

Preconditioned Sharpness-Aware Minimization: Unifying Analysis and a Novel Learning Algorithm

Yilang Zhang*

Dept. of ECE, Univ. of Minnesota
Minneapolis, MN 55455, USA
zhan7453@umn.edu

Bingcong Li*

Dept. of CS, ETH Zürich
8092 Zürich, Switzerland
bingcong.li@inf.ethz.ch

Georgios B. Giannakis

Dept. of ECE, Univ. of Minnesota
Minneapolis, MN 55455, USA
georgios@umn.edu

Abstract—Targeting solutions over ‘flat’ regions of the loss landscape, sharpness-aware minimization (SAM) has emerged as a powerful tool to improve generalizability of deep neural network based learning. While several SAM variants have been developed to this end, a unifying approach that also guides principled algorithm design has been elusive. This contribution leverages preconditioning (pre) to unify SAM variants and provide not only unifying convergence analysis, but also valuable insights. Building upon preSAM, a novel algorithm termed infoSAM is introduced to address the so-called adversarial model degradation issue in SAM by adjusting gradients depending on noise estimates. Extensive numerical tests demonstrate the superiority of infoSAM across various benchmarks.

Index Terms—sharpness-aware minimization, preconditioning, generalizability, convergence analysis, machine learning.

I. INTRODUCTION

Advances in deep neural network (DNN) architectures have led to impressive success across various domains including language, audio, and vision [1]–[3]. Owing to the markedly high dimensionality, DNNs can memorize a large gamut of training data [4]. As a result, small loss during training does not guarantee generalization to unseen data. Catalyzing generalizability of DNNs through fine-grained training schemes remains a long-standing and prohibitively critical challenge.

Popular approaches to cope with generalization include data augmentation and regularization [5], [6]. Albeit effective, data augmentation is often picked in a handcrafted way, and may not universally fit various models and datasets. This prompts combining augmentations, but the optimal mix requires extensive trials. On the other hand, regularization methods such as weight decay and dropout, though straightforward to use, largely omit data properties. For complex models, simply stacking data augmentation and regularization is insufficient [3]. In image classification for example, optimal augmentation and regularization can be class dependent [7]. These limitations unveil the need for fine-grained approaches that jointly account for data and model characteristics.

One such approach resorts to advanced optimization by carefully accounting for the loss landscape, which depends on latent properties of both data distribution, and the DNN architecture. Among possible solutions on the loss curve, those lying on a flatter valley have higher potential for improving

TABLE I: Existing instances of our preSAM framework

Approach	Precond.	Additional as.
ASAM [13]	CP	non-divergent
FisherSAM [14]	CP	lo. bound stoch. grad.
ℓ_∞ or ℓ_1 -SAM [12]	OP	N/A
modified-SSAM [18]	OP	N/A
Lazy SAM [19], [20]	OP	N/A
VaSSO [21]	OP	N/A
InfoSAM (ours)	OP	N/A

generalizability [8]. Supporting evidence comes from theoretical analysis [9], [10] and empirical experimentation [11]. The resultant sharpness-aware minimization (SAM) [12], seeks a flatter region by forcing the surrounding neighborhood to have small loss. Various approaches have also been proposed to further boost the effectiveness of SAM [13]–[17]. Unfortunately, a unifying framework is lacking to encompass existing SAM variants, and inspire the principled design of novel approaches.

Toward this goal, the present work relies on *preconditioning* to unify SAM variants; hence, the term preconditioned (pre) SAM. Depending on where preconditioning is effected, PreSAM categorizes existing SAM variants into objective/constraint preconditioning (OP/CP); see also Table I. Unifying convergence analysis will be offered for both CP and OP. This will not only establish convergence for e.g., ASAM [13] and FisherSAM [14], but will also guide the development of novel algorithms. Building on preSAM, a novel OP approach will be developed to ameliorate the effect of stochastic gradient noise that causes what is termed *adversarial model degradation* (AMD). This new approach, dubbed infoSAM, relies on a preconditioner that adjusts gradient entries depending on noise estimates, thus effectively bypassing the effect of gradient noise and leading to improved generalization. All in all, our contribution is three-fold.

