Towards Understanding Sharpness-Aware Minimization

Maksym Andriushchenko [1] **Nicolas Flammarion** [1]

Abstract

Sharpness-Aware Minimization (SAM) is a recent training method that relies on worst-case weight perturbations which significantly improves generalization in various settings. We argue that the existing justifications for the success of SAM which are based on a PAC-Bayes generalization bound and the idea of convergence to flat minima are incomplete. Moreover, there are no explanations for the success of using m-sharpness in SAM which has been shown as essential for generalization. To better understand this aspect of SAM, we theoretically analyze its implicit bias for diagonal linear networks. We prove that SAM always chooses a solution that enjoys better generalization properties than standard gradient descent for a certain class of problems, and this effect is amplified by using m-sharpness. We further study the properties of the implicit bias on non-linear networks empirically, where we show that fine-tuning a standard model with SAM can lead to significant generalization improvements. Finally, we provide convergence results of SAM for non-convex objectives when used with stochastic gradients. We illustrate these results empirically for deep networks and discuss their relation to the generalization behavior of SAM. The code of our experiments is available at https://github.com/tml-epfl/understanding-sam.

1. Introduction

Understanding generalization of overparametrized deep neural networks is a central topic of machine learning. Training objective has many global optima where the training data are perfectly fitted (Zhang et al., 2017), but different global optima lead to dramatically different generalization performance (Liu et al., 2019). However, it has been observed that stochastic gradient descent (SGD) tends to converge to well-generalizing solutions, even *without* any explicit regularization methods (Zhang et al., 2017). This suggests that the leading role is played by the *implicit* bias of the optimization algorithms used (Neyshabur et al., 2015): when the training objective is minimized using a particular algorithm and initialization method, it converges to a specific solution with favorable generalization properties. However, even though SGD has a very beneficial implicit bias, significant overfitting can still occur, particularly in the presence of label noise (Nakkiran et al., 2020) and adversarial perturbations (Rice et al., 2020).

Recently it has been observed that the *sharpness* of the training loss, i.e., how quickly it changes in some neighborhood around the parameters of the model, correlates well with the generalization error (Keskar et al., 2016; Jiang et al., 2019), and generalization bounds related to the sharpness have been derived (Dziugaite & Roy, 2018). The idea of minimizing the sharpness to improve generalization has motivated recent works of Foret et al. (2021), Zheng et al. (2021), and Wu et al. (2020) which propose to use worst-case perturbations of the weights on every iteration of training in order to improve generalization. We refer to this method as *Sharpness-Aware Minimization* (SAM) and focus mainly on the version proposed in Foret et al. (2021) that performs only one step of gradient ascent to approximately solve the weight perturbation problem before updating the weights.

Despite the fact that SAM significantly improves generalization in various settings, the existing justifications based on the generalization bounds provided by Foret et al. (2021) and Wu et al. (2020) do not seem conclusive. The main reason is that their generalization bounds do not distinguish the robustness to *worst-case* weight perturbation from *average-case* robustness to Gaussian noise. However the latter does not sufficiently improve generalization as both Foret et al. (2021) and Wu et al. (2020) report. Furthermore, their analysis does not distinguish whether the worst-case weight perturbation is computed based on some or on all training examples. As we will discuss, this feature has a crucial impact on generalization.

In our paper, we aim to further investigate the reasons for SAM's success and make the following contributions:

- We discuss why the current understanding of the suc-

[1]EPFL, Switzerland. Correspondence to: Maksym Andriushchenko <maksym.andriushchenko@epfl.ch>.

Proceedings of the 39^{th} International Conference on Machine Learning, Baltimore, Maryland, USA, PMLR 162, 2022.

cess of SAM which is based on a PAC-Bayesian generalization bound and on convergence to a flatter minimum is incomplete.

- We test hypotheses regarding why maximization in SAM taken over *fewer* training points can lead to better generalization and conclude that the benefit is likely to come from the better objective.
- We study the implicit bias of this objective theoretically for diagonal linear networks. For non-linear networks, we study the implicit bias empirically and relate it to the theoretical model.
- We prove convergence of SAM for non-convex objectives in the stochastic setting. We check convergence empirically for deep networks and relate it to the generalization behavior of SAM.

2. Background on SAM

Related work. Here we discuss relevant works on robustness in the *weight space* and its relation to generalization. Works on weight-space robustness of neural networks date back at least to the 1990s (Murray & Edwards, 1993; Hochreiter & Schmidhuber, 1995). Random perturbations of the weights are used extensively in deep learning (Jim et al., 1996; Graves et al., 2013), and most prominently in approaches such as dropout (Srivastava et al., 2014). Many practitioners have observed that using SGD with larger batches for training leads to worse generalization (LeCun et al., 2012), and Keskar et al. (2016) have shown that this degradation of performance is *correlated* with the sharpness of the found parameters. This observation has motivated many further works which focus on closing the generalization gap between small-batch and large-batch SGD (Wen et al., 2018; Haruki et al., 2019; Lin et al., 2020). More recently, Jiang et al. (2019) have shown a strong correlation between the sharpness and the generalization error on a large set of models under a variety of different settings hyperparameters, beyond the batch size. This has motivated the idea of minimizing the sharpness during training to improve standard generalization, leading to Sharpness-Aware Minimization (SAM) (Foret et al., 2021). SAM modifies SGD such that on every iteration of training, the gradient is taken not at the current iterate but rather at a worst-case point in its vicinity. Zheng et al. (2021) concurrently propose a similar weight perturbation method which also successfully improves standard generalization on multiple deep learning benchmarks. Wu et al. (2020) have also proposed an almost identical algorithm with the same motivation, but with the focus on improving robust generalization of adversarial training. On the theoretical side, Mulayoff & Michaeli (2020) study the sharpness properties of minima of deep linear network, and Neu (2021); Wang & Mao (2022) study generalization bounds based on average-case sharpness and quantities related to the optimization trajectory of SGD.

Sharpness. Let $S_{train} = \{x_i, y_i\}_{i=1}^n$ be the training data and $\ell_i(w)$ be the loss of a classifier parametrized by weights $w \in \mathbb{R}^{|w|}$ and evaluated at point (x_i, y_i). Then the *sharpness* on a set of points $S \subseteq S_{train}$ is defined as:

$$s(w, S) \triangleq \max_{\|\delta\|_2 \le \rho} \frac{1}{|S|} \sum_{i:(x_i,y_i)\in S} \ell_i(w+\delta) - \ell_i(w). \quad (1)$$

In most of the past literature, sharpness is defined for $S = S_{train}$ (Keskar et al., 2016; Neyshabur et al., 2017; Jiang et al., 2019). However, Foret et al. (2021) recently introduced the notion of m-*sharpness* which is the average of the sharpness computed over all the batches S of size m from the training set S_{train}.

Lower sharpness is correlated with lower test error (Keskar et al., 2016), however, the correlation is not always perfect (Neyshabur et al., 2017; Jiang et al., 2019). Moreover, the sharpness definition itself can be problematic since rescaling of incoming and outcoming weights of a node that leads to the same function can lead to very different sharpness values (Dinh et al., 2017). Kwon et al. (2021) suggest a sharpness definition that fixes this rescaling problem but other problems still exist such as the sensitivity of classification losses to the scale of the parameters (Neyshabur et al., 2017).

Sharpness-aware minimization. Foret et al. (2021) theoretically base the SAM algorithm on the following objective:

$$\textbf{n-SAM:} \quad \min_{w\in\mathbb{R}^{|w|}} \max_{\|\delta\|_2 \le \rho} \sum_{i=1}^{n} \ell_i(w+\delta), \quad (2)$$

which we denote as **n-SAM** since it is based on maximization of the sum of the losses over the n training points. They justify this objective via a PAC-Bayesian generalization bound, although they show empirically (see Fig. 3 therein) that the following objective leads to better generalization:

$$\textbf{m-SAM:} \min_{w\in\mathbb{R}^{|w|}} \sum_{S\subseteq S_{train};\ |S|=m} \max_{\|\delta\|_2 \le \rho} \sum_{i\in S} \ell_i(w+\delta), \quad (3)$$

which we denote as **m-SAM** since it is based on maximization of the sum of the losses over batches of m training points and therefore related to the m-sharpness.

To make SAM practical, Foret et al. (2021) propose to minimize the m-SAM objective with stochastic gradients. Denoting the batch indices at time t by I_t ($|I_t| = m$), this leads to the following update rule on each iteration of training:

$$w_{t+1} = w_t - \frac{\gamma_t}{|I_t|} \sum_{i\in I_t} \nabla \ell_i\left(w_t + \frac{\rho_t}{|I_t|} \sum_{j\in I_t} \nabla \ell_j(w_t)\right). \quad (4)$$

Importantly, the *same* batch I_t is used for the inner and outer gradient steps. We note that ρ_t can optionally include

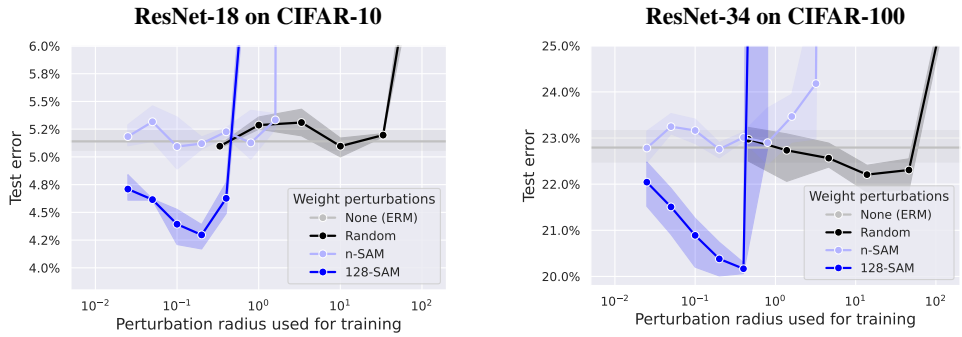


Figure 1: Comparison of different weight perturbation methods: no perturbations (ERM), random perturbations prior to taking the gradient on each iteration, n-SAM, and 128-SAM (see Sec. 2 for the notation). All models are trained with standard data augmentation and small batch sizes (128). We observe that among these methods only m-SAM with a low m (i.e., 128-SAM) substantially improves generalization.

the gradient normalization suggested in Foret et al. (2021), i.e., $\rho_t := \rho / \|\frac{1}{|I_t|} \sum_{j \in I_t} \nabla \ell_j(w_t)\|_2$. However, we show in Sec. 5 that its usage is not necessary for improving generalization, so we will omit it from our theoretical analysis.

Importance of low-m, worst-case perturbations. In order to improve upon ERM, Foret et al. (2021) use SAM with low-m and worst-case perturbations. To clearly illustrate the importance of these two choices, we show the performance of the following weight perturbation methods: no perturbations (ERM), random perturbations (prior to taking the gradient on each iteration), n-SAM, and 128-SAM. We use ResNet-18 on CIFAR-10 and ResNet-34 on CIFAR-100 (Krizhevsky & Hinton, 2009) with standard data augmentation and batch size 128 and refer to App. D for full experimental details, including our implementation of n-SAM. Fig. 1 clearly suggests that (1) the improvement from random perturbations is marginal, and (2) the only method that substantially improves generalization is low-m SAM (i.e., 128-SAM). Thus, worst-case perturbations and the use of m-sharpness in SAM are essential for the generalization improvement (which depends continuously on m as noted by Foret et al. (2021), see Fig. 16 in App. E.1). We also note that using too low m is inefficient in practice since it does not fully utilize the computational accelerators such as GPUs. Thus, using higher m values (such as 128) helps to balance the generalization improvement with the computational efficiency. Finally, we note that using SAM with large batch sizes without using a smaller m leads to suboptimal generalization (see Fig. 17 in App. E.2).

3. Challenging the Existing Understanding of SAM

In this section, we show the limitations of the current understanding of SAM. In particular, we discuss that the generalization bounds on which its only *formal* justification relies on (such as those presented in Foret et al. (2021); Wu et al. (2020); Kwon et al. (2021)) cannot explain its success. Second, we argue that contrary to a common belief, convergence of SAM to flatter minima measured in terms of m-sharpness does not always translate to better generalization.

The existing generalization bound does not explain the success of SAM. The main theoretical justification for SAM comes from the PAC-Bayesian generalization bound presented, e.g., in Theorem 2 of Foret et al. (2021). However, the bound is derived for *random* perturbations of the parameters, i.e. the leading term of the bound is $\mathbb{E}_{\mathcal{N}(0;\,)} \sum_{i=1}^{n} \ell_i(w + \delta)$. The extension to *worst-case* perturbations, i.e. $\max_{\|\delta\|_2} \sum_{i=1}^{n} \ell_i(w + \delta)$, is done post hoc and only makes the bound less tight. Moreover, we can see empirically (Fig. 1) that *both* training methods suggested by the derivation of this bound (random perturbations and n-SAM) do not substantially improve generalization. This generalization bound can be similarly extended to m-SAM by upper bounding the leading term via the maximum taken over mini-batches. However, this bound would incorrectly suggest that 128-SAM should have the worst generalization among all the three weight-perturbation methods while it is the only method that successfully improves generalization.

We note that coming up with tight generalization bounds even for well-established ERM for overparametrized models is an open research question (Nagarajan & Kolter, 2019). One could expect, however, that at least the *relative* tightness of the bounds could reflect the correct ranking between the three methods, but it is not the case. Thus, we conclude that the existing generalization bound cannot explain the generalization improvement of low-m SAM.

A flatter minimum does not always lead to better generalization. One could assume that although the generalization bound that relies on m-sharpness is loose, m-sharpness can still be an important quantity for generalization. This is suggested by its better correlation with the test error com-

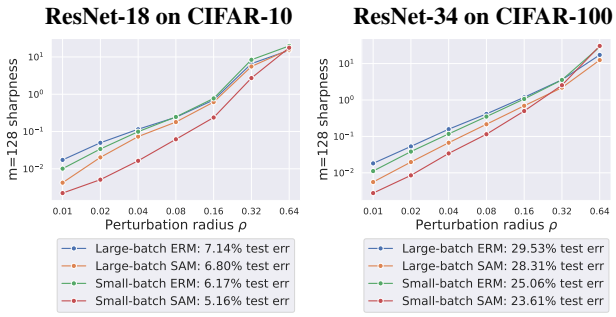


Figure 2: $m = 128$ sharpness computed over different perturbation radii ρ at the minima of ERM and SAM models trained with large (1024) and small batches (128). All models are trained with group normalization and achieve zero training error.

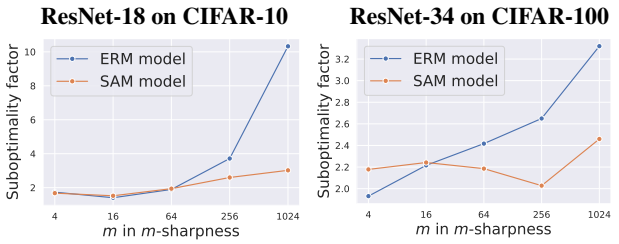


Figure 3: Suboptimality factor of m-sharpness ($\rho = 0.1$) computed using 100 steps of projected gradient ascent compared to only 1 step for ERM and SAM models with group normalization.

pared to the sharpness computed on the whole training set (Foret et al., 2021). In particular, we could expect that convergence of SAM to better-generalizing minima can be explained by a lower m-sharpness of these minima. To check this hypothesis, we select multiple models trained with group normalization[1] that achieve zero training error and measure their m-sharpness for $m = 128$ and different perturbation radii ρ in Fig. 2. We note that the considered networks are not reparametrized in an adversarial way (Dinh et al., 2017) and they all use the same weight decay parameters which makes them more comparable to each other. First of all, we observe that *none* of the radii ρ gives the correct ranking between the methods according to their test error, although m-sharpness ranks correctly SAM and ERM for the same batch size. In particular, we see that the minimum found by SAM with a large batch size (1024) is flatter than the minimum found by ERM with a small batch size (128) although the ERM model leads to a better test error: 6.17% vs. 6.80% on CIFAR-10 and 25.06% vs. 28.31% on CIFAR-100. This shows that it is easy to find counterexamples where flatter minima generalize worse.

