing accuracies gradually diverge, marking the model

entering an overfitting or "memorization" regime – when the data contains the noisy labels, the network mostly tries to memorize the labels in the training set. In the third phase, the gradient dispersion descends again, reaching a low value. In this phase, the model continuously overfits the training data, until the training and testing curves reach their respective maximum and minimum. It appears that the timing of the three phases depends on the dataset and the label noise level. For simpler data (e.g. MNIST) and cleaner datasets , the first phase may be shorter. This is arguably because in these datasets, extracting useful patterns is relatively easier. Nonetheless, the valley in the double-descent curve appears to mark a "great divide" between generalization and memorization.



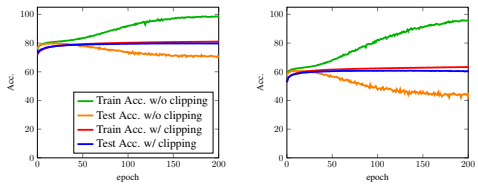
(a) noise=0 (MNIST) (b) noise=0.2 (MNIST) (c) noise=0.4 (MNIST) (d) noise=0.6 (MNIST)

(e) noise=0 (CIFAR10) (f) noise=0.2 (CIFAR10) (g) noise=0.4 (CIFAR10) (h) noise=0.6 (CIFAR10)

Figure 4: Dynamics of gradient dispersion, in relation to training/testing accuracies.

Dynamic Gradient Clipping Inspired by our generalization bounds and above observations, one way to reduce the generalization error is to control the trajectory term of the bounds by reducing the gradient dispersion in each training step. Here we investigate a simple scheme that dynamically clips the gradient norm so as to reduce the gradient dispersion. Specifically, whenever the current gradient norm is larger than the gradient norm K steps earlier, or $||g(W_t, B_t)||_2 > ||g(W_{t-K}, B_{t-K})||_2$ (i.e., the model is expected to have entered the "memorization" regime), we reduce the norm of the current gradient $g(W_t, B_t)$ to α fraction of $||g(W_{t-K}, B_{t-K})||_2$, for some prescribed value $\alpha < 1$.

The effectiveness of this scheme is best demonstrated when the labels contain noise. As shown in Figure 5 (and Figure 8 in Appendix E.4), dynamic gradient clipping significantly closes the gap between the training accuracy and the testing accuracy. The models trained with this scheme maintain a near-optimal testing accuracy (e.g., about 80% when the label noise level of MNIST is 0.2), without suffering from the severe memorization effect as seen in models trained without this scheme. Further understanding of the double-descent phenomenon of the gradient dispersion may enable more delicate design of such a dynamic clipping scheme and potentially lead to novel and powerful regularization techniques.



(a) noise=0.2 (MNIST) (b) noise=0.4 (MNIST)

Figure 5: Dynamic Gradient Clipping.

6 A Practical Implication: Gaussian Model Perturbation

The appearance of the flatness term in our generalization bounds suggests that for an empirical risk minimizer w^* to generalize well, it is necessary that the empirical risk surface at w^* is flat, or insensitive to a small perturbation of w^*. This naturally motivates a training scheme using the following regularized loss:

$$\min_w L_s(w) + \rho \mathop{\mathbb{E}}_{\Delta \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)} [L_s(w + \Delta) - L_s(w)],$$

where ρ is a hyper-parameter. Replacing the expectation above with its stochastic approximation using k realizations of Δ gives rise to the following optimization problem.

$$\min_w \frac{1}{b} \sum_{z \in B} \left((1-\rho)\ell(w,z) + \rho \frac{1}{k} \sum_{i=1}^{k} (\ell(w+\delta_i, z)) \right).$$

We refer to the SGD training scheme using this loss as *Gaussian model perturbation* or GMP. Notably, GMP requires $k+1$ forward passes for every parameter update. Empirical evidence shows that a small k, for example, $k=3$, already gives competitive performance. Implementing the $k+1$ forward passes on parallel processors further reduces the computation load.

Method	SVHN	CIFAR-10	CIFAR-100
ERM	96.86±0.060	93.68±0.193	72.16±0.297
Dropout	97.04±0.049	93.78±0.147	72.28±0.337
L. S.	96.93±0.070	93.71±0.158	72.51±0.179
Flooding	96.85±0.085	93.74±0.145	72.07±0.271
MixUp	96.91±0.057	**94.52±0.112**	73.19±0.254
Adv. Tr.	97.06±0.091	93.51±0.130	70.88±0.145
AMP	**97.27±0.015**	94.35±0.147	74.40±0.168
GMP^3	97.18±0.057	94.33±0.094	74.45±0.256
GMP^{10}	97.09±0.068	94.45±0.158	**75.09±0.285**

Table 1: Top-1 classification accuracy acc.(%) of VGG16. We run experiments 10 times and report the mean and the standard deviation of the testing accuracy. Superscript denotes the number of sampled Gaussian noises during training.

We compare GMP with several major regularization schemes in the current art, including Dropout (Srivastava et al., 2014), label smoothing (L.S.) (Szegedy et al., 2016), Flooding (Ishida et al., 2020), MixUp (Zhang et al., 2018), adversarial training (Adv. Tr.) (Goodfellow et al., 2015), and AMP (Zheng et al., 2021). We will also include ERM as the baseline, where no regularization method applied. The compared schemes are evaluated on three popular benchmark image classification datasets SVHN (Netzer et al., 2011), CIFAR-10 and CIFAR-100 (Krizhevsky, 2009). Two representative deep architectures VGG16 (Simonyan & Zisserman, 2015) and PreActResNet18 (He et al., 2016) are taken as the underlying model. We train the models for 200 epochs by SGD. The learning rate is initialized as 0.1 and divided by 10 after 100 and 150 epochs. For all compared models, the batch size is set to 50 and weight decay is set to 10^{-4}. For GMP, we choose $\rho = 0.5$ and set the standard deviation of the Gaussian noise Δ to 0.03. The value of k is chosen as 3 and 10 respectively (referred to as GMP^3 and GMP^{10}).

The performances of all compared schemes are given in Table 1 (and the results of PreActResNet18 are shown in Appendix E.6). For the compared regularization schemes except GMP, we directly report their performances as given in Zheng et al. (2021). Table 1 demonstrates the effectiveness of GMP. Overall GMP performs comparably to the current art of regularization schemes, although appearing slighly inferior to the most recent record given by AMP on SVHN and MixUp on CIFAR-10, respectively (Zheng et al., 2021). Noting that the key ingredient of AMP, "max-pooling" in the parameter space, greatly resembles regularization term in GMP, which may be seen as "average-pooling" in the same space.