- Rooted on preconditioning, a framework termed preSAM is developed to unify existing SAM variants, and categorize them as OP/CP according to their preconditioners.
- PreSAM offers a unifying convergence analysis for its two subcategories, which fulfills the missing analysis and unjustified experimental preferences of many SAM variants.
- InfoSAM is our novel OP algorithm that handles the AMD issue in SAM. Numerical tests showcase the effectiveness of infoSAM in enhancing generalizability.

Notation. Bold lowercase (capital) letters denote vectors

*Equal contribution. Work was supported in part by NSF grants 2312547, 2212318, and 2126052.

(matrices); $\|\cdot\|$ and $\langle\cdot,\cdot\rangle$ stand for ℓ_2 -norm and inner product; $\mathbb{KL}(\cdot\|\cdot)$ represents the KL divergence; and $\mathbf{e}_i \in \mathbb{R}^d$ is the i th column of the identity matrix $\mathbf{I}_d \in \mathbb{R}^{d \times d}$.

II. SAM RECAP

Let $\mathbf{x} \in \mathbb{R}^d$ denote the parameters of a DNN, and f the non-convex empirical risk (loss) given a dataset $\mathcal{D} := \{\mathbf{a}_i, b_i\}_{i=1}^D$ with feature \mathbf{a}_i and label b_i . To find a solution lying in a flat basin of f , SAM enforces small loss on the neighborhood of \mathbf{x} . This is achieved by the minimax problem

$$\min_{\mathbf{x}} \max_{\|\epsilon\| \leq \rho} f(\mathbf{x} + \epsilon) \quad (1)$$

where $\mathbf{x} + \epsilon$ acts as the most ‘adversarial’ model in the neighborhood sphere of radius ρ . The highly-nonconvex nature of (1) discourages solving the inner maximization exactly. SAM effects this using two approximations

$$\begin{aligned} \epsilon_t &= \arg\max_{\|\epsilon\| \leq \rho} f(\mathbf{x}_t + \epsilon) \stackrel{(a)}{\approx} \arg\max_{\|\epsilon\| \leq \rho} f(\mathbf{x}_t) + \langle \nabla f(\mathbf{x}_t), \epsilon \rangle \\ &\stackrel{(b)}{\approx} \arg\max_{\|\epsilon\| \leq \rho} \langle \mathbf{g}_t(\mathbf{x}_t), \epsilon \rangle \end{aligned} \quad (2)$$

where (a) follows from a first-order Taylor expansion, and (b) replaces the gradient $\nabla f(\mathbf{x}_t)$ with the stochastic gradient $\mathbf{g}_t(\mathbf{x}_t)$. For convenience, we will refer to (2), as *SAM subproblem*. The latter admits the closed-form solution

$$\epsilon_t = \rho \mathbf{g}_t(\mathbf{x}_t) / \|\mathbf{g}_t(\mathbf{x}_t)\|. \quad (3)$$

SAM then updates \mathbf{x}_t using the stochastic gradient $\mathbf{g}_t(\mathbf{x}_t + \epsilon_t)$ at $\mathbf{x}_t + \epsilon_t$. The steps of SAM are listed under Alg. 1.

III. UNIFYING SAM VIA PRECONDITIONING

This section introduces a unifying approach to finding the adversarial model, where popular SAM variants are subsumed as special cases. All proofs are deferred to the Appendix accessible at <https://arxiv.org/pdf/2501.06603>.

A. Preconditioned SAM

PreSAM leverages preconditioning to encompass several SAM variants, each with different preconditioners. In its most general form, **preSAM** finds ϵ_t by solving a preconditioned version of (2):

$$\textbf{PreSAM: } \max_{\epsilon} \langle \mathbf{C}_t \mathbf{g}_t(\mathbf{x}_t), \epsilon \rangle \text{ s.t. } \|\mathbf{D}_t \epsilon\| \leq \rho. \quad (4)$$