We further note that there are simple examples that illustrate that m-sharpness cannot be a universal quantity at distinguishing well-generalizing minima. E.g., consider a linear model $f_x(w) = \langle w, x \rangle$ and a decreasing margin-based loss ℓ, then the 1-sharpness has a closed-form solution:

$$\sum_{i=1} \max_{\|\delta\|_2} \ell(y_i \langle w + \delta, x_i \rangle) \quad \ell(y_i \langle w, x_i \rangle) =$$
$$\sum_{i=1} \ell(y_i \langle w, x_i \rangle \quad \rho \|x_i\|_2) \quad \ell(y_i \langle w, x_i \rangle).$$

The 1-sharpness is influenced only by the term $\rho \|x_i\|_2$ which does not depend on a specific w. In particular, it implies that all global minimizers w of the training loss are *equally sharp* according to the 1-sharpness which, thus, cannot suggest which global minima generalize better.

Since (m-)sharpness does not always distinguish better-from worse-generalizing minima, the common intuition about sharp vs. flat minima (Keskar et al., 2016) can be incomplete. This suggests that it is likely that some other quantity is responsible for generalization which can be correlated with (m-)sharpness in *some* cases, but not always. This motivates us to develop a better understanding of the role of m in m-SAM, particularly on simpler models which are amenable for a theoretical study.

4. Understanding the Generalization Benefits of SAM

In this section, we first check empirically whether the advantage of lower m in m-SAM comes from a more accurate solution of the inner maximization problem or from specific properties of batch normalization. We conclude that it is not the case and hypothesize that the advantage comes rather from a better implicit bias of gradient descent induced by m-SAM. We characterize this implicit bias for diagonal linear networks showing that SAM can *provably* improve generalization, and the improvement is larger for 1-SAM than for n-SAM. Then we complement the theoretical results with experiments on deep networks showing a few intriguing properties of SAM.

4.1. Testing Two Natural Hypotheses for Why Low m in m-SAM Could be Beneficial

As illustrated in Fig. 1, the success of m-SAM fully relies on the effect of low m which is, however, remains unexplained in the current literature. As a starting point, we could consider the following two natural hypotheses for why low m could be beneficial.

Hypothesis 1: **lower m leads to more accurate maximization.** Since m-SAM relies only on a *single* step of projected gradient ascent for the inner maximization prob-

[1] We consider networks with group normalization (Wu & He, 2018) instead of the more common batch normalization (Ioffe & Szegedy, 2015) since we observed a large discrepancy between m-sharpness computed with the training-time vs. test-time batch normalization (see the experiment in Fig. 19 in App. E.4).

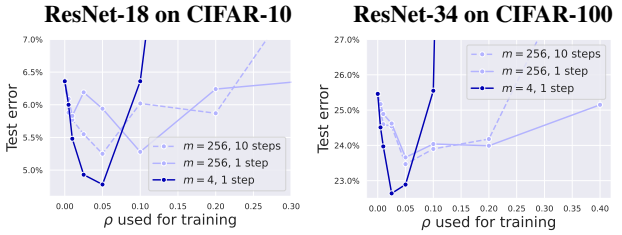


Figure 4: Test error of SAM models with group normalization trained with different numbers of projected gradient ascent steps (10 vs. 1) for m-SAM and different m values (256 vs. 4) using batch size 256.

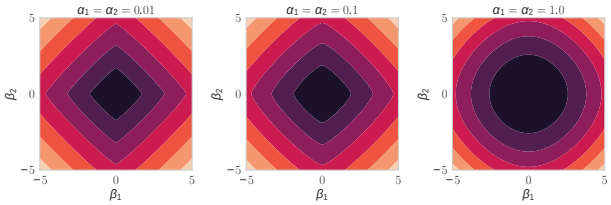


Figure 5: Illustration of the hyperbolic entropy $\phi_\alpha(\beta)$ for $\beta \in \mathbb{R}^2$ that interpolates between $k\beta k_1$ for small α and $k\beta k_2$ for large α.

lem in Eq. (3), it is unclear in advance how accurately this problem is solved. One could assume that using a lower m can make the single-step solution more accurate as intuitively the function which is being optimized might become "simpler" due to fewer terms in the summation. Indeed, there is evidence towards this hypothesis: Fig. 3 shows the suboptimality factor between m-sharpness computed using 100 steps vs. 1 step of projected gradient ascent for $\rho = 0.1$ (the optimal ρ for 256-SAM in terms of generalization) for ERM and SAM models. We can see that the suboptimality factor tends to increase over m and can be as large as 10 for the ERM model on CIFAR-10 for $m = 1024$. This finding suggests that the standard single-step m-SAM can indeed fail to find an accurate maximizer and the value of m can have a significant impact on it. However, despite this fact, using multiple steps in SAM *does not* improve generalization as we show in Fig. 4. E.g., on CIFAR-10 it merely leads to a shift of the optimal ρ from 0.1 to 0.05, without noticeable improvements of the test error. This is also in agreement with the observation from Foret et al. (2021) on why including second-order terms can slightly hurt generalization: solving the inner maximization problem more accurately leads to the fact that the same radius ρ can become effectively too large (as on CIFAR-10) leading to worse performance.

Hypothesis 2: **lower m results in a better regularizing effect of batch normalization.** As pointed out in Hoffer et al. (2017) and Goyal et al. (2017), batch normalization (BN) has a beneficial regularization effect that depends on the mini-batch size. In particular, using the BN statistics from a smaller subbatch is coined as *ghost batch normalization* (Hoffer et al., 2017) and tends to improve generalization. Thus, it could be the case that the generalization improvement of m-SAM is due to this effect as its implementation assumes using a smaller subbatch of size m. To test this hypothesis, in Fig. 4, we show results of networks trained instead with *group normalization* that does not lead to any extra dependency on the effective batch size. We can see that a significant generalization improvement by m-SAM is still achieved for low m ($m = 4$ for batch size 256), and this holds for both datasets. Thus, the generalization improvement of m-SAM is not specific to BN.

We hypothesize instead that low-m SAM leads to a better *implicit* bias of gradient descent for commonly used neural network architectures, meaning that some important complexity measure of the model gets implicitly minimized over training that may not be obviously linked to m-sharpness.

4.2. Provable Benefit of SAM for Diagonal Linear Networks

Here we theoretically study the implicit bias of full-batch 1-SAM and n-SAM for diagonal linear networks on a sparse regression problem. We show that 1-SAM has a better implicit bias than ERM *and* n-SAM which explains its improved generalization in this setting.

Implicit bias of 1-SAM and n-SAM. The implicit bias of gradient methods is well understood for overparametrized linear models where all gradient-based algorithms enjoy the same implicit bias towards minimization of the ℓ_2-norm of the parameters. For diagonal linear neural networks, where a linear predictor $\langle \beta, x \rangle$ can be parametrized via $\beta = w_+^2 - w^2$ [2] with a parameter vector $w = \begin{smallmatrix} w_+ \\ w \end{smallmatrix} \in \mathbb{R}^{2d}$, first-order algorithms have a richer implicit bias. We consider here an overparametrized sparse regression problem, meaning that the ground truth β is a sparse vector, with the squared loss:

$$L(w) := \frac{1}{4n} \sum_{i=1}^{n} (\langle w_+^2 - w^2, x_i \rangle - y_i)^2, \qquad (5)$$

where overparametrization means that $n < d$ and there exist many w such that $L(w) = 0$. We note that in our setting, any global minimizer w of $L(w)$ is also a global minimizer for the m-SAM algorithm for any $m \in \{1, \dots, n\}$ since all per-example gradients are zero and hence the ascent step of SAM will not modify w. Thus, any difference in generalization between m-SAM and ERM has to be attributed rather to the *implicit* bias of each of these algorithms.

We first recall the seminal result of Woodworth et al. (2020) and refer the readers to App. B for further details. Assuming

[2] See Woodworth et al. (2020) for why this parametrization is equivalent to a diagonal network $\beta = u \odot v$. Moreover, the signs of u_i and v_i will not change throughout training, hence the use of the notation w_+ and w_-.

global convergence, the solution selected by the gradient flow initialized as $w_+ = w = \alpha\, 2\, \mathsf{R}^d_{>0}$ and denoted β_1 solves the following constrained optimization problem:

$$\beta_1 = \underset{2\mathsf{R}^d \text{ s.t. } X = y}{\arg\min} \phi(\beta), \tag{6}$$

where the potential ϕ is given as $\phi(\beta) = \sum_{i=1}^d \alpha_i^2 q(\beta_i/\alpha_i^2)$ with $q(z) = 2 - \sqrt{4+z^2} + z\,\mathsf{arcsinh}(z/2)$. As illustrated in Fig. 5, ϕ interpolates between the ℓ_1 and the ℓ_2 norms of β according to the initialization scale α. Large α's lead to low ℓ_2-type solutions, while small α's lead to low ℓ_1-type solutions which are known to induce good generalization properties for sparse problems (Woodworth et al., 2020).

Our main theoretical result is that both 1-SAM and n-SAM dynamics, when considered in their full-batch version (see Sec. A for details), bias the flow towards solutions which minimize the potential ϕ but with effective parameters $\alpha_{\text{1-SAM}}$ and $\alpha_{\text{n-SAM}}$ which are strictly smaller than α for a suitable inner step size ρ. In addition, typically $\|\alpha_{\text{1-SAM}}\|_1 < \|\alpha_{\text{n-SAM}}\|_1$ and, therefore, the solution chosen by 1-SAM has better sparsity-inducing properties than the solution of n-SAM and standard ERM.

Theorem 1 (Informal)**.** *Assuming global convergence, the solutions selected by the full-batch versions of the* 1*-SAM and* n*-SAM algorithms taken with infinitesimally small step sizes and initialized at* $w_+ = w = \alpha\, 2\, \mathsf{R}^d_{>0}$*, solve the optimization problem (6) with effective parameters:*

$$\alpha_{\text{1-SAM}} = \alpha\, e^{-\frac{\rho}{\alpha}+\mathcal{O}(\rho^2)},\quad \alpha_{\text{n-SAM}} = \alpha\, e^{-\frac{\rho}{\alpha}+\mathcal{O}(\rho^2)},$$

where $_{\text{1-SAM}}, {}_{\text{n-SAM}} \in \mathsf{R}^d_+$ *for which typically:*

$$\|{}_{\text{1-SAM}}\|_1 \approx \int_0^{\frac{\rho}{\alpha}} L(w(s))ds \quad \text{and}$$

$$\|{}_{\text{n-SAM}}\|_1 \approx \frac{d}{n}\int_0^{\frac{\rho}{\alpha}} L(w(s))ds.$$

The results are formally stated in Theorem 4 and 5 in App. B. 1-SAM has better implicit bias properties since its effective scale of α is considerably smaller than the one of n-SAM due to the lack of the $\frac{1}{n}$ factor in the exponent. It is worth noting that the vectors $_{\text{1-SAM}}$ and $_{\text{n-SAM}}$ are linked with the integral of the loss function along the flow. Thereby, the speed of convergence of the training loss impacts the magnitude of the biasing effect: the slower the convergence, the better the bias, similarly to what is observed for SGD in Pesme et al. (2021). Extending this result to stochastic implementations of 1-SAM and n-SAM algorithms could be done following Pesme et al. (2021) but is outside of the scope of this paper.

Empirical evidence for the implicit bias. We compare the training and test loss of ERM, 1-SAM, and n-SAM in Fig. 6

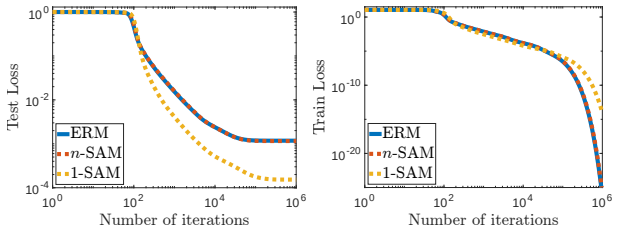


Figure 6: Implicit bias of 1-SAM and n-SAM compared to ERM for a diagonal linear network on a sparse regression problem. We can see that 1-SAM generalizes significantly better than n-SAM and ERM.

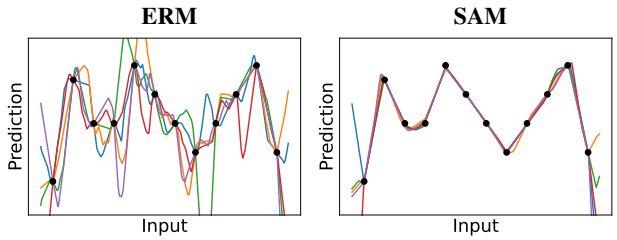


Figure 7: The effect of the implicit bias of ERM vs. SAM for a one hidden layer ReLU network trained with full-batch gradient descent. Each run is replicated over five random initializations.

for the same perturbation radius ρ, and for different ρ in App. B.3 (Fig. 14). As predicted, the methods show different generalization abilities: ERM and n-SAM achieve approximately the same performance whereas 1-SAM clearly benefits from a better implicit bias. This is coherent with the deep learning experiments presented in Fig. 1 on CIFAR-10 and CIFAR-100. We also note that the training loss of all the variants is converging to zero but the convergence of 1-SAM is slower. Additionally, we show a similar experiment with *stochastic* variants of the algorithms in App. B.3 (Fig. 13) where their performance is, as expected, better compared to their deterministic counterparts.

4.3. Empirical Study of the Implicit Bias in Non-Linear Networks

Here we conduct a series of experiments to characterize the implicit bias of SAM on *non-linear* networks.

The sparsity-inducing bias of SAM for a simple ReLU network. We start from the simplest non-linear network: a one hidden layer ReLU network applied to a simple 1D regression problem from Blanc et al. (2020). We use it to illustrate the implicit bias of SAM in terms of the geometry of the learned function. For this, we train ReLU networks with 100 hidden units using full-batch gradient descent on the quadratic loss with ERM and SAM[3] over five different random initializations. We plot the resulting functions in

[3] Since $n = 12$ for this task, we observed no substantial difference between 1-SAM and n-SAM.

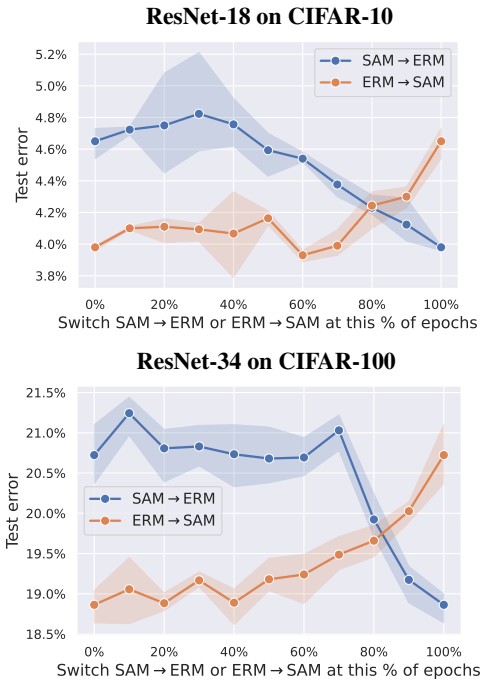


Figure 8: Test error of SAM ! ERM and ERM ! SAM when the methods are switched at different % of epochs. For example, for SAM ! ERM, 0% corresponds to ERM and 100% corresponds to SAM. We observe that a method which is run at the beginning of training has little influence on the final performance.