7 Conclusion and Outlook

This paper presents new information theoretic generalization bounds for models, such as linear networks and two-layer ReLU neural networks, trained with SGD, improving upon the results of Neu (2021). Our bounds naturally point to new and effective regularization schemes. At the same time, our bounds and experimental study reveal interesting phenomena in the SGD training of neural networks.

There are still many directions for improving the bounds. First, it is possible to extend the theoretical results in this work by using the techniques introduced by other information-theoretic bounds such as Haghifam et al. (2020) and Wang et al. (2021a). Additionally, it's worth mentioning that one may obtain better results by further developing the relationship between the trajectory term and the flatness term from an optimization perspective. Some recent works based on backward error analysis (Barrett & Dherin, 2021; Smith et al., 2021) may provide analytic tools to investigate this.

REFERENCES

Zeyuan Allen-Zhu, Yuanzhi Li, and Zhao Song. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pp. 242–252. PMLR, 2019.

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*, pp. 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*, pp. 322–332. PMLR, 2019.

Amir R Asadi and Emmanuel Abbe. Chaining meets chain rule: Multilevel entropic regularization and training of neural networks. *Journal of Machine Learning Research*, 21(139):1–32, 2020.

Amir R Asadi, Emmanuel Abbe, and Sergio Verdú. Chaining mutual information and tightening generalization bounds. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, pp. 7245–7254, 2018.

David Barrett and Benoit Dherin. Implicit gradient regularization. In *International Conference on Learning Representations*, 2021.

Peter L Bartlett, Dylan J Foster, and Matus Telgarsky. Spectrally-normalized margin bounds for neural networks. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 6241–6250, 2017.

Raef Bassily, Shay Moran, Ido Nachum, Jonathan Shafer, and Amir Yehudayoff. Learners that use little information. In *Algorithmic Learning Theory*, pp. 25–55. PMLR, 2018.

Raef Bassily, Vitaly Feldman, Cristóbal Guzmán, and Kunal Talwar. Stability of stochastic gradient descent on nonsmooth convex losses. *Advances in Neural Information Processing Systems*, 33, 2020.

Mikhail Belkin, Daniel Hsu, Siyuan Ma, and Soumik Mandal. Reconciling modern machine-learning practice and the classical bias–variance trade-off. *Proceedings of the National Academy of Sciences*, 116(32):15849–15854, 2019.

Chris M Bishop. Training with noise is equivalent to tikhonov regularization. *Neural computation*, 7(1):108–116, 1995.

Léon Bottou, Frank E Curtis, and Jorge Nocedal. Optimization methods for large-scale machine learning. *Siam Review*, 60(2):223–311, 2018.

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

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.

Alexander Camuto, Matthew Willetts, Umut Simsekli, Stephen J. Roberts, and Chris C. Holmes. Explicit regularisation in gaussian noise injections. In *Advances in Neural Information Processing Systems*, 2020.

Alexander Camuto, Xiaoyu Wang, Lingjiong Zhu, Chris Holmes, Mert Gürbüzbalaban, and Umut Şimşekli. Asymmetric heavy tails and implicit bias in gaussian noise injections. *arXiv preprint arXiv:2102.07006*, 2021.

Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer T. Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. In *5th International Conference on Learning Representations*. OpenReview.net, 2017.

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 and Joy A Thomas. *Elements of Information Theory*. John Wiley & Sons, 2012.

Felix Dangel, Frederik Kunstner, and Philipp Hennig. Backpack: Packing more into backprop. In *8th International Conference on Learning Representations*. OpenReview.net, 2020.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, pp. 1019–1028. PMLR, 2017.

Simon S Du, Xiyu Zhai, Barnabas Poczos, and Aarti Singh. Gradient descent provably optimizes over-parameterized neural networks. In *International Conference on Learning Representations*, 2018.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. Generalization in adaptive data analysis and holdout reuse. In *Proceedings of the 28th International Conference on Neural Information Processing Systems-Volume 2*, pp. 2350–2358, 2015.

Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Proceedings of the 33rd Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, 2017.

Fartash Faghri, David Duvenaud, David J Fleet, and Jimmy Ba. A study of gradient variance in deep learning. *arXiv preprint arXiv:2007.04532*, 2020.

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

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.

Saul B Gelfand and Sanjoy K Mitter. Recursive stochastic algorithms for global optimization in r^d. *SIAM Journal on Control and Optimization*, 29(5):999–1018, 1991.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *14th International Conference on Artificial Intelligence and Statistics*, volume 15, pp. 315–323, 2011.

Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *3rd International Conference on Learning Representations, ICLR*, 2015.

Dongning Guo, Shlomo Shamai, and Sergio Verdú. Mutual information and minimum mean-square error in gaussian channels. *IEEE transactions on information theory*, 51(4):1261–1282, 2005.

Hassan Hafez-Kolahi, Zeinab Golgooni, Shohreh Kasaei, and Mahdieh Soleymani. Conditioning and processing: Techniques to improve information-theoretic generalization bounds. *Advances in Neural Information Processing Systems*, 33, 2020.

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. *arXiv preprint arXiv:2004.12983*, 2020.

Moritz Hardt, Ben Recht, and Yoram Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning*, pp. 1225–1234. PMLR, 2016.

Fengxiang He, Tongliang Liu, and Dacheng Tao. Control batch size and learning rate to generalize well: Theoretical and empirical evidence. In *Advances in Neural Information Processing Systems 32*, pp. 1141–1150, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pp. 630–645, 2016.

Sepp Hochreiter and Jürgen Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997.

Takashi Ishida, Ikko Yamane, Tomoya Sakai, Gang Niu, and Masashi Sugiyama. Do we need zero training loss after achieving zero training error? In *Proceedings of the 37th International Conference on Machine Learning, ICML*, 2020.

Stanislaw Jastrzebski, Zachary Kenton, Devansh Arpit, Nicolas Ballas, Asja Fischer, Yoshua Bengio, and Amos Storkey. Three factors influencing minima in sgd. *arXiv preprint arXiv:1711.04623*, 2017.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2019.

Rie Johnson and Tong Zhang. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems*, 26:315–323, 2013.

Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *5th International Conference on Learning Representations*. OpenReview.net, 2017.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012.

Yunwen Lei and Yiming Ying. Fine-grained analysis of stability and generalization for stochastic gradient descent. In *International Conference on Machine Learning*, pp. 5809–5819. PMLR, 2020.

Ben London. A pac-bayesian analysis of randomized learning with application to stochastic gradient descent. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 2935–2944, 2017.

Stephen Merity, Nitish Shirish Keskar, and Richard Socher. Regularizing and optimizing lstm language models. In *International Conference on Learning Representations*, 2018.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. In *International Conference on Learning Representations*, 2019.