Here, $\mathbf{C}_t, \mathbf{D}_t \in \mathbb{R}^{d \times d}$ are preconditioners that alter the geometry of the SAM subproblem. In doing so, the adversarial model can be equipped with designable properties. In particular, \mathbf{C}_t skews the direction of $\mathbf{g}_t(\mathbf{x}_t)$ in the objective, while \mathbf{D}_t reshapes the constraint set. Both \mathbf{C}_t and \mathbf{D}_t can change over iterations, allowing preSAM to adapt to the local geometry for each t . The original SAM subproblem (2) can be recovered by simply fixing $\mathbf{C}_t = \mathbf{D}_t = \mathbf{I}_d$. Supposing for simplicity that \mathbf{D}_t is invertible, preSAM also admits a closed-form solution

$$\epsilon_t = \rho \mathbf{D}_t^{-2} \mathbf{C}_t \mathbf{g}_t(\mathbf{x}_t) / \|\mathbf{D}_t^{-1} \mathbf{C}_t \mathbf{g}_t(\mathbf{x}_t)\|. \quad (5)$$

Before delving into specific choices for \mathbf{C}_t and \mathbf{D}_t in existing algorithms, a natural question is whether the preconditioners conflict with finding a ‘good’ solution of (1).

Algorithm 1 PreSAM

```

1: Initialize:  $\mathbf{x}_0, \rho$ 
2: for  $t = 0, \dots, T - 1$  do
3:   Sample a minibatch  $\mathcal{B}_t$ 
4:   Denote the stochastic gradient on  $\mathcal{B}_t$  as  $\mathbf{g}_t(\cdot)$ 
5:   (preSAM) Find  $\epsilon_t$  via a unified manner (5).
   // SAM:  $\mathbf{C}_t = \mathbf{D}_t = \mathbf{I}_d$ ; InfoSAM:  $\mathbf{C}_t$  and  $\mathbf{D}_t$  via (7)
6:   Calculate stochastic gradient  $\mathbf{g}_t(\mathbf{x}_t + \epsilon_t)$ 
7:   Update model via  $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \mathbf{g}_t(\mathbf{x}_t + \epsilon_t)$ 
8: end for
9: Return:  $\mathbf{x}_T$ 

```

The challenge arises from the fact that (4) is no longer obtained from Taylor’s expansion of $f(\mathbf{x}_t + \epsilon)$. We answer this question under several standard assumptions for nonconvex optimization and SAM [15], [18], [22], [23].

Assumption 1. $f(\mathbf{x})$ is lower bounded, i.e., $f(\mathbf{x}) \geq f^*, \forall \mathbf{x}$.

Assumption 2. $\mathbf{g}(\mathbf{x})$ is L -Lipschitz, i.e., $\|\mathbf{g}(\mathbf{x}) - \mathbf{g}(\mathbf{y})\| \leq L\|\mathbf{x} - \mathbf{y}\|, \forall \mathbf{x}, \mathbf{y}$.

Assumption 3. $\mathbf{g}(\mathbf{x})$ is unbiased with bounded variance, i.e., $\mathbb{E}[\mathbf{g}(\mathbf{x})|\mathbf{x}] = \nabla f(\mathbf{x})$, and $\mathbb{E}[\|\mathbf{g}(\mathbf{x}) - \nabla f(\mathbf{x})\|^2|\mathbf{x}] \leq \sigma^2$.

Under these mild assumptions, the unified convergence is established in the following theorem.

Theorem 1 (Unified convergence). *Suppose As. 1 – 3 hold. Let $\eta_t \equiv \eta = \frac{\eta_0}{\sqrt{T}} \leq \frac{2}{3L}$, and $\rho = \frac{\rho_0}{\sqrt{T}}$. In addition, suppose $\|\mathbf{D}_t^{-1}\| \leq D_0, \forall t$. Then, preSAM in Alg. 1 guarantees that*

$$\begin{aligned} \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \|\nabla f(\mathbf{x}_t)\|^2 &\leq \mathcal{O} \left(\frac{f(\mathbf{x}_0) - f^*}{\eta_0 \sqrt{T}} + \frac{L \rho_0^2 D_0^2}{\eta_0 \sqrt{T}} + \frac{L \eta_0 \sigma^2}{\sqrt{T}} \right), \\ \frac{1}{T} \sum_{t=0}^{T-1} \mathbb{E} \|\nabla f(\mathbf{x}_t + \epsilon_t)\|^2 &\leq \frac{2}{T} \sum_{t=0}^{T-1} \mathbb{E} \|\nabla f(\mathbf{x}_t)\|^2 + \frac{2L^2 \rho_0^2 D_0^2}{T}. \end{aligned}$$

Thm. 1 reveals that \mathbf{D}_t has to be designed carefully to avoid slowing down convergence. In contrast, \mathbf{C}_t is more flexible to choose as it does not explicitly influence the convergence rate, which is yet critical for generalization because it determines how powerful the adversarial model is.