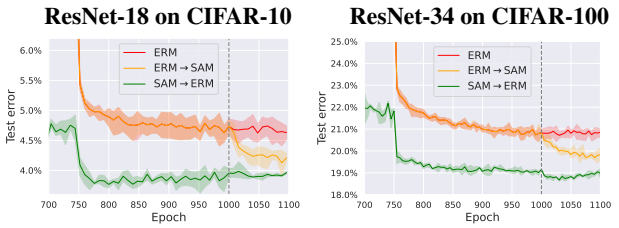


Figure 9: Test error over epochs for ERM compared to ERM ! SAM and SAM ! ERM training where the methods are switched only at the end of training. In particular, we can see that SAM can gradually escape the worse-generalizing minimum found by ERM.

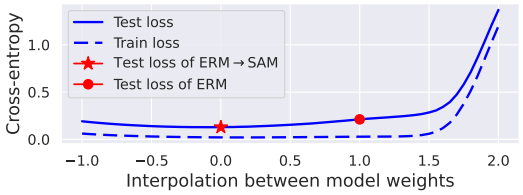


Figure 10: Loss interpolations between $w_{ERM\ !\ SAM}$ and w_{ERM} for a ResNet-18 trained on CIFAR-10.

Fig. 7. We observe that SAM leads to simpler interpolations of the data points than ERM, and it is much more stable over random initializations. In particular, SAM seems to be biased toward a sparse combination of ReLUs which is reminiscent of Chizat & Bach (2020) who show that the limits of the gradient flow can be described as a max-margin classifier that favors hidden low-dimensional structures by implicitly regularizing the F_1 variation norm. Moreover, this also relates to our Theorem 1 where sparsity rather shows up in terms of the lower ℓ_1-norm of the resulting linear predictor. This further illustrates that there can exist multiple ways in which one can describe the beneficial effect of SAM. For deep non-linear networks, however, the effect of SAM is hard to visualize, but we can still characterize some of its important properties.

The effect of SAM for deep networks at different stages of training. To develop a better understanding of the implicit bias of SAM for deep networks, we can analyze at which stages of training using SAM is necessary to get generalization benefits. One could assume, for example, that its effect is important *only* early in training so that the first updates of SAM steer the optimization trajectory towards a better-generalizing minimum. In that case, switching from SAM to ERM would not degrade the performance. To better understand this, we train models first with SAM and then switch to ERM for the remaining epochs (SAM ! ERM) and also do a complementary experiment by switching from ERM to SAM (ERM ! SAM) and show results in Fig. 8. Interestingly, we observe that a method that is used at the beginning of training has little influence on the final performance. E.g., when SAM is switched to ERM within the first 70% epochs on CIFAR-100, the resulting model generalizes as well as ERM. Furthermore, we note a high degree of continuity of the test error with respect to the number of epochs at which we switch the methods. This does not support the idea that the models converge to some entirely distinct minima and instead suggests convergence to different minima in a connected valley where some directions generalize progressively better. Another intriguing observation is that enabling SAM only towards the end of training is sufficient to get a significant improvement in terms of generalization. We discuss this phenomenon next in more detail.

The importance of the implicit bias of SAM at the end of training. We take a closer look on the performance of ERM ! SAM and SAM ! ERM when we switch between the methods only for the last 10% of epochs in Fig. 9 where we plot the test error over epochs. First, we see that for SAM ! ERM, once SAM converges to a well-generalizing minimum thanks to its implicit bias, then it is not important whether we continue optimization with SAM or with ERM, and we do not observe significant overfitting when switching to ERM. At the same time, for ERM ! SAM we observe a different behavior: the test error clearly improves when switching from ERM to SAM. This suggests that SAM (using a higher ρ than the standard value, see App. D) can gradually escape the worse-generalizing minimum which ERM converged to. This phenomenon

is interesting since it suggests a *practically relevant* fine-tuning scheme that can save computations as we can start from any pre-trained model and substantially improve its generalization. Moreover, interestingly, the final point of the ERM ! SAM model is situated *in the same basin* as the original ERM model as we show in Fig. 10 which resembles the asymmetric loss interpolations observed previously for stochastic weight averaging (He et al., 2019).

We make very similar observations regarding fine-tuning with SAM and linear connectivity also on a diagonal linear network as shown in App. B.3 (Fig. 15). We believe the observations from Fig. 9 can be explained by our Theorem 1 which shows that for diagonal linear networks, the key quantity determining the magnitude of the implicit bias for SAM is the integral of the loss over the optimization trajectory $w(s)$. In the case of ERM ! SAM, the integral is taken only over the last epochs but this can still be sufficient to improve the biasing effect. At the same time, for SAM ! ERM, the integral is already large enough due to the first 1000 epochs with SAM and switching back to ERM preserves the implicit bias. We discuss it in more detail in App. B.3.

5. Understanding the Optimization Aspects of SAM

The results on the implicit bias of SAM presented above require that the algorithm converges to zero training error. In the current literature, however, a convergence analysis (even to a stationary point) is missing for SAM. In particular, we do not know what are the conditions on the training ERM loss, inner step size γ_t, and perturbation radius ρ_t so that SAM is guaranteed to converge. We also do not know whether SAM converges to a stationary point of the ERM objective. To fill in this gap, we first theoretically study convergence of SAM and then relate the theoretical findings with empirical observations on deep networks.

5.1. Theoretical Analysis of Convergence of SAM

Here we show that SAM leads to convergence guarantees in terms of the standard training loss. In the following, we analyze the convergence of the m-SAM algorithm whose update rule is defined in Eq. (4). We make the following assumptions on the training loss $L(w) = \frac{1}{n}\sum_{i=1}^{n} \ell_i(w)$:

(A1) (Bounded variance). *There exists* $\sigma \geq 0$ *s.t.* $\mathbb{E}[\|\nabla \ell_i(w) - \nabla L(w)\|^2] \leq \sigma^2$ *for all* $i \sim U([\![1, n]\!])$ *and* $w \in \mathbb{R}^d$.

(A2) (Individual β-smoothness). *There exists* $\beta \geq 0$ *s.t.* $\|\nabla \ell_i(w) - \nabla \ell_i(v)\| \leq \beta\|w - v\|$ *for all* $w, v \in \mathbb{R}^d$ *and* $i \in [\![1, n]\!]$.

(A3) (Polyak-Lojasiewicz). *There exists* $\mu > 0$ *s.t.* $\frac{1}{2}\|\nabla L(w)\|^2 \geq \mu(L(w) - L^*)$ *for all* $w, v \in \mathbb{R}^d$.

Both assumptions **(A1)** and **(A2)** are standard in the optimization literature and should hold for neural networks with smooth activations and losses (such as cross-entropy). The assumption **(A2)** requires the inputs to be bounded but this is typically satisfied (e.g., images are all in $[0, 1]^d$). The assumption **(A3)** corresponds to easier problems (e.g., strongly convex ones) for which global convergence can be proven. We have the following convergence result:

Theorem 2. *Assume* ***(A1)*** *and* ***(A2)*** *for the iterates (4). Then for any number of iterations* $T \geq 0$*, batch size* b*, and step sizes* $\gamma_t = \frac{1}{T^{1/4}}$ *and* $\rho_t = \frac{1}{T^{1/4}}$*, we have:*

$$\frac{1}{T}\mathbb{E}\left[\sum_{t=0}^{T-1} \|\nabla L(w_t)\|^2\right] \leq \frac{4\beta}{T^{3/4}}(L(w_0) - L^*) + \frac{8\sigma^2}{b\sqrt{T}},$$

In addition, under ***(A3)****, with step sizes* $\gamma_t = \min\left\{\frac{8t+4}{3(t+1)^2}, \frac{1}{2\beta}\right\}$ *and* $\rho_t = \gamma_t/\beta$*:*

$$\mathbb{E}[L(w_T)] - L^* \leq \frac{3\beta^2(L(w_0) - L^*)}{\mu^2 T^2} + \frac{22\beta\sigma^2}{\mu^2 bT}.$$

We provide the proof in App. C.2 and make several remarks:

- We recover the rates of SGD with the usual condition on the step size γ_t (Ghadimi & Lan, 2013; Karimi et al., 2016).
- The ascent step size ρ_t, however, has to be $O(\sqrt{\gamma_t})$ to ensure convergence, i.e., it tolerates a slower decrease than γ_t. This finding is aligned with the observation that the ascent step size should not be decreased as drastically as the descent step size when training neural networks (see Fig. 21 in App. E.6).
- On the technical side, the proof relies on the bound $\langle \nabla L(w_t + \eta \nabla L(w_t)), \nabla L(w_t)\rangle \geq (1 - \eta\beta)\|\nabla L(w_t)\|^2$ which shows that SAM-step is well aligned with the gradient step (see Lemma 16 in App. C.2).

5.2. Convergence of SAM for Deep Networks

Here we relate the convergence analysis to empirical observations for deep learning tasks.

Both ERM and SAM converge for deep networks. We compare the behavior of ERM and SAM by training a ResNet-18 on CIFAR-10 and CIFAR-100 for 1000 epochs (see App. D for experimental details) and plot the results over epochs in Fig. 11. We observe that not only the ERM model but also the model trained with SAM fits all the training points and converges to a *nearly zero training loss*: 0.0013 ± 0.00002 for ERM vs 0.0034 ± 0.0004 for SAM on CIFAR-10. However, the SAM model has significantly better generalization performance due to its implicit bias: $4.75\% \pm 0.14\%$ vs. $3.94\% \pm 0.09\%$ test error. Moreover,

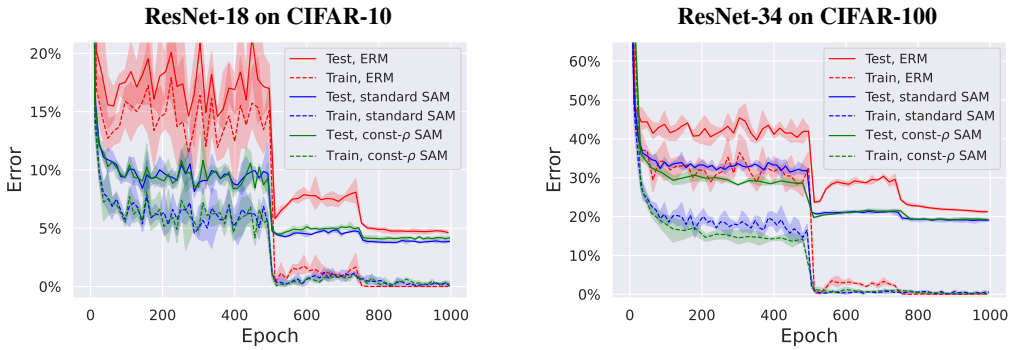


Figure 11: Training and test error of ERM, standard SAM, and SAM with a constant step size ρ (i.e., without gradient normalization) over epochs. We can see that both ERM and SAM converge to zero training error and the gradient normalization is not crucial for SAM.

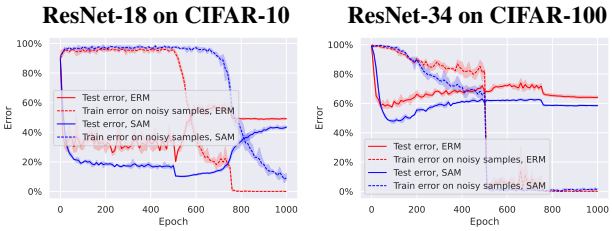


Figure 12: Error rates of ERM and SAM over epochs on CIFAR-10 and CIFAR-100 *with 60% label noise*. We see that the test error increases when the models fit the noisy samples.

we observe no noticeable overfitting throughout training: the best and last model differ by at most 0.1% test error for both methods. Finally, we note that the behavior of ERM vs. SAM on CIFAR-100 is qualitatively similar.

Performance of SAM with constant step sizes ρ_t. Our convergence proof in Sec. 5.1 for non-convex objectives relies on constant step sizes ρ_t. However, the standard SAM algorithm as introduced in Foret et al. (2021) uses step sizes ρ_t inversely proportional to the gradient norm. Thus, one can wonder if such step sizes are important for achieving better convergence or generalization. Fig. 11 shows that on CIFAR-10 and CIFAR-100, both methods converge to zero training error at a similar speed. Moreover, they achieve similar improvements in terms of generalization: 3.94% 0.09% test error for standard SAM vs. 4.15% 0.16% for SAM with constant ρ_t on CIFAR-10. For CIFAR-100, the test error matches almost exactly: 19.22% 0.38% vs. 19.30% 0.38%. We also note that the optimal ρ differs for both formulations: $\rho_t = 0.2/\|\mathbf{r}\|_2$ with normalization vs. $\rho_t = 0.3$ without normalization, so simply removing the gradient normalization without doing a new grid search over ρ_t can lead to suboptimal results.

Is it always beneficial for SAM to converge to zero loss? Here we consider the setting of uniform label noise, i.e., when a fraction of the training labels is changed to random labels and kept fixed throughout the training. This setting differs from the standard noiseless case (typical for many vision datasets such as CIFAR-10) as converging to nearly zero training loss is harmful for ERM and leads to substantial overfitting. Thus, one could assume that the beneficial effect of SAM in this setting can come from preventing convergence and avoiding fitting the label noise. We plot test error and training error on noisy samples for a ResNet-18 trained on CIFAR-10 and CIFAR-100 with 60% label noise in Fig. 12. We see that SAM noticeably improves generalization over ERM, although later in training SAM also starts to fit the noisy points which is in agreement with the convergence analysis. In App. E.7, we confirm the same findings for SAM with constant ρ_t. Thus, SAM also requires early stopping either explicitly via a validation set or implicitly via restricting the number of training epochs as done, e.g., in Foret et al. (2021). Interestingly, this experiment also suggests that the beneficial effect of SAM is observed not only close to a minimum but also along the whole optimization trajectory. Overall, we conclude that SAM can easily overfit and its convergence in terms of the training loss can be a negative feature for datasets with noisy labels.

6. Conclusions

We showed why the existing justifications for the success of m-SAM based on generalization bounds and the idea of convergence to flat minima are incomplete. We hypothesized that there exists some other quantity which is responsible for the improved generalization of m-SAM which is implicitly minimized. We analyzed the implicit bias of 1-SAM and n-SAM for diagonal linear networks showing that the implicit quantity which is minimized is related to the ℓ_1-norm of the resulting linear predictor, and it is stronger for 1-SAM than for n-SAM. We further studied the properties of the implicit bias on non-linear networks empirically where we showed that fine-tuning an ERM model with SAM can lead to significant generalization improvements. Finally, we provided convergence results of SAM for non-convex objectives when used with stochastic gradient which we confirmed empirically for deep networks and discussed its relation to the generalization behavior of SAM.

References

Guy Blanc, Neha Gupta, Gregory Valiant, and Paul Valiant. Implicit regularization for deep neural networks driven by an Ornstein-Uhlenbeck like process. In COLT, 2020.

Lénaïc Chizat and Francis Bach. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. In COLT, 2020.

Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In ICML, pp. 1019–1028. PMLR, 2017.

Gintare Karolina Dziugaite and Daniel Roy. Entropy-sgd optimizes the prior of a pac-bayes bound: Generalization properties of entropy-sgd and data-dependent priors. In ICML, 2018.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In ICLR, 2021.

Saeed Ghadimi and Guanghui Lan. Stochastic first- and zeroth-order methods for nonconvex stochastic programming. SIAM Journal on Optimization, 23(4):2341–2368, 2013.

Robert Mansel Gower, Nicolas Loizou, Xun Qian, Alibek Sailanbayev, Egor Shulgin, and Peter Richtárik. SGD: General analysis and improved rates. In ICML, 2019.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. arXiv preprint arXiv:1706.02677, 2017.

Alex Graves, Abdel-rahman Mohamed, and Geoffrey Hinton. Speech recognition with deep recurrent neural networks. In 2013 IEEE ICASSP, 2013.

Kosuke Haruki, Taiji Suzuki, Yohei Hamakawa, Takeshi Toda, Ryuji Sakai, Masahiro Ozawa, and Mitsuhiro Kimura. Gradient noise convolution (GNC): Smoothing loss function for distributed large-batch sgd. arXiv preprint arXiv:1906.10822, 2019.