Jeffrey Negrea, Mahdi Haghifam, Gintare Karolina Dziugaite, Ashish Khisti, and Daniel M Roy. Information-theoretic generalization bounds for sgld via data-dependent estimates. In *Advances in Neural Information Processing Systems*, pp. 11013–11023, 2019.

Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y Ng. Reading digits in natural images with unsupervised feature learning. In *NeurIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

Gergely Neu. Information-theoretic generalization bounds for stochastic gradient descent. *arXiv preprint arXiv:2102.00931*, 2021.

Gergely Neu, Gintare Karolina Dziugaite, Mahdi Haghifam, and Daniel M Roy. Information-theoretic generalization bounds for stochastic gradient descent. In *COLT*, 2021.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pp. 1376–1401. PMLR, 2015.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems*, pp. 5947–5956, 2017.

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*, 2018a.

Behnam Neyshabur, Zhiyuan Li, Srinadh Bhojanapalli, Yann LeCun, and Nathan Srebro. The role of over-parametrization in generalization of neural networks. In *International Conference on Learning Representations*, 2018b.

Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in Neural Information Processing Systems*, 32: 8026–8037, 2019.

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

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, 2018.

Yury Polyanskiy and Yihong Wu. Lecture notes on information theory. 2019.

Maxim Raginsky and Igal Sason. Concentration of measure inequalities in information theory, communications and coding. *Foundations and Trends in Communications and Information Theory; NOW Publishers: Boston, MA, USA*, 2018.

Borja Rodríguez-Gálvez, Germán Bassi, Ragnar Thobaben, and Mikael Skoglund. On random subset generalization error bounds and the stochastic gradient langevin dynamics algorithm. *arXiv preprint arXiv:2010.10994*, 2020.

Nicolas Le Roux, Mark Schmidt, and Francis Bach. A stochastic gradient method with an exponential convergence rate for finite training sets. In *Proceedings of the 25th International Conference on Neural Information Processing Systems-Volume 2*, pp. 2663–2671, 2012.

Daniel Russo and James Zou. Controlling bias in adaptive data analysis using information theory. In *Artificial Intelligence and Statistics*, pp. 1232–1240, 2016.

Daniel Russo and James Zou. How much does your data exploration overfit? controlling bias via information usage. *IEEE Transactions on Information Theory*, 66(1):302–323, 2019.

Shai Shalev-Shwartz and Shai Ben-David. *Understanding machine learning: From theory to algorithms*. Cambridge university press, 2014.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *3rd International Conference on Learning Representations, ICLR*, 2015.

Samuel L Smith, Benoit Dherin, David Barrett, and Soham De. On the origin of implicit regularization in stochastic gradient descent. In *International Conference on Learning Representations*, 2021.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

Thomas Steinke and Lydia Zakynthinou. Reasoning about generalization via conditional mutual information. In *Conference on Learning Theory*, volume 125 of *Proceedings of Machine Learning Research*, pp. 3437–3452. PMLR, 2020.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR*, pp. 2818–2826, 2016.

Valentin Thomas, Fabian Pedregosa, Bart Merriënboer, Pierre-Antoine Manzagol, Yoshua Bengio, and Nicolas Le Roux. On the interplay between noise and curvature and its effect on optimization and generalization. In *International Conference on Artificial Intelligence and Statistics*, pp. 3503–3513. PMLR, 2020.

Vladimir Vapnik. *Statistical learning theory*. Wiley, 1998. ISBN 978-0-471-03003-4.

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, 2021a.

Hao Wang, Yizhe Huang, Rui Gao, and Flavio P Calmon. Learning while dissipating information: Understanding the generalization capability of sgld. *arXiv preprint arXiv:2102.02976*, 2021b.

Colin Wei, Sham Kakade, and Tengyu Ma. The implicit and explicit regularization effects of dropout. In *International Conference on Machine Learning*, pp. 10181–10192. PMLR, 2020.

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)*, pp. 681–688. Citeseer, 2011.

Yeming Wen, Kevin Luk, Maxime Gazeau, Guodong Zhang, Harris Chan, and Jimmy Ba. An empirical study of stochastic gradient descent with structured covariance noise. In *International Conference on Artificial Intelligence and Statistics*, pp. 3621–3631. PMLR, 2020.

Lei Wu, Chao Ma, and E Weinan. How sgd selects the global minima in over-parameterized learning: A dynamical stability perspective. *Advances in Neural Information Processing Systems*, 2018: 8279–8288, 2018.

Aolin Xu and Maxim Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. *Advances in Neural Information Processing Systems*, 2017:2525–2534, 2017.

Zitong Yang, Yaodong Yu, Chong You, Jacob Steinhardt, and Yi Ma. Rethinking bias-variance trade-off for generalization of neural networks. In *International Conference on Machine Learning*, pp. 10767–10777. PMLR, 2020.

Zhewei Yao, Amir Gholami, Kurt Keutzer, and Michael W Mahoney. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE International Conference on Big Data (Big Data)*, pp. 581–590. IEEE, 2020.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In *5th International Conference on Learning Representations*. OpenReview.net, 2017.

Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR*, 2018.

Jingzhao Zhang, Tianxing He, Suvrit Sra, and Ali Jadbabaie. Why gradient clipping accelerates training: A theoretical justification for adaptivity. In *International Conference on Learning Representations*, 2019.

Yaowei Zheng, Richong Zhang, and Yongyi Mao. Regularizing neural networks via adversarial model perturbation. In *CVPR*, 2021.

Ruida Zhou, Chao Tian, and Tie Liu. Individually conditional individual mutual information bound on generalization error. *arXiv preprint arXiv:2012.09922*, 2020.

Appendix

Table of Contents

A HWI Inequality: Proof of Lemma 3 **15**

B A Key Lemma: Lemma 4 **16**

C Improved Mutual Information Bounds: Proof of Theorem 1 **16**
- C.1 Lemma 5 . . . 16
- C.2 Proof of Theorem 1 . . . 16
- C.3 Corollary 1 . . . 18

D Application in Neural Networks: Proof of Theorems 2 and 3 **18**
- D.1 Lemma 6 . . . 18
- D.2 Proof of Theorem 2 . . . 19
- D.3 Proof of Theorem 3 . . . 19

E Experiment Details **20**
- E.1 Architectures and Hyperparameters . . . 20
- E.2 Learning Rate and Batch Size. . . . 20
- E.3 Implementation Details of Figure 2 . . . 21
- E.4 Algorithm of Dynamic Gradient Clipping and Additional Results . . . 21
- E.5 Discussion on Gradient Dispersion of Models Trained on Clean Datasets . . . 22
- E.6 Algorithm of Gaussian Model Perturbation and Additional Results . . . 22