Next, we elaborate on choices of \mathbf{C}_t and \mathbf{D}_t to link preSAM to existing SAM variants. We will also dive deeper into their influences on convergence, which has been overlooked by existing works. Even though it is possible to jointly design \mathbf{C}_t and \mathbf{D}_t , most SAM variants only work with a single preconditioner. Depending on whether $\mathbf{C}_t = \mathbf{I}_d$ or $\mathbf{D}_t = \mathbf{I}_d$, preSAM can be further categorized into constraint preconditioning (CP) and objective preconditioning (OP).

B. Constraint preconditioning (CP)

CP aims to alter the constraint geometry in (4), where it keeps $\mathbf{C}_t = \mathbf{I}_d$, and designs \mathbf{D}_t on demand. Essentially, \mathbf{D}_t converts the ℓ_2 -norm ball $\{\epsilon : \|\epsilon\| \leq \rho\}$ into an ellipsoid. Intuitively, this is helpful when knowing a priori that certain dimensions contribute more to the adversarial model. A caveat for designing \mathbf{D}_t is that its inversion should be affordable;

cf. (5). As a consequence, most existing CP approaches rely on diagonal \mathbf{D}_t , as discussed next.

Scale-invariant adversarial model via CP. It was pointed out in [24] that proper rescaling of NN weights does not change the loss function. This means there exist multiple adversarial models with the same loss, rendering the optimal one indistinguishable from the rest. ASAM [13] copes with this issue by rescaling the constraint set, which serves as a specific instance of CP. In its simplest form, ASAM adopts $\mathbf{D}_t = \text{diag}(|\mathbf{x}_t|^{-1})$, where $|\cdot|$ and \cdot^{-1} are entry-wise operators. If $[\mathbf{x}_t]_i$ is small, ASAM tends to increase the perturbation $[\epsilon_t]_i$.

Fisher adversarial model via CP. While SAM seeks ϵ_t within a Euclidean ball, this can be extended to more sophisticated spaces. For example, FisherSAM [14] considers a ball induced by KL divergence, namely $\mathbb{E}_{\mathcal{D}}[\text{KL}(p(b_i|\mathbf{a}_i, \mathbf{x}_t + \epsilon)|p(b_i|\mathbf{a}_i, \mathbf{x}_t))] \leq \rho$. Modified with several approximations for computational efficiency, FisherSAM ends up with a specific form of CP, where $\mathbf{D}_t = \text{diag}(|\mathbf{g}_t|)$.

CP can challenge convergence. As stated in Thm. 1, the convergence rate of CP critically depends on D_0 . Unfortunately, both ASAM and FisherSAM are on the edge of divergence. For ASAM, it holds that $D_0 = \max_t \|\mathbf{x}_t\|_\infty$, which could be unbounded unless assuming non-divergence. For FisherSAM, $D_0 = \max_t \|\mathbf{g}_t^{-1}\|_\infty$ can also be unbounded and slowdowns convergence as $[\mathbf{g}_t]_i$ can be arbitrarily small.

Moreover, for CP to attain the same convergence rate as SAM, it requires $\rho = \rho_0/\sqrt{T} \propto 1/D_0$. Upon ASAM convergence, it typically holds that $D_0 < 1$. This explains the empirical observation that a larger ρ helps ASAM to perform best [13]. The same was also corroborated in our experiments, where adopting the same ρ as SAM degrades ASAM's performance. Somehow ironically, an enlarged ρ makes the Taylor approximation (a) in (2) inaccurate, which can weaken the adversarial model. This leads to another issue for CP, that is, to determine the best ρ through extra effort.