Haowei He, Gao Huang, and Yang Yuan. Asymmetric valleys: Beyond sharp and flat local minima. In NeurIPS, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In ECCV, 2016.

Sepp Hochreiter and Jürgen Schmidhuber. Simplifying neural nets by discovering flat minima. In NeurIPS, 1995.

Elad Hoffer, Itay Hubara, and Daniel Soudry. Train longer, generalize better: closing the generalization gap in large batch training of neural networks. In NeurIPS, 2017.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML, 2015.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In ICLR, 2019.

Kam-Chuen Jim, C Lee Giles, and Bill G Horne. An analysis of noise in recurrent neural networks: convergence and generalization. In IEEE Transactions on Neural Networks, 1996.

Hamed Karimi, Julie Nutini, and Mark Schmidt. Linear convergence of gradient and proximal-gradient methods under the Polyak-Lojasiewicz condition. In Joint European Conference on Machine Learning and Knowledge Discovery in Databases, 2016.

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 ICLR, 2016.

Galina Korpelevich. Extragradient method for finding saddle points and other problems. In Matekon, 1977.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical Report, 2009.

Jungmin Kwon, Jeongseop Kim, Hyunseo Park, and In Kwon Choi. ASAM: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks. In ICML, 2021.

Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In Neural networks: Tricks of the trade, pp. 9–48. Springer, 2012.

Tao Lin, Lingjing Kong, Sebastian Stich, and Martin Jaggi. Extrapolation for large-batch training in deep learning. In ICML, 2020.

Shengchao Liu, Dimitris Papailiopoulos, and Dimitris Achlioptas. Bad global minima exist and SGD can reach them. In NeurIPS, 2019.

Rotem Mulayoff and Tomer Michaeli. Unique properties of flat minima in deep networks. In ICML, 2020.

Alan F Murray and Peter J Edwards. Synaptic weight noise during MLP learning enhances fault-tolerance, generalization and learning trajectory. In NeurIPS, 1993.

Vaishnavh Nagarajan and J Zico Kolter. Uniform convergence may be unable to explain generalization in deep learning. In NeurIPS, 2019.

Preetum Nakkiran, Gal Kaplun, Yamini Bansal, Tristan Yang, Boaz Barak, and Ilya Sutskever. Deep double descent: Where bigger models and more data hurt. In ICLR, 2020.

Yurii Nesterov. Introductory Lectures on Convex Optimization. Kluwer Academic, 2004.

Gergely Neu. Information-theoretic generalization bounds for stochastic gradient descent. In COLT, 2021.

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. In ICLR workshops, 2015.

Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In NeurIPS, 2017.

Scott Pesme, Loucas Pillaud-Vivien, and Nicolas Flammarion. Implicit bias of sgd for diagonal linear networks: a provable benefit of stochasticity. In NeurIPS, 2021.

Leslie Rice, Eric Wong, and J Zico Kolter. Overfitting in adversarially robust deep learning. In ICML, 2020.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. JMLR, 2014.

Ziqiao Wang and Yongyi Mao. On the generalization of models trained with SGD: Information-theoretic bounds and implications. In ICLR, 2022.

Michael L. Waskom. Seaborn: statistical data visualization. Journal of Open Source Software, 6(60):3021, 2021. doi: 10.21105/joss.03021. URL https://doi.org/10.21105/joss.03021.

Wei Wen, Yandan Wang, Feng Yan, Cong Xu, Chunpeng Wu, Yiran Chen, and Hai Li. Smoothout: Smoothing out sharp minima to improve generalization in deep learning. arXiv preprint arXiv:1805.07898, 2018.

Blake Woodworth, Suriya Gunasekar, Jason D. Lee, Edward Moroshko, Pedro Savarese, Itay Golan, Daniel Soudry, and Nathan Srebro. Kernel and rich regimes in overparametrized models. In COLT, 2020.

Dongxian Wu, Shu-tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. In NeurIPS, 2020.

Yuxin Wu and Kaiming He. Group normalization. In ECCV, 2018.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. In ICLR, 2017.

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

Appendix

Organization of the appendix

The appendix is organized as follows:

- Sec. A: implementations in the full-batch setting of 1-SAM and n-SAM.
- Sec. B: proofs related to the implicit bias of 1-SAM and n-SAM.
- Sec. C: proofs related to the convergence of different variants of SAM.
- Sec. D: experimental details for the experiments with deep networks and linear models.
- Sec. E: additional experiments complementary to the experiments in the main part.

A. Implementations of the SAM Algorithm in the Full-Batch Setting

We define here the implementations of the SAM algorithm in the full-batch setting for the two extreme values of m we consider, i.e., $m = 1$ and $m = n$. They correspond to the following objectives:

$$\text{n-SAM: } \min_{w \in \mathbb{R}^d} \max_{\|\epsilon\|_2 \leq \rho} \frac{1}{n} \sum_{i=1}^{n} \ell_i(w + \epsilon); \qquad \text{1-SAM: } \min_{w \in \mathbb{R}^d} \frac{1}{n} \sum_{i=1}^{n} \max_{\|\epsilon\|_2 \leq \rho} \ell_i(w + \epsilon): \tag{7}$$

The update rule of the SAM algorithm for these objectives amounts to a variant of gradient descent with step size η_t where the gradients are taken at intermediate points $w^i_{t+1/2}$, i.e., $w_{t+1} = w_t - \frac{\eta_t}{n} \sum_{i=1}^{n} \nabla \ell_i(w^i_{t+1/2})$. The updates, however, differ in how the points $w^i_{t+1/2}$ are computed since they approximately maximize different functions with inner step sizes ρ_t:

$$\text{n-SAM: } w^i_{t+1/2} = w_t + \frac{\rho_t}{n} \sum_{j=1}^{n} \nabla \ell_j(w_t); \qquad \text{1-SAM: } w^i_{t+1/2} = w_t + \rho_t \nabla \ell_i(w_t): \tag{8}$$

To make the SAM algorithm practical, Foret et al. (2021) propose to combine SAM with stochastic gradients which corresponds to the m-SAM algorithm defined in Eq. (4) in the main part.

B. Theoretical Analysis of the Implicit Bias for Diagonal Linear Networks

To understand why m-SAM is generalizing better than ERM, we consider the simpler problem of noiseless regression with 2-layer diagonal linear network for which we can precisely characterize the implicit bias of different optimization algorithms.

Optimization algorithms. We consider minimizing the training loss $L(w)$ using the following optimization algorithms:

- Gradient descent with an infinitesimally small step size, i.e., the gradient flow limit:

$$\dot{w}_t = -\nabla L(w_t): \tag{9}$$

- The n-SAM algorithm from Eq. (8) taken with an infinitesimally small outer step size and inner step size ρ:

$$\dot{w}_t = -\nabla L(w_t + \rho \nabla L(w_t)): \tag{10}$$

- The 1-SAM algorithm from Eq. (8) taken with an infinitesimally small outer step size and inner step size ρ:

$$\dot{w}_t = -\frac{1}{n} \sum_{i=1}^{n} \nabla \ell_i(w_t + \rho \nabla \ell_i(w_t)): \tag{11}$$

Previous work: implicit bias of the gradient ow. We rst de ne the function for $2 R^d$ which will be very useful to precisely characterize the implicit bias of the optimization algorithms we consider:

$$( ) = \sum_{i=1}^{d} {}_i^2 q(_i = {}_i^2) \text{ where } q(z) = \int_0^z \operatorname{arcsinh}(u=2)du = 2 \sqrt{4 + z^2} + z \operatorname{arcsinh}(z=2): \quad (12)$$

Following Woodworth et al. (2020), one can show the following result for the gradient ow dynamics in Eq. (9).

Theorem 3 (Theorem 1 of Woodworth et al. (2020)). If the solution $_1$ of the gradient ow (9) started from $w_+ = w = 2 R^d_{>0}$ for the squared parameter problem in Eq. (5) satisfies $_1 = y$, then

$$_1 = \arg\min_{2 R^d} ( ) \text{ s.t. } X = y; \quad (13)$$

where is de ned in Eq. (12).

It is worth noting that the implicit regularizer interpolates between the and $`_2$ norms (see Woodworth et al., 2020, Theorem 2). Therefore the scale of the initialization determines the implicit bias of the gradient ow. The algorithm, started from , converges to the minimum $`_1$-norm interpolator for small and to the minimum $`_2$-norm interpolator for large . The proof follows from (a) the KKT condition for the optimization problem (13): $(w) = X^>$ for a Lagrange multiplier and (b) the closed form solution obtained by integrating the gradient ow $w = b(X^>)$ for some function b and some vector. Identifying $r (w) = b^{-1}(w)$ leads to the solution. Considering the same proof technique, we now derive the implicit bias for the -SAM and 1-SAM algorithms.

B.1. Implicit Bias of the -SAM Algorithm.

We start from characterizing the implicit bias of the SAM dynamics (10) in the following theorem using the function de ned in Eq. (12). We will also make use of this notation: a parameter vector $\begin{smallmatrix} w_+ \\ w \end{smallmatrix} 2 R^{2d}$, a concatenation of matrices $X = [X \quad X] 2 R^{n \quad 2d}$ and a residual vector $r(t) = X w(t)^2 \quad y$.

Theorem 4. If the solution $_1$ of the -SAM gradient ow (10) started from $w_+ = w = 2 R^d_{>0}$ for the squared parameter problem in Eq. (5) satisfies $_1 = y$, then

$$_1 = \arg\min_{\quad} {}_{\text{-SAM}}( ) \text{ s.t. } X = y;$$

where $_{\text{-SAM}} = \exp \frac{2}{n^2} \int_0^{R_1} (X^> r_s)^2 ds + O(^2)$.

We note that for a small enough , the implicit bias parameter $_{\text{-SAM}}$ is smaller than . The scale of the vector $\frac{1}{n^2} \int_0^{R_1} (X^> r_s)^2 ds$ which in uences the implicit bias effect is related to the loss integral $\frac{d}{n} \int_0^{R_1} L(w(s)) ds$ since $kr_s k^2 = nL(w(s))$ (see intuition in Eq. (19)). Thereby the speed of convergence of the loss controls the magnitude of the biasing effect. However in the case of SAM, as explained in Sec. B.3, this effect is typically negligible because of the extra prefactor $\frac{d}{n}$ and this implementation behaves similarly as ERM as shown in the experiments in Sec. 4.2.

Proof. We follow the proof technique of Woodworth et al. (2020). We denote the intermediate step of SAM as $w_{sam}(t) = w(t) + r L(w(t))$ and the residual of $w_{sam}(t)$ as $r_{sam}(t) = X w_{sam}(t)^2 \quad y$. We start from deriving the equation satis ed by the ow

$$\begin{aligned} \underline{w}(t) &= r \quad L(w_{sam}(t)) \\ &= \frac{1}{n} X^> r_{sam}(t) \quad w_{sam}(t) \\ &= \frac{1}{n} X^> r_{sam}(t) \quad w(t) + \frac{}{n} X^> r(t) \quad w(t) : \end{aligned}$$

Now we can directly integrate this ODE to obtain an expression of $w(t)$:

$$w(t) = w(0) \quad \exp \frac{1}{n} X^> \int_0^t r_{sam}(s) ds \quad \exp \frac{}{n^2} \int_0^t X^> r_{sam}(s) \quad X^> r(s) \quad ds :$$

Using that the flow is initialized at $w(0) = \alpha$ and the definition of $\Gamma(t)$ yields to

$$\begin{aligned}\Gamma(t) &= \|w_+(t)\|^2 - \|w_-(t)\|^2 \\ &= \alpha^2 \exp\left(\frac{2}{n} X^\top \int_0^t r_{\text{sam}}(s)ds\right) \exp\left(\frac{2}{n^2}\int_0^t X^\top r_{\text{sam}}(s) - X^\top r(s)\, ds\right) \\ &\quad - \alpha^2 \exp\left(-\frac{2}{n} X^\top \int_0^t r_{\text{sam}}(s)ds\right) \exp\left(\frac{2}{n^2}\int_0^t X^\top r_{\text{sam}}(s) - X^\top r(s)\, ds\right) \\ &= 2\alpha^2 \exp\left(\frac{2}{n^2}\int_0^t X^\top r_{\text{sam}}(s) - X^\top r(s)\, ds\right) \sinh\left(\frac{2}{n} X^\top \int_0^t r_{\text{sam}}(s)ds\right).\end{aligned}$$

Recall that we are assuming that β_1 is a global minimum of the loss, i.e., $X\beta_1 = y$. Thus, β_1 has to simultaneously satisfy

$$X\beta_1 = y \quad \text{and} \quad \beta_1 = b_{\text{n-SAM}}(X^\top \gamma);$$

where $b(z) = 2\alpha^2 \sinh(z)$ and $\gamma = \frac{2}{n}\int_0^1 r_{\text{sam}}(s)ds$, and

$$\Gamma_{\text{n-SAM}} = \exp\left(\frac{2}{n^2}\int_0^1 (X^\top r_{\text{sam}}(s)) - (X^\top r(s))\, ds\right). \tag{14}$$

Next we combine the flow expression $b^{-1}_{\text{n-SAM}}(\beta_1) = X^\top \gamma$ with a KKT condition $r(w) = X^\top \gamma$ and get that

$$r(\beta_1) = b^{-1}(\beta_1) = \operatorname{arcsinh}\left(\frac{1}{2\alpha^2}\right).$$

Integration of this equation leads to $\Gamma(\beta) = \sum_{i=1}^d q(\beta_i^2/\alpha_i^2)$ where $q(z) = \int_0^z \operatorname{arcsinh}(u/2)du = 2\sqrt{4+z^2} + z\operatorname{arcsinh}(z/2)$, i.e., exactly the potential function defined in Eq. (12). Thus, we conclude that β_1 satisfies the KKT conditions $X\beta_1 = y$ and $r(\beta_1) = X^\top \gamma$ for the minimum norm interpolator problem:

$$\min_{\beta \in \mathbb{R}^d} \Gamma(\beta) \quad \text{s.t.} \quad X\beta = y;$$

which proves the first part of the result.

Now to get the expression for $\Gamma_{\text{n-SAM}}$, we apply the definition of $r_{\text{sam}}(s)$ and obtain

$$\begin{aligned} r_{\text{sam}}(t) &= X^\top w_{\text{sam}}(t)^2 - y \\ &= X^\top \Big(w(t) + \frac{1}{n} X^\top r(t)\Big)^2 - y \\ &= r(t) + \frac{2}{n} X^\top X^\top r(t) + \frac{1}{n^2} X^\top X^\top r(t)^2 - w(t)^2 \\ &= r(t) + \frac{2}{n} X^\top X^\top r(t) - (w_+(t) + w_-(t)) + \frac{2}{n^2} X^\top X^\top r(t)^2 - (w_+(t)^2 + w_-(t)^2). \end{aligned}$$

Thus we conclude that $X^\top r_{\text{sam}}(t) = X^\top r(t) + O(\alpha^2)$ which we plug in Eq. (14) to obtain the second part of the theorem:

$$\Gamma_{\text{n-SAM}} = \exp\left(\frac{2}{n^2}\int_0^1 (X^\top r_s)^2 ds + O(\alpha^2)\right).$$

□

B.2. Implicit Bias of the 1-SAM Algorithm

We characterize similarly the implicit bias of the 1-SAM dynamics (11) in the following theorem using the function defined in Eq. (12).

Theorem 5. If the solution $_1$ of the 1-SAM gradient ow (11) started from $w_+ = w = 2 R^d_{>0}$ for the squared parameter problem in Eq. (5) satis es $X _1 = y$, then

$$_1 = \arg\min _{1\text{-SAM}}( ) \text{ s.t. } X = y;$$

where $_{1\text{-SAM}} = \exp \frac{8}{n} \int_0^1 \sum_{i=1}^n x_i^2(x_i^> (s) y_i)^2 ds + O(^2)$.