A HWI Inequality: Proof of Lemma 3

Proof. The proof given here is a simple extension of the proof of Raginsky & Sason (2018, Lemma 3.4.2), which is a special instance of the (weak) HWI inequality.

$$
\begin{aligned}
\mathrm{D_{KL}}(P_{X+\sqrt{t}N}||P_{Y+\sqrt{t}N}) \leq & \mathrm{D_{KL}}(P_{X,Y,X+\sqrt{t}N}||P_{X,Y,Y+\sqrt{t}N}) && (1)\\
= & \mathop{\mathbb{E}}_{X,Y}\left[\mathrm{D_{KL}}(P_{X+\sqrt{t}N|X,Y}||P_{Y+\sqrt{t}N|X,Y})\right] \\
= & \mathop{\mathbb{E}}_{X,Y}\left[\mathrm{D_{KL}}(\mathcal{N}(X,t)||\mathcal{N}(Y,t))|X,Y\right] && (2)\\
= & \frac{1}{2t}\mathop{\mathbb{E}}_{X,Y}\left[||X-Y||^2\right], && (3)
\end{aligned}
$$

where Eq.1 is by the chain rule of the KL divergence, Eq.2 holds since N is independent of (X, Y) and Eq.3 is a special case of the equality

$$
\mathcal{N}(\mu_1,\Sigma_1)||\mathcal{N}(\mu_2,\Sigma_2)) = \frac{1}{2}\left[\log\frac{|\Sigma_2|}{|\Sigma_1|} - d + \mathrm{Tr}\left(\Sigma_2^{-1}\Sigma_1\right) + (\mu_2-\mu_1)^T\Sigma_2^{-1}(\mu_2-\mu_1)\right].
$$

Eq.3 holds for any joint distribution of (X, Y), so by the definition of Wasserstein distance, we have

$$
\mathrm{D_{KL}}(P_{X+\sqrt{t}N}||P_{Y+\sqrt{t}N}) \leq \frac{1}{2t}\mathbb{W}_2^2(P_X,P_Y). \tag{4}
$$

This completes the proof. □

B A KEY LEMMA: LEMMA 4

Proof. By the properties of mutual information,

$$I(f(Z,X)+\sigma N; X|Y) = I(\frac{f(Z,X)-\mathbb{E}\left[f(Z,X)\right]}{\sigma}+N; X|Y),$$

where the expectation is taken over $(Z,X)\sim P_{Z,X}$.

Let $g(Z,X)\triangleq \frac{f(Z,X)-\mathbb{E}[f(Z,X)]}{\sigma}$. Then,

$$\begin{aligned}
I(g(Z,X)+N;X|Y) =& \mathop{\mathbb{E}}_{Y,X}\left[\mathrm{D_{KL}}\left(P_{g(Z,x)+N|X=x,Y=y}||P_{g(Z,X)+N|Y=y}\right)\right] \\
=& \mathop{\mathbb{E}}_{Y,X}\left[\mathrm{D_{KL}}\left(P_{g(Z,x)+N|X=x,Y=y}||P_N\right)-\mathrm{D_{KL}}\left(P_{g(Z,X)+N|Y=y}||P_N\right)\right] \quad (5)\\
\leq& \mathop{\mathbb{E}}_{Y,X}\left[\mathrm{D_{KL}}\left(P_{g(Z,x)+N|X=x,Y=y}||P_{\Omega+N}\right)\right] \quad (6)\\
\leq& \frac{1}{2}\mathop{\mathbb{E}}_{Y,X}\left[\mathbb{W}_2^2\left(P_{g(Z,x)|X=x,Y=y}||P_{\Omega}\right)\right] \quad (7)\\
=& \frac{1}{2}\mathop{\mathbb{E}}_{Y,X}\left[\mathop{\mathbb{E}}_{\Delta}\left[||g(y+\Delta,x)||^2|X=x,Y=y\right]\right]\\
=& \frac{1}{2\sigma^2}\mathop{\mathbb{E}}_{Z,X}\left[||f(Z,X)-\mathop{\mathbb{E}}_{Z,X}\left[f(Z,X)\right]||^2\right]
\end{aligned}$$

where Eq. 5 is the golden formula (see Theorem 3.3 in Polyanskiy & Wu (2019)), Eq. 6 is by the non-negativity of relative entropy, and Ω is a random variable that equals to zero with probability one, Eq. 7 is due to Lemma 3. □

C IMPROVED MUTUAL INFORMATION BOUNDS: PROOF OF THEOREM 1

In this section, we provide a proof for Theorem 1, while elaborating on why the local gradient sensitivity term in Lemma 2 can be removed. The key ingredient that makes this possible is a construction given in the proof of Lemma 5.

C.1 LEMMA 5

Proof. We first notice that

$$I(G_t+N_t;S|\widetilde{W}_{t-1}) = I(-\lambda_t g(W_{t-1},B_t)+\sigma_t N; S|\widetilde{W}_{t-1}),$$

where $N\sim\mathcal{N}(0,I_d)$.

Then by applying Lemma 4, we have

$$\begin{aligned}
I(-\lambda_t g(W_{t-1},B_t)+\sigma_t N; S|\widetilde{W}_{t-1}) \leq& \frac{\lambda_t^2}{2\sigma_t^2}\mathbb{E}\left[||g(W_{t-1},B_t)-\mathbb{E}\left[\nabla_w \ell(W_{t-1},Z)\right]||^2\right]\\
=& \frac{\lambda_t^2}{2\sigma_t^2}\mathbb{E}\left[\mathbb{V}_{t-1}(W_{t-1})\right]
\end{aligned}$$

This completes the proof. □

C.2 PROOF OF THEOREM 1

Proof. We first follow the similar decomposition of the expected generalization error and apply Lemma 1 as in Neu (2021),

$$\begin{aligned}
|\mathrm{gen}(\mu,P_{W_T|S})| =& \left|\mathrm{gen}(\mu,P_{\widetilde{W}_T|S})+\mathop{\mathbb{E}}_{W_T,\Delta_T}\left[L_\mu(W_T)-L_\mu(\widetilde{W}_T)\right]+\mathop{\mathbb{E}}_{W_T,\Delta_T,S}\left[L_S(\widetilde{W}_T)-L_S(W_T)\right]\right|\\
\leq& \sqrt{\frac{2R^2}{n}I(\widetilde{W}_T;S)}+\left|\mathop{\mathbb{E}}_{W_T,S,S'}\left[\gamma(W_T,S)-\gamma(W_T,S')\right]\right|. \quad (8)
\end{aligned}$$

The remaining task is to bound the mutual information $I(\widetilde{W}_T; S)$.