C. Objective preconditioning (OP)

For the objective in (4), OP fixes $\mathbf{D}_t = \mathbf{I}_d$, and adapts merely \mathbf{C}_t . As asserted by Thm. 1, OP is more flexible since convergence rate is not explicitly dependent on its preconditioner. In addition, OP is less stringent than CP because: i) \mathbf{C}_t need not be invertible; and ii) scaling \mathbf{C}_t has no impact on ϵ_t . The latter can be verified by replacing \mathbf{C}_t with $\alpha\mathbf{C}_t, \forall \alpha > 0$, which does not alter the solution (5). By redirecting $\mathbf{g}_t(\mathbf{x}_t)$, OP seeks an improved adversarial model. Depending on the specific \mathbf{C}_t , OP can be used for various purposes.

Adversarial models in non-ellipsoidal neighborhood via OP. While CP's constraint set is an ellipsoid, OP gives rise to a non-ellipsoidal neighborhood when \mathbf{C}_t is properly designed. Table II exemplifies three choices of \mathbf{C}_t for which the resultant ϵ_t amounts to solving (4) under ℓ_1 , ℓ_∞ , or n -support norm ball [25] constraints. The former two are found in [12], while the last is our extension, where a n -support norm ball can be viewed as a combination of ℓ_1 and ℓ_2 norm constraint.

Sparse perturbation via OP. The second and third method in Table II both result in a sparse ϵ_t . This helps reduce the

TABLE II: OP and its equivalent constraint.

OP	Equiv. constr. for (4)
$\mathbf{C}_t = \text{diag}(\mathbf{g}_t ^{-1})$	$\ \epsilon\ _\infty \leq \rho$
$\mathbf{C}_t = \text{diag}(\mathbf{e}_i)$ with $i = \text{argmax}_i \mathbf{g}_t(\mathbf{x}_t) _i$	$\ \epsilon\ _1 \leq \rho$
$\mathbf{C}_t = \text{diag}(\sum_{i \in \mathcal{I}} \mathbf{e}_i)$ with $\mathcal{I} = \text{argtop}_n(\mathbf{g}_t(\mathbf{x}_t))$	$\ \epsilon\ _{n\text{-supp}} \leq \rho$

backpropagation complexity of $\mathbf{g}_t(\mathbf{x}_t + \epsilon_t)$. More involved approaches along this line include SSAM [18], which not only assumes bounded gradient, but also suffers from rate slower than SAM. These issues can be addressed by changing the algorithmic order; that is, first sparsify the gradient via OP by setting the corresponding entries of \mathbf{C}_t to 0 as [18, Alg. 2], and then use infoSAM (7) to obtain ϵ_t . We term this method modified SSAM, and our experiments show that it matches the performance of vanilla SSAM.

Lazy adversary model via OP. Lazy SAM [19], [20] switches between SAM's adversarial objective (1) and empirical risk minimization (ERM) to lower the computational cost. With ERM-induced update $\mathbf{x}_{t+1} = \mathbf{x}_t - \eta \mathbf{g}_t(\mathbf{x}_t)$, this avoids SAM's second gradient computation $\mathbf{g}(\mathbf{x}_t + \epsilon_t)$. Given that $\mathbf{C}_t = \mathbf{0}$ in (4) leads to $\epsilon_t = \mathbf{0}$, preSAM is able to recover lazy SAM by setting $\mathbf{C}_t = \mathbf{0}$ whenever switching to ERM.

Chain of preconditioners. It is also possible to equip an adversarial model with multiple desired properties through a cascade of preconditioners. For example, if $\{\mathbf{C}_{t,i}\}_{i=1}^I$ are valid OP choices, $\mathbf{C}_t = \prod_{i=1}^I \mathbf{C}_{t,i}$ is also a valid OP preconditioner.

IV. INFOSAM

This section develops a new instance of preSAM that copes with the adversarial model degradation challenge of SAM.

A. Adversarial model degradation (AMD)

The stochastic noise in $\mathbf{g}_t(\mathbf{x}_t)$ can markedly harm the adversarial model $\mathbf{x}_t + \epsilon_t$ obtained via (3) [21]. We term this *adversarial model degradation*, and further elaborate on its harmfulness, which motivates our novel algorithm, infoSAM.