In addition, assume that there exist B 0 such that almost surely (1) the inputs are bounded $k k_2 R$ and (2) the trajectory of the ow is bounded $k (t)k_2 B$ for all t 0. Then for all $\frac{1}{4R^2}\sqrt{\frac{1}{B(B+k k_2)}}$, we have that $_{1\text{-SAM},i} _i$ for $i 2 f 1; : : : ; dg$.

Proof. The proof follows the same lines as the proof of Theorem 4. We denote a concatenation of positive and negative copies of the i-th training example as $x_i = [\ x_i \ x_i] 2 R^{2d}$, the intermediate step of 1-SAM based on the i-th training example as $w^{(i)}_{sam}(t) 2 R^d$, the residuals of $w(t)$ and $w^{(i)}_{sam}(t)$ on the i-th training example as $r_i(t) = x_i^> w(t)^2 y_i$ and $r_{sam;i}(t) = x_i^> w^{(i)}_{sam}(t)^2 y_i$. Then we have that the dynamics of the ow (11) satis es

$$\begin{aligned} \dot{w}(t) &= \frac{1}{n}\sum_{i=1}^n r \dot{}_i(w^{(i)}_{sam}(t)) \\ &= \frac{1}{n}\sum_{i=1}^n r_{sam;i}(t) x_i \ w^{(i)}_{sam}(t) \\ &= \frac{1}{n}\sum_{i=1}^n r_{sam;i}(t) x_i \ w(t) (1 + 4 r_i(t) x_i) : \end{aligned}$$

Integration of this ODE leads to

$$w(t) = w(0) \exp \frac{1}{n} X^> \int_0^t r_{sam}(s)ds \exp \frac{4}{n}\sum_{i=1}^n x_i^2 \int_0^t r_{sam;i}(s) r_i(s) ds \ :$$

The rest of the proof is similar to the one of Theorem 4 and we directly obtain that

$$_{1\text{-SAM}} = \exp \frac{8}{n}\sum_{i=1}^n x_i^2 \int_0^t r_{sam;i}(s) r_i(s) ds \ : \tag{15}$$

Using the de nition of $r_{sam;i}(t)$ we have

$$\begin{aligned} r_{sam;i}(t) &= x_i^> w_{sam}(t)^2 y_i \\ &= x_i^> w(t)^2 (1 + 4 r_i(t) x_i)^2 y_i \\ &= x_i^> w(t)^2 1 + 8 r_i(t) x_i + 16 ^2 r_i(t)^2 x_i^2 y_i \\ &= r_i(t) + 8 r_i(t) w_+(t)^2 + w (t)^2 > x_i^2 + 16 ^2 r_i(t)^2 w_+(t)^2 w (t)^2 > x_i^3 \\ &= r_i(t) + 8 r_i(t) w_+(t)^2 + w (t)^2 > x_i^2 + 16 ^2 r_i(t)^2 (t)^> x_i^3 \end{aligned}$$

And therefore

$$x_i^2 r_{sam;i}(t) r_i(t) = r_i(t)^2 x_i^2 1 + 8 w_+(t)^2 + w (t)^2 > x_i^2 + 16 ^2 r_i(t) (t)^> x_i^3 \tag{16}$$

This leads to the result stated in the theorem

$$_{1\text{-SAM}} = \exp \frac{8}{n} \int_0^1 \sum_{i=1}^n x_i^2(x_i^> (s) y_i)^2 ds + O(^2) \ : \tag{17}$$

Additionally, from Eq. (16) we can conclude that having ρ such that $1 + 16\,\rho^2 r_i(t)\,(t)^\top x_i^3 \geq 0$ is sufficient to guarantee that $\rho_{1\text{-SAM},i} \geq \rho_i$ for every i. We can use Cauchy-Schwarz inequality twice to upper bound $|r_i(t)\,(t)^\top x_i^3|$:

$$|r_i(t)\,(t)^\top x_i^3| = |x_i^\top (\quad)\,(t)^\top x_i^3| \leq \|x_i\|_2 \|(t)\|_2 \|x_i^3\|_2 \leq \|x_i\|_2^4 (\|(t)\|_2 + \|\quad\|_2)\|(t)\|_2 \leq R^4 (B + \|\quad\|_2) B$$

Thus, we have that $\rho^2 r_i(t)\,(t)^\top x_i^3 \leq \rho^2 R^4 (B + \|\quad\|_2) B \leq \frac{1}{16}$ which leads to the upper bound stated in the theorem $\rho \leq \frac{1}{4R^2\sqrt{B\,(B + \|\quad\|_2)}}$. □

B.3. Comparison between 1-SAM and n-SAM

Theoretical comparison. We wish to compare the two leading terms of the exponents in $\rho_{n\text{-SAM}}$ and $\rho_{1\text{-SAM}}$:

$$I_{n\text{-SAM}}(t) = \frac{1}{n^2} \|X^\top r(t)\|^2 = \frac{1}{n^2} \left\| \sum_{i=1}^n x_i r_i(t) \right\|^2 \quad \text{and} \quad I_{1\text{-SAM}}(t) = \frac{1}{n} \sum_{i=1}^n x_i^2 r_i(t)^2;$$

and relate them to the loss values $w(t)$.

We first note that using Cauchy-Schwarz inequality can directly imply that $I_{1\text{-SAM},i}(t) \leq I_{n\text{-SAM},i}(t)$. However, we aim at obtaining a more quantitative result, even though the following derivations will be informal. Comparing norms of $I_{n\text{-SAM}}(t)$ and $I_{1\text{-SAM}}(t)$ amounts to compare the following two quantities:

$$\|I_{n\text{-SAM}}(t)\|_1 = (w(t) - w^*)^\top \left[\frac{1}{n} \sum_{i=1}^n x_i x_i^\top \right]^2 (w(t) - w^*),$$
$$\|I_{1\text{-SAM}}(t)\|_1 = (w(t) - w^*)^\top \left[\frac{1}{n} \sum_{i=1}^n \|x_i\|_2^2 x_i x_i^\top \right] (w(t) - w^*).$$

We can compare the typical operator norms of the random matrices that define the two quadratic forms. If we assume that $x_i \sim \mathcal{N}(0; I_d)$, then following the Bai-Yin's law, the operator norm of a Wishart matrix is with high probability $\|\frac{1}{n}\sum_{i=1}^n x_i x_i^\top\|_{\text{op}} \leq \frac{d}{n}$ and that with high probability, the squared norm of a Gaussian vector $\|x_i\|_2^2 \leq d$. Therefore we obtain that

$$\left[\frac{1}{n} \sum_{i=1}^n x_i x_i^\top \right]^2_{\text{op}} = \left[\frac{1}{n} \sum_{i=1}^n x_i x_i^\top \right]^2_{\text{op}} \leq \frac{d^2}{n^2};$$
$$\frac{1}{n} \sum_{i=1}^n \|x_i\|_2^2 x_i x_i^\top \Big\|_{\text{op}} \leq d \left\| \frac{1}{n} \sum_{i=1}^n x_i x_i^\top \right\|_{\text{op}} \leq \frac{d^2}{n}.$$

Therefore in the overparametrized regime ($d > n$), we typically have that $\frac{\|I_{1\text{-SAM}}(t)\|_1}{\|I_{n\text{-SAM}}(t)\|_1} \leq n$ and the biasing effect of 1-SAM would tend to be $\mathcal{O}(n)$ times better compared to n-SAM.

However, this first insight only enables to compare $I_{n\text{-SAM}}(t)$ and $I_{1\text{-SAM}}(t)$. It is not informative on the intrinsic biasing effect of n-SAM and 1-SAM. With this aim, we would like to relate the quantities $I_{n\text{-SAM}}(t)$ and $I_{1\text{-SAM}}(t)$ to the loss function evaluated in $w(t)$. Using the concentration of Wishart matrices, i.e. $\frac{1}{d}[XX^\top] \approx I$ for large dimension d, we have with high probability

$$\begin{aligned}
\|I_{n\text{-SAM}}(t)\|_1 &= \frac{1}{n^2}(w(t) - w^*)^\top X^\top X X^\top X (w(t) - w^*) \\
&= \frac{d}{n^2}(w(t) - w^*)^\top X^\top \frac{1}{d}[XX^\top] X (w(t) - w^*) \\
&\approx \frac{d}{n}(w(t) - w^*)^\top \frac{1}{n}[X^\top X](w(t) - w^*) \\
&= \frac{d}{n} L(w(t)).
\end{aligned} \tag{18}$$

Figure 13: Implicit bias of SAM on a sparse regression problem using a diagonal linear network with $d = 30$, $n = 20$, $x_i \sim N(0; I)$, $w = w_0 = 3$, $y_i = x_i^\top$. All methods are initialized at $= 0.01$ and used with step size $= 1/d$ and $= 1/d$. We can see that 1-SAM (SumMax) SGD converges to a solution which generalizes better (left plot) and enjoys a different implicit bias from the other methods. At the same time, all algorithms converge to a global minimum at a linear rate (right plot). The convergence speed is inversely proportional to the biasing effect.

Figure 14: A grid search over for full-batch n-SAM vs. 1-SAM ($= 0.05$, $= 15/d$ for all methods). We can see that even with the optimal , n-SAM generalizes much worse than 1-SAM which is coherent with our deep learning experiments in Fig. 1.

And using the concentration of Gaussian vectors, we also have that

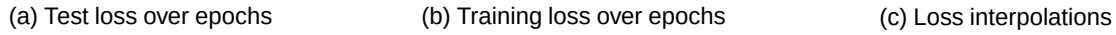
$$\begin{aligned} \|I_{1\text{-SAM}}(t)\|_1 &= (w(t) - w)^\top \frac{1}{n}\sum_{i=1}^n \|x_i\|^2 x_i x_i^\top (w(t) - w) \\ &\quad d(w(t) - w)^\top \frac{1}{n}\sum_{i=1}^n x_i x_i^\top (w(t) - w) \\ &= d\,L(w(t)). \end{aligned} \tag{19}$$

These approximations provide some intuition on why the biasing effect of 1-SAM and n-SAM can be related to the integral of the loss and that typically the difference is on the order of $1/n$. We let a formal derivation of these results as future work.

Experiments with stochastic ERM, n-SAM, 1-SAM. We provide an additional experiment to investigate the performance of stochastic implementations of the ERM, n-SAM and 1-SAM. As explained by Pesme et al. (2021), we observe in Fig. 13 that the stochastic implementations enjoy a better implicit bias than their deterministic counterparts. We note that the fact that small batch versions generalize better than full batch version is commonly observed in practice for deep networks Keskar et al. (2016). We let the characterization of the implicit bias of these stochastic implementations as future works.

Grid search over for n-SAM vs. 1-SAM. We note that for Fig. 6 and Fig. 13, we used a fixed which was the same for both n-SAM and 1-SAM. Tuning for each method separately can help to achieve a better test loss for both methods as shown in Fig. 14. We can see that 1-SAM still significantly outperforms ERM and n-SAM for the optimally chosen radius and that n-SAM leads only to marginal improvements.

Connection to the ERM $\to$ SAM and SAM $\to$ ERM experiment. Here we provide further details on the connection



(a) Test loss over epochs (b) Training loss over epochs (c) Loss interpolations

Figure 15: Test loss (a) and training loss (b) for full-batch ERM compared to ERM, 1-SAM and 1-SAM ! ERM on a diagonal linear network where we switch between the methods after 10k iterations. We can see that 1-SAM can quickly escape the worse-generalizing minimum found by ERM. Moreover, in (c) we show loss interpolations between ERM and 1-SAM and ERM that show that they are linearly connected and situated in the same basin.

between Theorem 1 and the empirical results in Fig. 9. First of all, we show in Fig. 15 that the same observations as we observed for deep networks also hold on a diagonal linear network. In this experiment, we used the initialization scale $= 0:05$, $_{1\text{-SAM}} = 0:175$, and $_{GD!\ 1\text{-SAM}} = 10:0$. We note that we had to take $_{GD!\ 1\text{-SAM}}$ significantly larger than $_{1\text{-SAM}}$ since after running GD, we are already near a global minimum where the gradients (which are also used for the ascent step of SAM) are very small so we need to increase the inner step size $_{GD!\ 1\text{-SAM}}$ to observe a difference. In addition, a loss interpolation between $w_{GD!\ 1\text{-SAM}}$ and w_{GD} reveals linear connectivity between the two found minima suggesting that both minima are situated in the same asymmetric basin, similarly to what we observed for deep networks in Fig. 10.

First we note that Theorem 1 can be trivially adapted to the case where SAM is used with varying inner step size $_t$, and we would therefore show that for diagonal linear networks, the key quantity determining the magnitude of the implicit bias for SAM is the integral of the step size $_s$ times the loss over the optimization trajectory $w(s)$, i.e., $\| _{1\text{-SAM-}\ s}\|_1 \quad d \int_0^{R_1} {}_s L(w(s))\,ds$ which leads to a smaller value in the exponent $_{1\text{-SAM-}\ s} = e^{\ _{1\text{-SAM-}\ s} + O(\ ^2)}$, thus decreasing the effective and biasing the ow to a sparser solution.

In the case of ERM ! 1-SAM, it amounts to consider a step size $_s = 0$ if $s < t$ and $_s =$ after the switch. Therefore the integral is taken only over the last epochs, and $\| _{1\text{-SAM-}t\text{-}1}\|_1 \quad d \int_t^{R_1} L(w(s))\,ds$ where the integral starts at the time step t. The resulting $\| _{1\text{-SAM-}t\text{-}1}\|_1$ is smaller than $\| _{1\text{-SAM}}\|_1$ but it can still be sufficient (especially, when using a higher as we do for Fig. 15) to improve the biasing effect so that it leads to noticeable improvements in generalization.

At the same time, for 1-SAM ! ERM, which amounts to consider a step size $_s =$ if $s < t$ and $_s = 0$ after the switch, the integral is already large enough due to the first 1000 epochs with SAM, leading to a term $\| _{1\text{-SAM-}0\text{-}t}\|_1 \quad d \int_0^{R_t} L(w(s))\,ds$ and switching back to ERM preserves the implicit bias due to a low enough effective . This explains why switching back to ERM does not negatively affect generalization of the model.

C. Convergence of the SAM Algorithm

In this section we provide proofs of convergence for SAM. We consider first the full-batch SAM algorithm and then its stochastic version.

C.1. Convergence of Full-Batch -SAM

We first consider the full-batch version of SAM, i.e., the following update rule:

$$w_{t+1} = w_t \quad r L(w_t + r L(w_t)) : \tag{20}$$

We note that this update rule is reminiscent of the extra-gradient algorithm (Korpelevich, 1977) but with an ascent in the inner step instead of a descent. Moreover, this update rule can also be seen as a realization of the general extrapolated gradient descent framework suggested in Lin et al. (2020). However, taking a constant step for extrapolation is not discussed there, and the convergence properties of the update rule from Eq. (20), to the best of our knowledge, have not been proven.

Summary of the convergence results. Let us first recall the definition of -smoothness which we will use in our proofs.

(A2') (-smoothness) There exists > 0 such that $\|\nabla L(w) - \nabla L(v)\| \leq \|w - v\|$ for all $w, v \in \mathbb{R}^d$.

When the function L is -smooth, convergence to stationary points can be obtained.

Theorem 6. Assume (A2'). For any $< 1/$ and $< 1/$, the iterates (20) satisfy for all $T > 0$:

$$\frac{1}{T}\sum_{t=0}^{T-1} \|\nabla L(w_t)\|^2 \leq \frac{2}{(1 -)T}(L(w_0) - L^*);$$

If, in addition, the function L satisfies (A3), then:

$$L(w_T) - L^* \leq \left(1 - \frac{(1 -)}{2}\right)^T (L(w_0) - L^*).$$

We can make the following remarks:

- We recover the rates of gradient descent but with constants increasing with the ascent step size
- The condition $< 1/$ is necessary since the point $w_t + 1/\|\nabla L(w)\|$ can be a local maximum of L. Such w would be a fixed point of the algorithm without being a stationary point of L.
- The proof crucially relies on the bound $\langle \nabla L(w_t + \nabla L(w_t)), \nabla L(w_t)\rangle \leq (1 -)\|\nabla L(w_t)\|^2$ which shows that the SAM step is well-aligned with the gradient step (see Lemma 7) and on a descent inequality similar to the classical one for gradient descent (see Lemma 8).
- For non-convex functions, full details are provided in Theorem 9. When the function satisfies in addition Polyak-Łojasiewicz inequality, a stronger result holds which is stated in Theorem 10.
- For convex functions, $\langle \nabla L(w_t + \nabla L(w_t)), \nabla L(w_t)\rangle \geq \|\nabla L(w_t)\|^2$ and convergence holds for any step size given that is small enough. Details are provided in Theorem 11.