Then,

$$
\begin{aligned}
& I(\widetilde{W}_T; S) \\
=& I\left(\widetilde{W}_{T-1} - \lambda_T g(W_{T-1}, B_T) + N_T; S\right) \\
\leq& I\left(\widetilde{W}_{T-1}, -\lambda_T g(W_{T-1}, B_T) + N_T, ; S\right) && (9) \\
=& I(\widetilde{W}_{T-1}; S) + I\left(-\lambda_T g(W_{T-1}, B_T) + N_T; S | \widetilde{W}_{T-1}\right) && (10) \\
\leq& I(\widetilde{W}_{T-2}; S) + I\left(-\lambda_{T-1} g(W_{T-2}, B_{T-1}) + N_{T-1}; S | \widetilde{W}_{T-2}\right) + I\left(-\lambda_T g(W_{T-1}, B_T) + N_T; S | \widetilde{W}_{T-1}\right) && (11) \\
& \vdots \\
\leq& I(\widetilde{W}_0; S) + \sum_{t=1}^{T} I\left(-\lambda_t g(W_{t-1}, B_t) + N_t; S | \widetilde{W}_{t-1}\right), && (12) \\
=& \sum_{t=1}^{T} I\left(-\lambda_t g(W_{t-1}, B_t) + N_t; S | \widetilde{W}_{t-1}\right), && (13)
\end{aligned}
$$

where Eq.9 is by $I(f(X); Y) \leq I(X; Y)$, Eq.10 is by the chain rule of mutual information, Eq.11 is by applying the similar procedure (namely, Eq.9-Eq.10) to $I(\widetilde{W}_{T-1}; S)$, Eq. 12 is by doing these steps recursively and Eq.13 is due to the fact that $\widetilde{W}_0$ is independent of S (i.e. $I(\widetilde{W}_0; S) = 0$). Applying Lemma 5 and putting everything together, we have

$$
|\mathrm{gen}(\mu, P_{W_T|S})| \leq \sqrt{\frac{R^2}{nb} \sum_{t=1}^{T} \frac{\lambda_t^2}{\sigma_t^2} \mathbb{E}\left[\widetilde{\mathbb{V}}_t(W_{t-1})\right]} + \left|\mathbb{E}\left[\gamma(W_T, S) - \gamma(W_T, S')\right]\right|.
$$

Next, to handle the mismatch between the outputs of perturbed SGD and SGD, we apply Taylor expansion around $\Delta_T = \vec{0}$,

$$
\begin{aligned}
\mathop{\mathbb{E}}_{W_T, S, \Delta_T}\left[L_S(W_T + \Delta_T) - L_S(W_T)\right] &= \frac{1}{n} \sum_{i=1}^{n} \mathop{\mathbb{E}}_{W_T, Z_i, \Delta_T}\left[\ell(W_T + \Delta_T, Z_i) - \ell(W_T, Z_i)\right] \\
&\approx \mathop{\mathbb{E}}_{W_T, Z, \Delta_T}\left[\langle \nabla_w \ell(W_T, Z), \Delta_T \rangle + \frac{1}{2} \Delta_T^T \mathrm{H}_{W_T}(Z) \Delta_T\right] \\
&= \mathop{\mathbb{E}}_{W_T, Z, \Delta_T}\left[\frac{1}{2} \Delta_T^T \mathrm{H}_{W_T}(Z) \Delta_T\right] && (14) \\
&= \frac{1}{2} \langle \mathop{\mathbb{E}}_{W_T, Z}\left[\mathrm{H}_{W_T}(Z)\right], \mathop{\mathbb{E}}_{\Delta_T}\left[\Delta_T \Delta_T^T\right] \rangle \\
&= \frac{1}{2} \langle \mathop{\mathbb{E}}_{W_T, Z}\left[\mathrm{H}_{W_T}(Z)\right], \mathrm{diag}(\sum_{t=1}^{T} \sigma_t^2) \rangle && (15) \\
&= \frac{\sum_{t=1}^{T} \sigma_t^2}{2} \mathrm{Tr}(\mathop{\mathbb{E}}_{W_T, Z}\left[\mathrm{H}_{W_T}(Z)\right]),
\end{aligned}
$$

where Eq.14 is by the zero mean of the perturbation, Eq.15 is by the independence of the coordinates of Δ_T, $\langle \cdot, \cdot \rangle$ denotes the inner product of two matrices, $\mathrm{diag}(A)$ is the diagonal matrix with element A and $\mathrm{Tr}(\cdot)$ is the trace of a matrix.

Under the condition $\mathop{\mathbb{E}}_{W_T,S'}[\gamma(W_T,S')] \geq 0$, we now bound $\mathrm{gen}(\mu, \mathrm{P}_{\widetilde{W}_T|S})$ instead of its absolute value, $|\mathrm{gen}(\mu, \mathrm{P}_{\widetilde{W}_T|S})|$. The following is straightforward,

$$\begin{aligned}
\mathrm{gen}(\mu, \mathrm{P}_{\widetilde{W}_T|S}) \leq & \frac{R}{n}\sqrt{\sum_{t=1}^{T}\frac{\lambda_t^2}{\sigma_t^2}\mathop{\mathbb{E}}_{W_{t-1}}[\mathbb{V}(W_{t-1})]} + \mathop{\mathbb{E}}_{W_T,S,S'}[\gamma(W_T,S)-\gamma(W_T,S')] \\
\leq & \frac{R}{n}\sqrt{\sum_{t=1}^{T}\frac{\lambda_t^2}{\sigma_t^2}\mathop{\mathbb{E}}_{W_{t-1}}[\mathbb{V}(W_{t-1})]} + \mathop{\mathbb{E}}_{W_T,S}[\gamma(W_T,S)] \\
= & \frac{R}{n}\sqrt{\sum_{t=1}^{T}\frac{\lambda_t^2}{\sigma_t^2}\mathop{\mathbb{E}}_{W_{t-1}}[\mathbb{V}(W_{t-1})]} + \frac{\sum_{t=1}^{T}\sigma_t^2}{2}\mathrm{Tr}(\mathop{\mathbb{E}}_{W_T,Z}[\mathrm{H}_{W_T}(Z)]).
\end{aligned}$$

Since every choice of σ gives a valid generalization bound. The optimal bound can be found by simply utilizing the fact $A/\sigma+\sigma^2 B \geq 3(A/2)^{2/3}B^{1/3}$ for any positive A and B, where the equality is achieved by the optimal σ. Finally, rearranging the terms will complete the proof. □

C.3 Corollary 1

Corollary 1. *If the loss function is differentiable and β-smooth with respect to w, then,*

$$|\mathrm{gen}(\mu, P_{W_T|S})| \leq \sqrt{\frac{R^2}{n}\sum_{t=1}^{T}\frac{\lambda_t^2}{\sigma_t^2}\mathbb{E}\left[\mathbb{V}_t(W_{t-1})\right]} + \beta d\sum_{t=1}^{T}\sigma_t^2.$$