Auxiliary Lemmas. The following lemma shows that the SAM update is well correlated with the gradient $\nabla L(w)$ and will be a cornerstone to our proof.

Lemma 7. Let L be a differentiable function and $w \in \mathbb{R}^d$. We have the following bound for any ≥ 0:

$$\langle \nabla L(w + \nabla L(w)), \nabla L(w)\rangle \leq (1 +)\|\nabla L(w)\|^2 \text{ where } = \begin{cases} \geq 0 & \text{if } L \text{ is -smooth,} \\ 0 & \text{if } L \text{ is convex} \\ < 0 & \text{if } L \text{ is -strongly convex} \end{cases}$$

Proof. We simply add and subtract a term $\|\nabla L(w)\|^2$ in order to make use of classical inequalities bounding $\langle \nabla L(w_1) - \nabla L(w_2), w_1 - w_2\rangle$ by $\|w_1 - w_2\|^2$ for smooth or convex functions and $w_1, w_2 \in \mathbb{R}^d$.

$$\begin{aligned}\langle \nabla L(w + \nabla L(w)), \nabla L(w)\rangle &= \langle \nabla L(w + \nabla L(w)) - \nabla L(w), \nabla L(w)\rangle + \|\nabla L(w)\|^2 \\ &= 1/\langle \nabla L(w + \nabla L(w)) - \nabla L(w), \nabla L(w)\rangle + \|\nabla L(w)\|^2 \\ &\leq (1 +)\|\nabla L(w)\|^2;\end{aligned}$$

where the last inequality is using that

$$\langle \nabla L(w_1) - \nabla L(w_2), w_1 - w_2\rangle \geq \|w_2 - w_1\|^2; \text{ where } = \begin{cases} \geq 0 & \text{if } L \text{ is -smooth,} \\ 0 & \text{if } L \text{ is convex} \\ < 0 & \text{if } L \text{ is -strongly convex} \end{cases}$$

□

The next lemma shows that the decrease of function values of the SAM algorithm defined in Eq. (20) can be controlled similarly as in the case of gradient descent (Nesterov, 2004).

Lemma 8. Assume (A2'). For any $\rho \leq 1/\beta$, the iterates (20) satisfy for all $t \geq 0$:

$$L(w_{t+1}) \leq L(w_t) - \eta(1 - \tfrac{\eta\beta}{2})\|\nabla L(w_t)\|^2.$$

If, in addition, the function L satisfies (A3) with potentially $\mu = 0$, then for all $\eta, \rho > 0$ such that $\eta(2 - \eta\beta) \geq 2\rho$, we have

$$L(w_{t+1}) \leq L(w_t) - \tfrac{\eta\mu}{2} + \Big(1 - \tfrac{\eta\mu}{2}\Big)^2 \|\nabla L(w_t)\|^2.$$

We note that the constraints on the step size are different depending on the assumptions on the function L. In the non-convex case, η has to be smaller than $1/\beta$, whereas in the convex case, it has to be smaller than $2/\beta$.

Proof. Let us define by $w_{t+1/2} = w_t + \rho \nabla L(w_t)$ the SAM ascent step. Using the smoothness of the function L (Assumption (A2')), we obtain

$$L(w_{t+1}) \leq L(w_t) - \langle \nabla L(w_{t+1/2}), \rho \nabla L(w_t)\rangle + \frac{\beta}{2}\|\rho \nabla L(w_{t+1/2})\|^2.$$

The main trick is to use the binomial squares

$$\|\nabla L(w_{t+1/2})\|^2 = \|\nabla L(w_t)\|^2 + \|\nabla L(w_{t+1/2}) - \nabla L(w_t)\|^2 - 2\langle \nabla L(w_{t+1/2}) - \nabla L(w_t), \nabla L(w_t)\rangle,$$

to bound

$$\begin{aligned}
L(w_{t+1}) &\leq L(w_t) - \langle \nabla L(w_{t+1/2}), \rho \nabla L(w_t)\rangle + \frac{\beta}{2}\|\rho \nabla L(w_{t+1/2})\|^2 \\
&= L(w_t) - \frac{\rho^2}{2}\|\nabla L(w_t)\|^2 + \frac{\rho^2}{2}\|\nabla L(w_{t+1/2}) - \nabla L(w_t)\|^2 - (1 - \rho\beta)\langle \nabla L(w_{t+1/2}), \rho \nabla L(w_t)\rangle \\
&\leq L(w_t) - [1 - \tfrac{\rho\beta}{2}(1 - \rho\beta)^2]\|\nabla L(w_t)\|^2,
\end{aligned}$$

where we have used Lemma 7 and that $\|\nabla L(w_{t+1/2}) - \nabla L(w_t)\|^2 \leq \beta^2 \|w_{t+1/2} - w_t\|^2 = \beta^2 \rho^2 \|\nabla L(w_t)\|^2$.

If, in addition, the function L is convex then we can use its co-coercivity (Nesterov, 2004) to bound $\|\nabla L(w_{t+1/2}) - \nabla L(w_t)\|^2 \leq \beta \langle \nabla L(w_{t+1/2}) - \nabla L(w_t), w_{t+1/2} - w_t\rangle$ and obtain a tighter bound:

$$\begin{aligned}
L(w_{t+1}) &\leq L(w_t) - \langle \nabla L(w_{t+1/2}), \rho \nabla L(w_t)\rangle + \frac{\beta}{2}\|\rho \nabla L(w_{t+1/2})\|^2 \\
&= L(w_t) - \frac{\rho^2}{2}\|\nabla L(w_t)\|^2 + \frac{\rho^2}{2}\|\nabla L(w_{t+1/2}) - \nabla L(w_t)\|^2 - (1 - \rho\beta)\langle \nabla L(w_{t+1/2}), \rho \nabla L(w_t)\rangle \\
&\leq L(w_t) - (1 - \tfrac{\rho\beta}{2})\|\nabla L(w_t)\|^2 - (1 - \tfrac{\rho\beta}{2})\langle \nabla L(w_{t+1/2}) - \nabla L(w_t), \rho \nabla L(w_t)\rangle \\
&\leq L(w_t) - (1 - \tfrac{\rho\beta}{2} + (1 - \tfrac{\rho\beta}{2})^2)\|\nabla L(w_t)\|^2,
\end{aligned}$$

where we have used Lemma 7. □

Convergence proofs. Using the previous Lemma 8 recursively, we can bound the average gradient value of the iterates (20) of SAM algorithm and ensure convergence to stationary points.

Theorem 9. Assume (A2'). For any $\eta < 1/\beta$ and $\rho < 1/\beta$, the iterates (20) satisfies for all $T \geq 0$:

$$\frac{1}{T}\sum_{t=0}^{T} \|\nabla L(w_t)\|^2 \leq \frac{L(w_0) - L(w_T)}{T\,\eta(1 - \rho\beta)[1 - \frac{\rho\beta}{2}(1 - \rho\beta)]}.$$

Proof. Using the Lemma 8 we obtain

$$(1 \quad) \; 1 \quad \frac{}{2}(1 \quad) \; \| r \; L(w_t)\|^2 \quad L(w_t) \quad L(w_{t+1}):$$

And summing these inequalities for $t = 0; : : : ; T \quad 1$ yields

$$\frac{1}{T} \sum_{t=0}^{T \; 1} \| r \; L(w_t)\|^2 \quad \frac{L(w_0) \quad L(w_T)}{T \; (1 \quad)[1 \quad \frac{}{2}(1 \quad)]}:$$

□

When the function L additionally satisfies a Polyak-Lojasiewicz condition (A3), linear convergence of the function value to the minimum function value can be obtained. This is the object of the following theorem:

Theorem 10. Assume (A2') and (A3). For any $< 1=$ and $< 1=$, the iterates (20) satisfies for all $T \quad 0$:

$$L(w_t) \quad L \quad 1 \quad 2 \quad (1 \quad) \; 1 \quad \frac{}{2}(1 \quad) \quad {}^t (L(w_0) \quad L):$$

Proof. Using the Lemma 8 and that the function is Polyak-Lojasiewicz (Assumption (A3)) we obtain

$$L(w_{t+1}) \quad L(w_t) \quad 2 \quad (1 \quad L) \; 1 \quad \frac{}{2}(1 \quad L) \; (L(w_t) \quad L):$$

And subtracting the optimal value L we get

$$\begin{aligned} L(w_t) \quad L \quad & 1 \quad 2 \quad (1 \quad) \; 1 \quad \frac{}{2}(1 \quad) \quad (L(w_{t \; 1}) \quad L) \\ & 1 \quad 2 \quad (1 \quad) \; 1 \quad \frac{}{2}(1 \quad) \quad {}^t (L(w_0) \quad L): \end{aligned}$$

□

When the function L is convex, convergence of the average of the iterates can be proved.

Theorem 11. Assume (A2') and L convex. For any step sizes and such that $(1 + \quad) < 2$, then the averaged $w_T = \frac{1}{T} \sum_{t=0}^{T \; 1} w_t$ of the iterates (20) satisfies for all $T \quad 0$:

$$L(w_T) \quad L \quad \frac{2 \quad + 1}{(2 \quad (1 + \quad))T} \|w_0 \quad w \|^2;$$

If, in addition, the function L is -strongly convex, then:

$$\|w_T \quad w \|^2 \quad 1 \quad (2 \quad (1 + \quad)) \quad {}^T (2 \quad + 1) \|w_0 \quad w \|^2:$$

The proof is using a different astute Lyapunov function which works for the non-strongly convex case.

Proof. Let us define by $V_t = [L(w_t) \quad L(w)] + \frac{1}{2}\|w_t \quad w \|^2$ and by $w_{t+1 \; =2} = w_t + \quad r \; L(w_t)$ the SAM ascent step.

$$\begin{aligned} V_{t+1} \quad V_t \quad & \quad \langle r \; L(w_{t+1 \; =2}); w_t \quad w \rangle \quad \langle r \; L(w_{t+1 \; =2}); r \; L(w_t)\rangle + \frac{^2}{2}(1 + \quad)\| r \; L(w_{t+1 \; =2})\|^2 \\ = \quad & \quad \langle r \; L(w_{t+1 \; =2}); w_t + \quad r \; L(w_t) \quad w \rangle + \frac{^2}{2}(1 + \quad)\| r \; L(w_{t+1 \; =2})\|^2 \\ = \quad & \quad \langle r \; L(w_{t+1 \; =2}); w_{t+1 \; =2} \quad w \rangle + \frac{^2}{2}(1 + \quad)\| r \; L(w_{t+1 \; =2})\|^2 \\ & \quad (1 \quad \frac{}{2}(1 + \quad))\langle r \; L(w_{t+1 \; =2}); w_{t+1 \; =2} \quad w \rangle: \end{aligned}$$

If L is convex then $L(w_{t+1=2}) \quad L(w) \quad \langle \nabla L(w_{t+1=2}); w_{t+1=2} \quad w \rangle$ and therefore we obtain

$$- \quad 1 \quad \frac{}{2}(1 + \quad) \quad L(w_{t+1=2}) \quad L(w) \quad V_t \quad V_{t+1}:$$

Using the definition of $w_{t+1=2}$ we always have that $L(w_{t+1=2}) \quad L(w_t) + \quad \|\nabla L(w_t)\|^2$ therefore

$$- \quad 1 \quad \frac{}{2}(1 + \quad) \quad (L(w_t) \quad L(w)) \quad V_t \quad V_{t+1}:$$

And taking the sum and using Jensen inequality we finally obtain:

$$L\left(\frac{1}{T}\sum_{t=0}^{T} w_t\right) \quad L(w) \quad \frac{V_0 \quad V_{T+1}}{T - (1 \quad \frac{}{2}(1 + \quad))}:$$

If L is $$-strongly convex, we use that $\langle \nabla L(w_{t+1=2}); w_{t+1=2} \quad w \rangle \quad \|w_{t+1=2} \quad w\|^2$ to obtain

$$\begin{aligned}
\|w_{t+1=2} \quad w\|^2 = \|w_t + \quad \nabla L(w_t) \quad w\|^2 &= \|w_t \quad w\|^2 + 2 \langle \nabla L(w_t); w_t \quad w \rangle + {}^2\|\nabla L(w_t)\|^2 \\
&\quad \|w_t \quad w\|^2 + 2 \langle \nabla L(w_t); w_t \quad w \rangle \\
&\quad \|w_t \quad w\|^2 + 2 [L(w_t) \quad L(w)] \\
&\quad 2 V_t:
\end{aligned}$$

Therefore we have

$$V_{t+1} \quad (1 \quad (2 \quad (1 + \quad))) V_t \quad (1 \quad (2 \quad (1 + \quad)))^{t+1} V_0:$$

□

C.2. Convergence of Stochastic SAM

C.2.1. Convergence of $$-SAM

When the SAM algorithm is implemented with the SAM objective as optimization objective, two different batches are used in the ascent and descent steps. We obtain the $$-SAM algorithm defined as

$$w_{t+1} = w_t \quad \frac{}{b}\sum_{i \in I_t} \nabla \ell_i\left(w_t + \frac{}{b}\sum_{i \in J_t} \nabla \ell_i(w_t)\right); \tag{21}$$

where I_t and J_t are two different mini-batches of data of size b. For this variant of the SAM algorithm, we obtain the following convergence result.

Theorem 12. Assume (A1), (A2') for the iterates (21). For any $T \quad 0$ and for step sizes $_t = \frac{1}{\sqrt{T}}$ and $_t = \frac{1}{T^{1=4}}$, we have:

$$\frac{1}{T} E\left[\sum_{t=0}^{T} \|\nabla L(w_t)\|^2\right] \quad \frac{4}{\sqrt{T}}(L(w_0) \quad L) + \frac{8}{b\sqrt{T}};$$

In addition, under (A2), with step sizes $_t = \min\{\frac{8t+4}{3(t+1)^2}; \frac{1}{2}\}$ and $_t = \quad _t =$:

$$E[L(w_T)] \quad L \quad \frac{3 \ ^2(L(w_0) \quad L)}{ ^2T^2} + \frac{22 \ ^2}{b \ ^2T}$$

We obtain the same convergence result as in Theorem 2, but under the relaxed smoothness assumption (A2').

As in the deterministic case, the proof relies on two lemmas which shows that the SAM update is well correlated with the gradient and that the decrease of function values can be controlled.

Auxiliary lemmas. The following lemma shows that the SAM update is well correlated with the gradient. Let us denote by $\nabla L_{t+1}(w) = \frac{1}{b}\sum_{i \in I_t} \nabla \ell_i(w)$, $\nabla L_{t+1=2}(w) = \frac{1}{b}\sum_{i \in J_t} \nabla \ell_i(w)$, and $w_{t+1=2} = w_t + \quad \nabla L_{t+1=2}(w_t)$ the SAM ascent step.

Lemma 13. Assume (A1) and (A2). Then for all $\rho > 0$, $t \geq 0$ and $w \in \mathbb{R}^d$,

$$\mathbb{E}\left\langle \nabla L_{t+1}(w + \rho \nabla L_{t+1/2}(w)), \nabla L(w)\right\rangle \leq (1/2 - \rho)\|\nabla L(w)\|^2 + \frac{\rho^2 \beta^2 \sigma^2}{2}.$$

The proof is similar to the proof of Lemma 7. Only the stochasticity of the noisy gradients has to be taken into account. For this goal, we consider instead the update which would have been obtained without noise, and bound the remainder using the bounded variance assumption (A1).

Proof. Let us denote by $\hat{w} = w + \rho \nabla L(w)$, the true gradient step. We first add and subtract $\nabla L_{t+1}(\hat{w})$

$$\left\langle \nabla L_{t+1}(w + \rho \nabla L_{t+1/2}(w)), \nabla L(w)\right\rangle = \left\langle \nabla L_{t+1}(w + \rho \nabla L_{t+1/2}(w)) - \nabla L_{t+1}(\hat{w}), \nabla L(w)\right\rangle + \left\langle \nabla L_{t+1}(\hat{w}), \nabla L(w)\right\rangle.$$

We bound the two terms separately. We use the smoothness of L (Assumption (A2')) to bound the first term:

$$\begin{aligned}
\mathbb{E}\left\langle \nabla L_{t+1}(w + \rho \nabla L_{t+1/2}(w)) - \nabla L_{t+1}(\hat{w}), \nabla L(w)\right\rangle &= \mathbb{E}\left\langle \nabla L(w + \rho \nabla L_{t+1/2}(w)) - \nabla L(\hat{w}), \nabla L(w)\right\rangle \\
&\leq \frac{1}{2}\mathbb{E}\|\nabla L(w + \rho \nabla L_{t+1/2}(w)) - \nabla L(\hat{w})\|^2 + \frac{1}{2}\|\nabla L(w)\|^2 \\
&\leq \frac{\beta^2}{2}\mathbb{E}\|w + \rho \nabla L_{t+1/2}(w) - \hat{w}\|^2 + \frac{1}{2}\|\nabla L(w)\|^2 \\
&\leq \frac{\rho^2 \beta^2}{2}\mathbb{E}\|\nabla L_{t+1/2}(w) - \nabla L(w)\|^2 + \frac{1}{2}\|\nabla L(w_t)\|^2 \\
&\leq \frac{\rho^2 \beta^2 \sigma^2}{2b} + \frac{1}{2}\|\nabla L(w)\|^2,
\end{aligned}$$

where we have used that the variance of a mini-batch of size b is bounded by σ^2/b. Note that this term can be equivalently bounded by $\leq \beta^2 \rho^2 \|\nabla L(w)\|^2$ if needed. For the second term, we directly apply Lemma 7 to obtain

$$\mathbb{E}\left\langle \nabla L_{t+1}(\hat{w}), \nabla L(w)\right\rangle = \mathbb{E}\left\langle \nabla L(\hat{w}), \nabla L(w)\right\rangle \leq (1 - \rho)\|\nabla L(w)\|^2.$$

□

The next lemma shows that the decrease of function values of stochastic SAM can be controlled similarly as for standard stochastic gradient descent.

Lemma 14. Let us assume (A1, A2') then for all $\rho \leq \frac{1}{2}$ and $\eta \leq \frac{1}{2\beta}$, the iterates (21) satisfies

$$\mathbb{E}\, L(w_{t+1}) \leq \mathbb{E}\, L(w_t) - \frac{\eta}{4}\mathbb{E}\|\nabla L(w_t)\|^2 + \eta^2\beta\left(\rho^2\beta\sigma^2 + \sigma^2\right).$$

This lemma is analogous to Lemma 8 in the stochastic case. The proof is very similar, with the slight difference that Lemma 13 is used instead of Lemma 7.

Proof. Let us define by $w_{t+1/2} = w_t + \rho \nabla L_{t+1/2}(w_t)$. Using the smoothness of the function (A2), we obtain

$$L(w_{t+1}) \leq L(w_t) - \eta\left\langle \nabla L_{t+1}(w_{t+1/2}), \nabla L(w_t)\right\rangle + \frac{\eta^2\beta}{2}\|\nabla L_{t+1}(w_{t+1/2})\|^2.$$

Taking the expectation and using that the variance is bounded (A1) yields to

$$\begin{aligned}
\mathbb{E}\, L(w_{t+1}) &\leq \mathbb{E}\, L(w_t) - \eta\mathbb{E}\left\langle \nabla L(w_{t+1/2}), \nabla L(w_t)\right\rangle + \frac{\eta^2\beta}{2}\mathbb{E}\|\nabla L_{t+1}(w_{t+1/2})\|^2 \\
&\leq \mathbb{E}\, L(w_t) - \eta\mathbb{E}\left\langle \nabla L(w_{t+1/2}), \nabla L(w_t)\right\rangle + \eta^2\beta\mathbb{E}\|\nabla L_{t+1}(w_{t+1/2}) - \nabla L(w_{t+1/2})\|^2 + \eta^2\beta\mathbb{E}\|\nabla L(w_{t+1/2})\|^2 \\
&\leq \mathbb{E}\, L(w_t) - \eta\mathbb{E}\left\langle \nabla L(w_{t+1/2}), \nabla L(w_t)\right\rangle + \eta^2\beta\left(\frac{\sigma^2}{b} + \sigma^2\right)\mathbb{E}\|\nabla L(w_{t+1/2})\|^2.
\end{aligned}$$

The main trick is still to use the binomial squares

$$\|\nabla L(w_{t+1,2})\|^2 = \|\nabla L(w_t)\|^2 + \|\nabla L(w_{t+1,2}) - \nabla L(w_t)\|^2 + 2\langle \nabla L(w_{t+1,2}) - \nabla L(w_t), \nabla L(w_t)\rangle$$

to bound

$$\begin{aligned}
\mathbb{E}L(w_{t+1}) &\le \mathbb{E}L(w_t) - \mathbb{E}\langle \nabla L(w_{t+1,2}), \nabla L(w_t)\rangle + \frac{\eta^2}{2}\mathbb{E}\|\nabla L(w_{t+1,2})\|^2 + \frac{\eta^2\rho^2}{2b} \\
&= \mathbb{E}L(w_t) - \frac{\eta}{2}\mathbb{E}\|\nabla L(w_t)\|^2 - \frac{\eta}{2}\mathbb{E}\|\nabla L(w_{t+1,2}) - \nabla L(w_t)\|^2 \\
&\quad + \frac{\eta}{2}\mathbb{E}\|\nabla L(w_{t+1,2}) - \nabla L(w_t)\|^2 - \mathbb{E}\langle \nabla L(w_{t+1,2}), \nabla L(w_t)\rangle + \frac{\eta^2\rho^2}{2b} \\
&= \mathbb{E}L(w_t) - \frac{\eta}{2}\mathbb{E}\|\nabla L(w_t)\|^2 - \frac{\eta}{2}\mathbb{E}\|\nabla L(w_{t+1,2}) - \nabla L(w_t)\|^2 + \frac{\eta}{2}\mathbb{E}\|\nabla L(w_{t+1,2}) - \nabla L(w_t)\|^2 \\
&\quad + \frac{\eta}{2}\mathbb{E}\|\nabla L(w_{t+1,2}) - \nabla L(w_t)\|^2 - \mathbb{E}\langle \nabla L(w_{t+1,2}) - \nabla L(w_t), \nabla L(w_t)\rangle + \frac{\eta^2\rho^2}{2b} \\
&\le \mathbb{E}L(w_t) - \frac{\eta}{2}\mathbb{E}\|\nabla L(w_t)\|^2 - \frac{\eta}{2}\mathbb{E}\|\nabla L(w_{t+1,2}) - \nabla L(w_t)\|^2 + \frac{\eta}{2}\mathbb{E}\|\nabla L(w_{t+1,2}) - \nabla L(w_t)\|^2 \\
&\quad + \frac{\eta}{2}\mathbb{E}\|\nabla L(w_{t+1,2}) - \nabla L(w_t)\|^2 + \frac{\eta}{2}\mathbb{E}\|\nabla L(w_t)\|^2 + \frac{\eta^2\rho^2}{2b} \\
&= \mathbb{E}L(w_t) - \frac{\eta}{2}\mathbb{E}\|\nabla L(w_t)\|^2 + \eta^3 L^2\mathbb{E}\|w_{t+1,2} - w_t\|^2 + \frac{\eta^2\rho^2}{2b} \\
&\le \mathbb{E}L(w_t) - \frac{\eta}{2}\mathbb{E}\|\nabla L(w_t)\|^2 + \eta^3 L^2\rho^2\mathbb{E}\|\nabla L_{t+1,2}(w_t)\|^2 + \frac{\eta^2\rho^2}{2b} \\
&\le L(w_t) - \frac{\eta}{2}[1 - 2\eta^2 L^2(1 + 2\eta^2 L^2)]\mathbb{E}\|\nabla L(w_t)\|^2 + \frac{\eta^2\rho^2}{2b}[1 + 2\eta^2 L^2(1 + 2\eta^2 L^2)]
\end{aligned}$$

where we have used Lemma 13 and that $\|\nabla L(w_{t+1,2}) - \nabla L(w_t)\|^2 \le \rho^2\|w_{t+1,2} - w_t\|^2$. □

Using Lemma 14 we directly obtain the following convergence result.

Theorem 15. Assume (A1) and (A2'). For $\eta \le 1/(2L)$ and $\rho \le 1/(2L)$, the iterates (4) satisfies:

$$\frac{1}{T}\sum_{t=0}^{T-1}\mathbb{E}\|\nabla L(w_t)\|^2 \le 4\frac{L(w_0) - \mathbb{E}L(w_T)}{\eta T} + 4\eta^2\rho^2(L^2 + \sigma^2/b).$$

This theorem gives the first part of Theorem 12. The proof of the stronger result obtained when the function is in addition PL (Assumption (A3)) is similar to the proof of Theorem 3.2 of Gower et al. (2019), only the constants are changing.

C.2.2. Convergence of m-SAM

In the m-SAM algorithm, the same batch is used in the ascent and descent steps unlike in the SAM algorithm analyzed above. We obtain then iterates (4) for which we have stated the convergence result in Theorem 2 in the main part. The proof follows the same lines as above with the minor difference that we are assuming instead that individual gradients $\nabla \ell_i$ are Lipschitz (Assumption (A2)) to control the alignment of the expected SAM direction. Let us denote by $\nabla L_t(w) = \frac{1}{b}\sum_{i\in\mathcal{J}_t}\nabla \ell_i(w)$.

Lemma 16. Assume (A1-2). Then we have for all $w \in \mathbb{R}^d$, $\eta > 0$ and $t \ge 0$

$$\mathbb{E}\langle \nabla L_t(w + \rho\nabla L_t(w)), \nabla L(w)\rangle \ge (1/2 - \eta L)\|\nabla L(w)\|^2 - \frac{\eta\rho^2 L^2}{2b}.$$

The proof is very similar to the proof of Lemma 13. The only difference is that the Assumption (A2) is used instead of (A2').

Proof. Let us denote by $\hat{w} = w + \rho\nabla L(w)$, the true gradient step. We first add and subtract $\nabla L_t(\hat{w})$

$$\langle \nabla L_t(w + \rho\nabla L_t(w)), \nabla L(w)\rangle = \langle \nabla L_t(w + \rho\nabla L_t(w)) - \nabla L_t(\hat{w}), \nabla L(w)\rangle + \langle \nabla L_t(\hat{w}), \nabla L(w)\rangle.$$

We bound the two terms separately. We use the smoothness of L_t to bound the first term (Assumption (A2)):

$$\begin{aligned}
&\langle \nabla L_t(w + \rho \nabla L_t(w)) - \nabla L_t(\hat{w}), \nabla L(\hat{w})\rangle \le \frac{1}{2}\|\nabla L_t(w + \rho \nabla L_t(w)) - \nabla L_t(\hat{w})\|^2 + \frac{1}{2}\|\nabla L(\hat{w})\|^2 \\
&\le \frac{\beta^2}{2}\|w + \rho \nabla L_t(w) - \hat{w}\|^2 + \frac{1}{2}\|\nabla L(\hat{w})\|^2 \\
&\le \frac{\beta^2 \rho^2}{2}\|\nabla L_t(w) - \nabla L(w)\|^2 + \frac{1}{2}\|\nabla L(w)\|^2.
\end{aligned}$$

And taking the expectation, we obtain:

$$\mathbb{E}\langle \nabla L_t(w + \rho \nabla L_t(w)) - \nabla L_t(\hat{w}), \nabla L(\hat{w})\rangle \le \frac{\beta^2 \rho^2 \sigma^2}{2b} + \frac{1}{2}\mathbb{E}\|\nabla L(w)\|^2.$$

For the second term, we apply directly Lemma 7

$$\mathbb{E}\langle \nabla L_t(\hat{w}), \nabla L(w_t)\rangle = \langle \nabla L(\hat{w}), \nabla L(w)\rangle \le (1 - \mu)\|\nabla L(w)\|^2.$$

Assembling the two inequalities yields the result. □

The next lemma shows that the decrease of function values of the SAM algorithm can be controlled similarly as in the case of gradient descent. It is analogous to Lemma 14 when different batches are used in both the ascent and descent steps of SAM algorithm.

Lemma 17. Assume (A1-2). For all $\rho \le \frac{1}{\beta}$ and $\eta \le \frac{1}{4\beta}$, the iterates (4) satisfy

$$\mathbb{E}L(w_{t+1}) \le \mathbb{E}L(w_t) - \frac{3\eta}{8}\mathbb{E}\|\nabla L(w_t)\|^2 + \frac{\eta^2\beta\sigma^2}{b}(\rho\beta + 2\eta\beta^2).$$

Proof. Let us define by $w_{t+1/2} = w_t + \rho \nabla L_{t+1}(w_t)$. Using the smoothness of the function L which is implied by (A2), we obtain

$$L(w_{t+1}) \le L(w_t) - \eta\langle \nabla L_{t+1}(w_{t+1/2}), \nabla L(w_t)\rangle + \frac{\eta^2\beta}{2}\|\nabla L_{t+1}(w_{t+1/2})\|^2.$$

We still use the binomial squares

$$\|\nabla L_{t+1}(w_{t+1/2})\|^2 = \|\nabla L(w_t)\|^2 + \|\nabla L_{t+1}(w_{t+1/2}) - \nabla L(w_t)\|^2 + 2\langle \nabla L_{t+1}(w_{t+1/2}), \nabla L(w_t)\rangle$$

and bound $L(w_{t+1})$ by

$$\begin{aligned}
L(w_{t+1}) &\le L(w_t) - \frac{\eta^2\beta}{2}\|\nabla L(w_t)\|^2 + \frac{\eta^2\beta}{2}\|\nabla L_{t+1}(w_{t+1/2}) - \nabla L(w_t)\|^2 - (1 - \eta\beta)\langle \nabla L_{t+1}(w_{t+1/2}), \nabla L(w_t)\rangle \\
&\le L(w_t) - \frac{\eta^2\beta}{2}\|\nabla L(w_t)\|^2 + \eta^2\beta\|\nabla L_{t+1}(w_{t+1/2}) - \nabla L_{t+1}(w_t)\|^2 + \eta^2\beta\|\nabla L_{t+1}(w_t) - \nabla L(w_t)\|^2 \\
&\quad - (1 - \eta\beta)\langle \nabla L_{t+1}(w_{t+1/2}), \nabla L(w_t)\rangle \\
&\le L(w_t) - \frac{\eta^2\beta}{2}\|\nabla L(w_t)\|^2 + \eta^2\beta^3\|w_{t+1/2} - w_t\|^2 + \eta^2\beta\|\nabla L_{t+1}(w_t) - \nabla L(w_t)\|^2 \\
&\quad - (1 - \eta\beta)\langle \nabla L_{t+1}(w_{t+1/2}), \nabla L(w_t)\rangle \\
&= L(w_t) - \frac{\eta^2\beta}{2}\|\nabla L(w_t)\|^2 + \eta^2\beta^3\rho^2\|\nabla L_{t+1}(w_t)\|^2 + \eta^2\beta\|\nabla L_{t+1}(w_t) - \nabla L(w_t)\|^2 \\
&\quad - (1 - \eta\beta)\langle \nabla L_{t+1}(w_{t+1/2}), \nabla L(w_t)\rangle \\
&= L(w_t) - \frac{\eta^2\beta}{2}(1 - 4\eta^2\beta^2\rho^2)\|\nabla L(w_t)\|^2 + \eta^2\beta(1 + 2\eta^2\beta^2\rho^2)\|\nabla L_{t+1}(w_t) - \nabla L(w_t)\|^2 \\
&\quad - (1 - \eta\beta)\langle \nabla L_{t+1}(w_{t+1/2}), \nabla L(w_t)\rangle
\end{aligned}$$