Proof. Recall the smoothness implies $f(\mathbf{v}) \leq f(\mathbf{w}) + \langle \nabla f(\mathbf{w}), \mathbf{v}-\mathbf{w}\rangle + \frac{\beta}{2}||\mathbf{v}-\mathbf{w}||^2$ for all $\mathbf{v}$ and $\mathbf{w}$. By the triangle inequality, we have

$$|\mathbb{E}\left[L_\mu(W_T) - L_\mu(W_T+\Delta_T)\right]| \leq |\mathbb{E}\left[\langle \nabla_w \ell(W_T,Z), \Delta_T\rangle\right]| + \frac{\beta}{2}\mathbb{E}\left[||\Delta_T||^2\right] = \frac{\beta d\sum_{t=1}^{T}\sigma_t^2}{2}$$

Thus, we can see that $|\mathbb{E}\left[L_\mu(W_T) - L_\mu(W_T+\Delta_T)\right]| + |\mathbb{E}\left[L_S(W_T+\Delta_T) - L_S(W_T)\right]| \leq \beta d\sum_{t=1}^{T}\sigma_t^2$.

This completes the proof. □

Remark 4. *In Corollary 1, we note that the dependence of d in the bound results from the spherical Gaussian noise used in the construction of the weight process $\widetilde{W}_T$. It is possible to replace the spherical Gaussian with a Gaussian noise having a non-diagonal covariance that reflects the geometry of the loss landscape. With this replacement, the dimension d in the flatness term will be replaced by the trace of $\sum_{t=1}^{T}\kappa_t$, where κ_t is the covariance matrix of the noise added at step t. Please refer to Neu et al. (2021) for a similar development.*

D Application in Neural Networks: Proof of Theorems 2 and 3

D.1 Lemma 6

Lemma 6. *Under the same conditions of Theorem 1, the generalization error of SGD is upper bounded by*

$$\mathrm{gen}(\mu, P_{W_T|S}) \leq \frac{3}{2}\left(\sum_{t=1}^{T}\frac{R^2\lambda_t^2 T}{n}\mathbb{E}\left[||g(W_{t-1},B_t)||_2^2\right]\mathbb{E}\left[\mathrm{Tr}\left(\mathrm{H}_{W_T}(Z)\right)\right]\right)^{\frac{1}{3}}$$

Proof. The proof of this lemma follow the same steps in the proof of Theorem 1 except that we require a modified version of Lemma 4 here.

Specially, we utilize $I(f(Z,X)+\sigma N;X|Y) = I(\frac{f(Z,X)}{\sigma}+N;X|Y)$ instead of $I(f(Z,X)+\sigma N;X|Y) = I(\frac{f(Z,X)-\mathbb{E}[f(Z,X)]}{\sigma}+N;X|Y)$, that is used in the proof of Lemma 4. The remaining steps should be straightforward. □

Remark 5. *Intuitively, the bound in Lemma 6 is looser than Eq. 2 in Theorem 1 as the centralized expected vector norm should be smaller than the original expected vector norm.*

D.2 Proof of Theorem 2

Proof. By Cauchy–Schwarz inequality, we have

$$\mathbb{E}\left[||g(W_{t-1},B_t)||_2^2\right] = \mathbb{E}\left[\left|\left|\frac{1}{b}\sum_{Z_i\in B_t}\nabla_w\ell(W_{t-1},Z_i)\right|\right|_2^2\right] \leq \mathbb{E}\left[||\nabla_w\ell(W_{t-1},Z)||_2^2\right] \tag{16}$$

Notice that $\nabla_w\ell(W,Z) = (W^TX-Y)X$. Let $\hat{Y} = f(W,X)$. Then,

$$||\nabla_w\ell(W_{t-1},Z)||_2^2 = ||(W_{t-1}^TX-Y)X||_2^2 \leq ||W_{t-1}^TX-Y||_2^2 = (\hat{Y}-Y)^2 = 2\ell(W_{t-1},Z) \tag{17}$$

For the Hessian matrix, it's easy to see that $\mathrm{Tr}(\mathrm{H}_{W_T}(Z)) = 1$.

Plugging everything into Lemma 6, we have

$$\mathrm{gen}(\mu, P_{W_T|S}) \leq 3\left(\sum_{t=1}^{T}\frac{R^2\lambda_t^2T}{4n}\mathop{\mathbb{E}}_{W_{t-1},Z}\left[\ell(W_{t-1},Z)\right]\right)^{\frac{1}{3}}$$

This completes the proof. □

D.3 Proof of Theorem 3

Proof. Since $\nabla_{w_r}\ell(W,Z_i) = \frac{1}{\sqrt{m}}A_r(f(W,X_i)-Y_i)X_i\mathbb{I}_{r,i}$, where $\mathbb{I}_{r,i} = \mathbb{I}\{W_r^TX_i \geq 0\}$, we have

$$\begin{aligned}
&||\nabla_w\ell(W_{t-1},Z)||_2^2 && (18)\\
=&\sum_{r=1}^{m}||\frac{1}{\sqrt{m}}A_r(f(W_{t-1},X_i)-Y_i)X_i\mathbb{I}_{r,i}||_2^2\\
\leq&\sum_{r=1}^{m}||\frac{1}{\sqrt{m}}A_r(f(W_{t-1},X_i)-Y_i)\mathbb{I}_{r,i}||_2^2\\
=&\frac{1}{m}\sum_{r=1}^{m}\mathbb{I}_{r,i}||(f(W_{t-1},X_i)-Y_i)||_2^2\\
=&\frac{1}{m}\sum_{r=1}^{m}2\mathbb{I}_{r,i}\ell(W_{t-1},Z_i) && (19)
\end{aligned}$$

In addition, we notice that $\mathrm{Tr}(\mathrm{H}_{W_T}(Z)) = \frac{1}{m}\sum_{r=1}^{m}\mathbb{I}_{r,i,T}$. Plugging everything into Lemma 6, we have

$$\mathrm{gen}(\mu, P_{W_T|S}) \leq 3\left(\sum_{t=1}^{T}\frac{R^2\lambda_t^2T}{4nm}\mathop{\mathbb{E}}_{W_{t-1},Z_i}\left[\sum_{r=1}^{m}\mathbb{I}_{r,i,t}\ell(W_{t-1},Z_i)\right]\mathop{\mathbb{E}}_{W_T,Z_i}\left[\frac{1}{m}\sum_{r=1}^{m}\mathbb{I}_{r,i,T}\right]\right)^{\frac{1}{3}}$$

This completes the proof. □

E EXPERIMENT DETAILS

E.1 ARCHITECTURES AND HYPERPARAMETERS

In Section 5, MLP has one hidden layer with 512 hidden units, and AlexNet has five convolution layers (conv. 3×3 (64 filters) $\rightarrow$ max-pool 3×3 $\rightarrow$ conv. 5×5 (192 filters) $\rightarrow$ max-pool 3×3 $\rightarrow$ conv. 3×3 (384 filters) $\rightarrow$ conv. 3×3 (256 filters) $\rightarrow$ conv. 3×3 (256 filters) $\rightarrow$ max-pool 3×3) followed by two fully connected layers both with 4096 units and a 10-way linear layer as the output layer. All of the convolution layers and the fully connected layers use standard rectified linear activation functions (ReLU).