Taking the expectation and using Lemma 16, we obtain

$$\begin{aligned}
\mathbb{E}\,L(w_{t+1}) &\le \mathbb{E}\,L(w_t) - \frac{\eta^2}{2}(1 - 4\rho^2\beta^2)\,\mathbb{E}\|\nabla L(w_t)\|^2 + \rho^2(1+2\rho^2\beta^2)\,\mathbb{E}\|\nabla L_{t+1}(w_t) - \nabla L(w_t)\|^2 \\
&\quad - (1-\eta\beta)\,\mathbb{E}\langle \nabla L_{t+1}(w_{t+1/2}) - \nabla L(w_t),\, \nabla L(w_t)\rangle \\
&\le \mathbb{E}\,L(w_t) - \frac{\eta^2}{2}(1 - 4\rho^2\beta^2)\,\mathbb{E}\|\nabla L(w_t)\|^2 + \rho^2(1+2\rho^2\beta^2)\,\sigma^2/b \\
&\quad + (1-\eta\beta)(1/2)\,\mathbb{E}\|\nabla L(w_t)\|^2 + (1-\eta\beta)\frac{\rho^2\beta^2\sigma^2}{2b} \\
&\le \mathbb{E}\,L(w_t) - \frac{\eta^2}{2}(1 - 4\rho^2\beta^2)\,\mathbb{E}\|\nabla L(w_t)\|^2 + \rho^2(1+2\rho^2\beta^2)\,\sigma^2/b \\
&\quad - \frac{\eta}{2}(1 - 2\eta\beta(1-\rho^2\beta^2))\,\mathbb{E}\|\nabla L(w_t)\|^2 + \eta^2\sigma^2/(2b)\Big[\rho^2\beta^2 + \frac{\eta\beta}{2}(1+3\rho^2\beta^2)\Big].
\end{aligned}$$

□

Using Lemma 17 we directly obtain the main convergence result for SAM.

Theorem 18. Assume (A1-2). For $\rho \le \frac{1}{\beta}$ and $\eta \le \frac{1}{4\beta}$, the iterates (4) satisfy:

$$\frac{1}{T}\,\mathbb{E}\Big[\sum_{t=0}^{T-1} \|\nabla L(w_t)\|^2\Big] \le \frac{8}{3\eta T}(L(w_0) - \mathbb{E}\,L(w_T)) + \frac{8\rho^2(\eta\beta + \rho^2\beta^2)}{3}\sigma^2/b.$$

In addition, under (A3), with step sizes $\eta_t = \min\{\frac{8t+4}{3\rho(t+1)^2}, \frac{1}{2\beta}\}$ and $\rho_t = \rho/\eta_t = \rho$:

$$\mathbb{E}[L(w_T)] - L^* \le \frac{3\rho^2\beta(L(w_0) - L^*)}{\rho^2 T^2} + \frac{22\rho^2\beta\sigma^2}{\rho^2 b T}.$$

Proof. The first bound directly comes from Lemma 17. The second bound is similar to the proof of Theorem 3.2 of Gower et al. (2019), only the constants are changing. □

Finally, we note that Theorem 2 is a direct consequence of Theorem 18 with $\rho_t = \frac{1}{\sqrt{T}}$, $\eta_t = \frac{1}{T^{1/4}}$ and slightly simplified constants.

D. Experimental Details

Training details for deep networks. In all experiments, we train deep networks using SGD with step size 0.1, momentum 0.9, and ℓ_2-regularization parameter $= 0.0005$. We perform experiments on CIFAR-10 and CIFAR-100 (Krizhevsky & Hinton, 2009) where for all experiments we apply basic data augmentations: random image crops and mirroring. We use batch size 128 for most experiments except when it is mentioned otherwise. We use a pre-activation ResNet-18 (He et al., 2016) for CIFAR-10 and ResNet-34 on CIFAR-100 with a width factor 64 and piece-wise constant learning rates (with a 10-times decay at 50% and 75% epochs). We train all models for 200 epochs except those in Sec. 4.3 and Sec. 5.2 for which we use 1000 epochs. We use batch normalization for most experiments, except when it is explicitly mentioned otherwise as, for example, in the experiments where we aim to compute sharpness and for this we use networks with group normalization.

For all experiments involving SAM, we select the best perturbation radius ρ based on a grid search over $\{0.025, 0.05, 0.1, 0.2, 0.3, 0.4\}$. In most cases, the optimal ρ is equal to 0.1 while in the ERM + SAM experiment, it is equal to $\rho = 0.4$ for CIFAR-10 and $\rho = 0.2$ for CIFAR-100. We note that using a higher ρ in this case is coherent with the experiments on diagonal linear networks which also required a higher ρ. For all experiments with SAM, we use a single GPU, so we do not implicitly rely on lower m-sharpness in m-SAM. The only exception where m is smaller than the batch size is the experiments shown in Fig. 4 and Fig. 16. Regarding SAM in Fig. 1, we implement it by doing the ascent step on a different batch compared to the descent step, i.e., as described in our convergence analysis part in Eq. (21).

Sharpness computation. We compute m-sharpness on 1024 training points (i.e., by averaging over 1024/m batches) of CIFAR-10 or CIFAR-100 using 100 iterations of projected gradient ascent using a step size of $\eta = 0.1$. For each iteration, we normalize the updates by the ℓ_2 gradient norm.

Con dence intervals on plots. Many experimental results are replicated over different random seeds used for training. We show the results using the mean and 95% bootstrap con dence intervals which is the standard way to show such results in the seaborn library Waskom (2021).

Code and computing infrastructure. The code of our experiments is publicly available[4]. We perform all our experiments with deep networks on a single NVIDIA V100 GPU with 32GB of memory. Since most of our experiments involved a grid search over the perturbation radius and replication over multiple random seeds, we could not do the same at the ImageNet scale due to our limited computational resources.

E. Additional Deep Learning Experiments

In this section, we show additional experimental results complementary to those presented in the main part. In particular, we provide multiple ablation study related to the role of m in m-SAM, batch size, and model width. We also provide additional experiments on the evolution of sharpness over training using training time and test time batch normalization, training loss of ERM vs. SAM models, and the performance under label noise for standard and unnormalized SAM.

E.1. The Effect of m in m-SAM

We show the results of SAM for different m in m-SAM (with a fixed batch size 256) in Fig. 16. We note that in this experiment, we used group normalization instead of batch normalization like, for example, in Fig. 1, so the exact test error values should not be compared between these two figures. We observe from Fig. 16, that the generalization improvement is larger for smaller m and it is continuous in m. We also note that a similar experiment has been done in the original SAM paper (Foret et al., 2021). Here, we additionally verified this finding on an additional dataset (CIFAR-100) and for networks trained without batch normalization (which may have had an extra regularization effect as we discussed in Sec. 4.1).

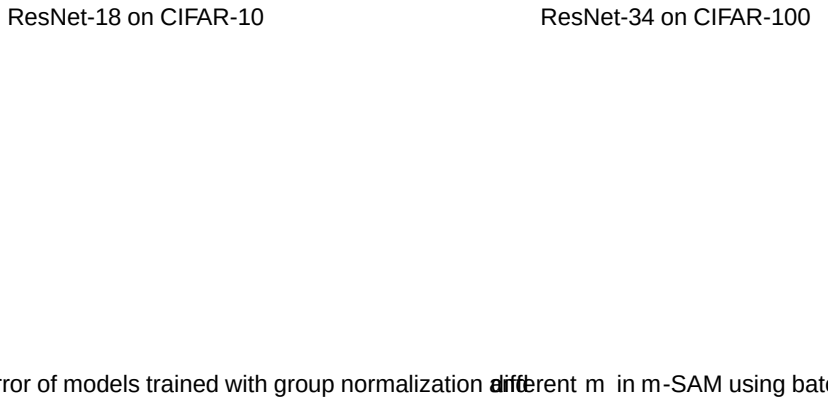


Figure 16: Test error of models trained with group normalization and different m in m-SAM using batch size 256.

E.2. The Effect of the Batch Size on SAM

We show the results of SAM for different batch sizes in Fig. 17 where we use m equal to the batch size. Note that a too high m leads to marginal improvements in generalization (0.2%) and is not able to bridge the gap between large-batch (1024) and small-batch (256 or 128) SGD.

E.3. The Effect of the Model Width on SAM

We show in Fig. 18 test error improvements of SAM over ERM for different model width factors. For comparison, in all other experiments we use model width factor 64. As expected, there is little improvement (or even no improvement as on CIFAR-10) from SAM for small networks where extra regularization is not needed. However, interestingly, the generalization improvement is the largest not for the widest models, but rather for intermediate model widths, such as model width 16.

[4] https://github.com/tml-epfl/understanding-sam

ResNet-18 on CIFAR-10 ResNet-34 on CIFAR-100

Figure 17: Test error of models trained with group normalization and different batch sizes for the same number of epochs (200). Note that for all models, we use ρ in m-SAM equal to the batch size.

ResNet-18 on CIFAR-10 ResNet-34 on CIFAR-100

Figure 18: Test error improvements of SAM over ERM for different model width factors.

E.4. Sharpness for Models with Batch Normalization

The main problem of measuring sharpness for networks with BatchNorm is the discrepancy between training and test-time behaviour. Fig. 19 illustrates this issue: the maximum loss computed over radius ρ is substantially different depending on whether we use training-time vs. test-time BatchNorm. This is an important discrepancy since the training-time BatchNorm is effectively used by SAM while the test-time BatchNorm is used by default for post-hoc sharpness computation. To avoid this discrepancy, we presented the results in the main part only on models trained with GroupNorm which does not have this problem.

ResNet-18 on CIFAR-10

Figure 19: 128-sharpness ($= 0.1$) over training for a network with batch normalization when measured with the training-time and test-time batch normalization. The model is trained with SAM using $\rho = 0.1$.

E.5. Training Loss for ERM vs. SAM Models

Fig. 11 in the main part shows that both training and test errors have a slight increasing trend after the first learning rate decay at 500 epochs. As a sanity check, in Fig. 20, we plot the total objective value (including the ℓ_2 regularization term) which shows a consistent decreasing trend. Thus, we conclude that the increasing training error is not some anomaly connected to a failure of optimizing the training objective.

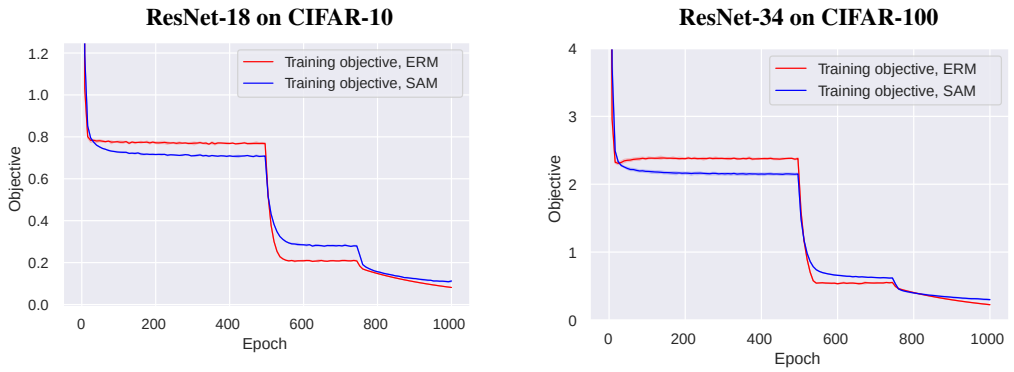


Figure 20: Training objective of ERM vs. SAM over epochs. For both models, we observe a clear decreasing trend.

E.6. SAM with a Decreasing Perturbation Radius ρ

In Fig. 21, we plot the test error over different ρ_t where we decay the ρ_t using the same schedule as for the outer learning rate γ_t. We denote this as *SAM with decreasing* ρ contrary to the standard SAM for which ρ is constant throughout training. We note that in both cases, we use the ℓ_2-normalized updates as in the original SAM. The results suggest that decreasing the perturbation radius ρ_t over epochs is detrimental to generalization. This observation is relevant in the context of the convergence analysis that suggests that SAM converges even if ρ_t is significantly larger than the outer step size γ_t which is the case when we decay γ_t over epochs while keeping ρ_t constant.

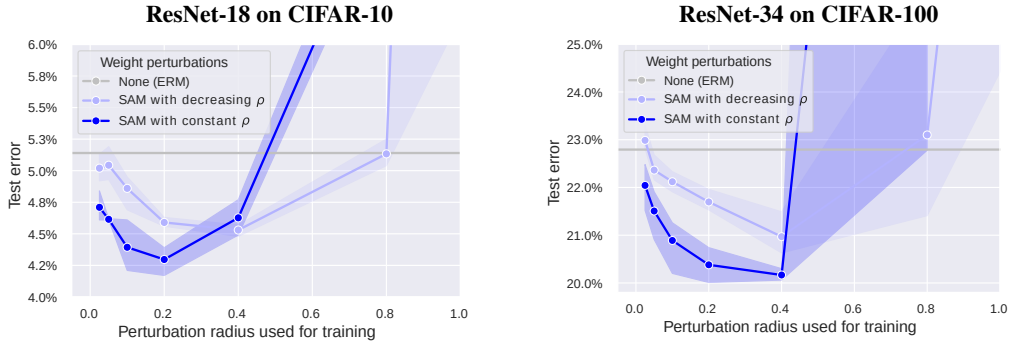


Figure 21: Test error of SAM with a constant perturbation radius ρ (i.e., standard SAM) compared to SAM with decreasing perturbation radii ρ_t. The decrease of ρ_t follows the same piecewise constant schedule as the learning rate γ_t. We note that in both cases, we use the ℓ_2-normalized updates as in the original SAM.

E.7. Experiments with Noisy Labels

In Fig. 22, we show experiments with CIFAR-10 and CIFAR-100 with 60% of noisy labels for SAM with a fixed inner step size ρ that does not include gradient normalization (denoted as *unnormalized SAM*). We did a prior grid search to determine the best fixed ρ for this case which we show in the figure. We can observe that the best test error taken over epochs almost exactly matches that of the standard SAM.

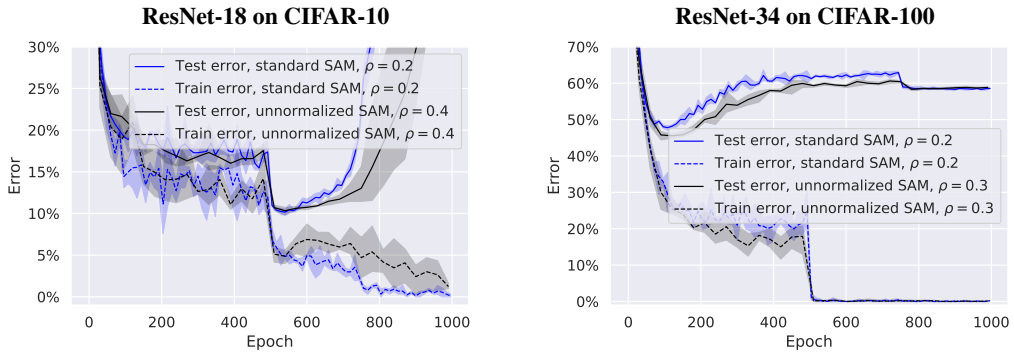


Figure 22: Plots over training for a ResNet-18 trained on CIFAR-10 with 60% label noise for SAM with and without gradient normalization.