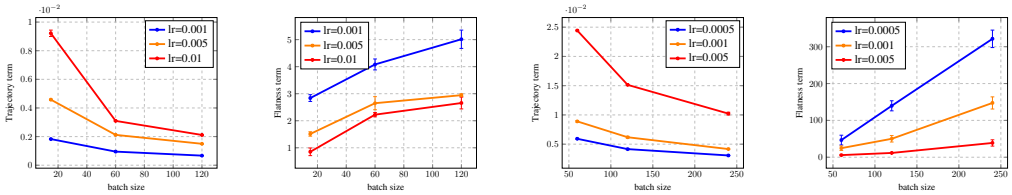
The fixed learning rates used for MLP and AlexNet are 0.01 and 0.001, respectively. The batch size is set to 60. For the corrupted label experiment, we train the models until the models achieve 100% training accuracy. For other cases, we train the neural networks until the training loss converges (e.g., < 0.0001). Other settings are either described in Section 5 or apparent in the figures. Standard techniques such as weight decay and batch normalization are not used.

In Section 6, we compare GMP with other advanced regularization methods. The results of other methods are reported directly from Zheng et al. (2021), and we now give their hyperparameter settings here for completeness. For Dropout, 10% of neurons are randomly selected to be deactivated in each layer. For label smoothing, the coefficient is 0.2. For flooding, the level is set to 0.02. For MixUp, we lineally combine random pairs of training data where the coefficient is drawn from $\mathrm{Beta}(1, 1)$. For adversarial training, the perturbation size is 1 for each pixel and we take one step to generate adversarial examples. For AMP, the number of inner iteration is 1, and the L_2 norm ball radius values are 0.5 for PreActResNet18 and 0.1 for VGG16, respectively.

The implementation in this paper is on PyTorch, and all the experiments are carried out on NVIDIA Tesla V100 GPUs (32 GB).

E.2 LEARNING RATE AND BATCH SIZE.

The learning rate and batch size have some direct impact on Eq. 2 in Theorem 1. We now investigate this by performing experiments with varying learning rates and batch sizes. In our experiments, the model is continuously updated until the average training loss drops below 0.0001. We separate trajectory and flatness terms of the bound and plot them in Figure 6.



(a) MLP on MNIST (b) MLP on MNIST (c) AlexNet on CIFAR10 (d) AlexNet on CIFAR10

Figure 6: The impact of learning rate and batch size on the trajectory term and the flatness term in Eq. 2

A key observation in Figure 6 is that the learning rate impacts the trajectory term and the flatness term in opposite ways, as seen, for example, in (a) and (b), where the two set of curves swap their orders in the two figures. On the other hand, the batch size also impacts the two terms in opposite ways, as seen in (a) and (b) where curves decrease in (a) but increase in (b). This makes the generalization bound, i.e., the sum of the two terms, have a rather complex relationship with the settings of learning rate and batch size. This relationship is further complicated by the fact that a small learning rate requires a longer training time, or a larger number T of training iterations, which increases the number that are summed over in the trajectory term. Nonetheless, we do observe that a smaller batch size gives a lower value of the flatness term ((b) and (d)), confirming the previous wisdom that small batch sizes enable the neural network to find a flat minima (Keskar et al., 2017).

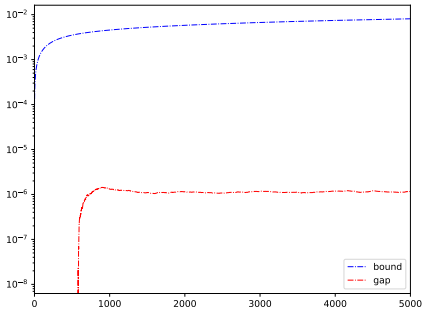


Figure 7: Generalization gap of a linear network v.s. Theorem 2. Note y-axe is log-scale.

E.3 Implementation Details of Figure 2

We let $d_0 = 200$ and use a two-layer ReLU network with hidden units 10000 to generate Y. Moreover, we apply a $\tanh$ function to the output of this network so that $|Y| \leq 1$. The input data $X \sim \mathcal{N}(0, \mathrm{I})$ and we normalize X before training so that $||X|| = 1$. For the training phase, we choose $m = 100$, let data size be $20,000$, batch size be 100 and learning rate be 0.5. Figure 7 shows the similar result of Theorem 2.

E.4 Algorithm of Dynamic Gradient Clipping and Additional Results

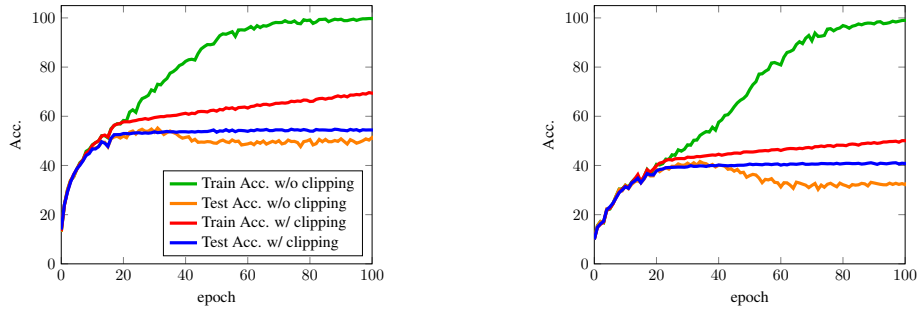
The dynamic gradient clipping algorithm is described in Algorithm 1. For both MLP and AlexNet, we let $\alpha = 0.1$. The start step for clipping, T_c, is also an important hyperparameter. However, it can be removed by detecting the evolution of the average gradient norm for each epoch. Specifically, whenever the average gradient norm of epoch j is larger than the average gradient norm of epoch $j-1$, the clipping operation begins.

Algorithm 1 Dynamic Gradient Clipping

Require: Training set S, Batch size b, Loss function ℓ, Initial model parameter $\boldsymbol{w}_0$, Learning rate λ, Initial minimum gradient norm $\mathcal{G}$, Number of iterations T, Clipping parameter α, Clipping step T_c

for $t \leftarrow 1$ to T **do**
  Sample $\mathcal{B} = \{\boldsymbol{z}_i\}_{i=1}^{b}$ from training set S
  Compute gradient:
  $g_{\mathcal{B}} \leftarrow \sum_{i=1}^{b} \nabla_{\boldsymbol{w}} \ell(\boldsymbol{w}_{t-1}, \boldsymbol{z}_i)/b$
  if $t > T_c$ **then**
    if $||g_{\mathcal{B}}||_2 > \mathcal{G}$ **then**
      $g_{\mathcal{B}} \leftarrow \alpha \cdot \mathcal{G} \cdot g_{\mathcal{B}}/||g_{\mathcal{B}}||_2$
    else
      $\mathcal{G} \leftarrow ||g_{\mathcal{B}}||_2$
    end if
  end if
  Update parameter: $\boldsymbol{w}_t \leftarrow \boldsymbol{w}_{t-1} - \lambda \cdot g_{\mathcal{B}}$
end for

From Figure 5 and Figure 8 we can see that dynamic gradient clipping effectively alleviates overfitting by conspicuously slowing down the transition of training to the memorization regime, without changing the convergence speed of testing accuracy. Unfortunately, the current design of the dynamic gradient clipping algorithm does not provide a significant improvement for models trained on a clean dataset (without label noise). Designing better regularization algorithms may require understanding the epoch-wise double descent curve of gradient dispersion where the model is trained on a clean dataset.



(a) noise=0.2 (CIFAR10)

(b) noise=0.4 (CIFAR10)

Figure 8: Dynamic Gradient Clipping (AlexNet).

E.5 Discussion on Gradient Dispersion of Models Trained on Clean Datasets

In the case of no noise injected, Figure 4a shows that the model with good generalization property has a exponentially-decaying gradient dispersion. This is consistent with our discussion of Lemma 5 in Section 3, that is, small $I(G_t + N_t; S|\widetilde{W}_{t-1})$ indicates good generalization. Notably, gradient dispersion of AlexNet trained on the real CIFAR10 data still has a epoch-wise double descent curve. The difference between Figure 4e with Figure 4f-4h is that the testing accuracy does not decrease in the second phase/memorization regime for AlexNet trained on the true CIFAR10 data. Loosely speaking, we conjugate that memorizing random labels will hurt the performance on unseen clean data but memorizing clean (or true) labels will not. This may explain why dynamic gradient clipping or preventing the training entering the memorization regime cannot improve the performance on a clean dataset.

E.6 Algorithm of Gaussian Model Perturbation and Additional Results

Algorithm 2 Gaussian Model Perturbation Training

Require: Training set S, Batch size b, Loss function ℓ, Initial model parameter $\boldsymbol{w}_0$, Learning rate λ, Number of noise k, Standard deviation of Gaussian distribution σ, Lagrange multiplier ρ

while $\boldsymbol{w}_t$ not converged **do**

Update iteration: $t \leftarrow t + 1$

Sample $\mathcal{B} = \{\boldsymbol{z}_i\}_{i=1}^b$ from training set S

Sample $\Delta_j \sim \mathcal{N}(0, \sigma^2)$ for $j \in [k]$

Compute gradient:

$g_{\mathcal{B}} \leftarrow \sum_{i=1}^b \left(\nabla_{\boldsymbol{w}} \ell(\boldsymbol{w}_t, z_i) + \rho \sum_{j=1}^k \left(\nabla_{\boldsymbol{w}} \ell(\boldsymbol{w}_t + \Delta_j, z_i) - \nabla_{\boldsymbol{w}} \ell(\boldsymbol{w}_t, z_i) \right) / k \right) / b$

Update parameter: $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t - \lambda \cdot g_{\mathcal{B}}$

end while

The GMP algorithm is given in Algorithm 2. Table 1 and Table 2 show that our method is competitive to the state-of-the-art regularization techniques. Specifically, our method achieves the best performance on SVHN for both models and on CIFAR-100 where VGG16 is employed. Particularly, testing accuracy is improved by nealy 2% on the CIFAR-100 dataset with VGG16. For other tasks, GMP is always able to achieve the top-3 performance. In addition, we find that increasing the number of sampled noises does not guarantee the improvement of testing accuracy and may even degrade the performance on some datasets (e.g., SVHN). This hints that we can use small number of noises to reduce the running time without losing performance. Moreover, we observe that **GMP with $k = 3$ usually takes around $1.76\times$ that of ERM training time**, which is affordable.

For PreActResNet18, the performance of GMP appears slighly inferior to the most recent record given by AMP (Zheng et al., 2021). Noting that the key ingredient of AMP, "max-pooling" in the parameter space, greatly resembles regularization term in GMP, which may be seen as "average-pooling" in the same space.

Method	SVHN	CIFAR-10	CIFAR-100
ERM	97.05±0.063	94.98±0.212	75.69±0.303
Dropout	97.20±0.065	95.14±0.148	75.52±0.351
L.S.	97.22±0.087	95.15±0.115	77.93±0.256
Flooding	97.16±0.047	95.03±0.082	75.50±0.234
MixUp	97.26±0.044	95.91±0.117	78.22±0.210
Adv. Tr.	97.23±0.080	95.01±0.085	74.77±0.229
AMP	**97.70±0.025**	**96.03±0.091**	**78.49±0.308**
GMP^3	97.43±0.037	95.64±0.053	78.05±0.208
GMP^{10}	97.34±0.058	95.71±0.073	78.07±0.170

Table 2: Top-1 classification accuracy acc.(%) of PreActResNet18. We run experiments 10 times and report the mean and the standard deviation of the testing accuracy. Superscript denotes the number of sampled Gaussian noises during training.

Method	SVHN	CIFAR-10	CIFAR-100
GMP$^3_{\text{abs}}$	97.10±0.054	94.21±0.139	74.80±0.113

Table 3: Top-1 classification accuracy acc.(%) of VGG16.

One potential extension of GMP is letting the variance of the noise distribution be a function of the iteration step t. In other words, using the time-dependent σ_t instead of a constant σ.

We also consider the following regularized scheme, which is a absolute value version.

$$\min_w L_s(w) + \rho \mathop{\mathbb{E}}_{\Delta \sim \mathcal{N}(0, \sigma^2 \mathbf{I}_d)} \Big[\Big| L_s(w + \Delta) - L_s(w) \Big| \Big].$$

This scheme can still perform well as shown in Table 3. In fact, it outperforms GMP^3 on CIFAR-100. This hints that it is possible to improve the performance by choosing other norm of $L_s(w + \Delta) - L_s(w)$.