Consider SAM in the ideally noise-free case, i.e., $\mathbf{g}_t(\mathbf{x}_t) = \nabla f(\mathbf{x}_t)$. Then, the perturbation of the i th dimension satisfies $[\epsilon_t]_i \propto [\nabla f(\mathbf{x}_t)]_i$; cf. (3). This matches the intuition for finding the most adversarial model, since it holds that

$$f(\mathbf{x}_t + \lambda \mathbf{e}_i) - f(\mathbf{x}_t) \leq \lambda \langle \nabla f(\mathbf{x}_t), \mathbf{e}_i \rangle + \frac{L\lambda^2}{2} \quad (6)$$

$$\stackrel{(a)}{=} \alpha [\nabla f(\mathbf{x}_t)]_i^2 + \frac{L\alpha^2}{2} [\nabla f(\mathbf{x}_t)]_i^2 \propto [\nabla f(\mathbf{x}_t)]_i^2$$

where (a) is by taking $\lambda = \alpha [\nabla f(\mathbf{x}_t)]_i$ for some $\alpha > 0$. When $[\nabla f(\mathbf{x}_t)]_i$ is large, the adversarial model has the potential to induce a higher loss by moving more toward this dimension.

In practice, SAM relies on $\mathbf{g}_t(\mathbf{x}_t)$ rather than $\nabla f(\mathbf{x}_t)$, with which (6) can hardly hold. When the stochastic noise is dominant, $[\nabla \mathbf{g}_t(\mathbf{x}_t)]_i$ can even correspond to a descent direction. When training a ResNet-18 on CIFAR10, we observed that the signal-to-noise ratio (SNR) is around $\mathcal{O}(10^{-2})$ throughout 200 training epochs. This suggests that the gradient noise is indeed a severe issue for SAM. Additional examples on how AMD affects the convergence behavior of SAM in an asymmetric valley can be found in App. C.

TABLE III: Comparison of infoSAM against other baselines.

	Architecture	SGD	SAM	ASAM	InfoSAM
CIFAR10	ResNet	96.25 \pm 0.06	96.58 \pm 0.10	96.33 \pm 0.09	96.71\pm0.09
	DenseNet	96.65 \pm 0.13	96.94 \pm 0.11	96.73 \pm 0.18	97.09\pm0.07
	WideResNet	97.08 \pm 0.16	97.32 \pm 0.11	97.15 \pm 0.05	97.56\pm0.12
	PyramidNet	97.39 \pm 0.09	97.85 \pm 0.14	97.56 \pm 0.11	98.04\pm0.06
CIFAR100	ResNet	77.90 \pm 0.07	80.96 \pm 0.12	79.91 \pm 0.04	81.31\pm0.15
	DenseNet	81.62 \pm 0.19	83.94 \pm 0.08	82.75 \pm 0.10	84.09\pm0.12
	WideResNet	81.71 \pm 0.13	84.88 \pm 0.10	83.54 \pm 0.14	85.01\pm0.07
	PyramidNet	83.50 \pm 0.12	85.60 \pm 0.11	83.72 \pm 0.09	85.83\pm0.11

B. A novel OP approach to handle AMD

Unfortunately, no preSAM approach is available to deal with the AMD challenge caused by gradient noise. This section develops such an OP-based algorithm that we term infoSAM.

Our conception of infoSAM is straightforward – when seeking the adversarial model, we should be more cautious on dimensions with smaller SNR since they are less informative. Quantitatively, with $[\sigma_t]_i^2$ denoting the variance of $[\mathbf{g}_t(\mathbf{x}_t)]_i$, infoSAM’s perturbation is $[\epsilon_t]_i \propto [\mathbf{g}_t(\mathbf{x}_t)]_i / [\sigma_t]_i^2$. App. D details how infoSAM works using a numerical case study.

While alleviating AMD using $[\epsilon_t]_i \propto [\mathbf{g}_t(\mathbf{x}_t)]_i / [\sigma_t]_i^2$ is intriguing, the variance vector σ_t^2 is generally intractable. Inspired by [26], we estimate σ_t^2 by the squared difference between $\mathbf{g}_t(\mathbf{x}_t)$ ’s exponentially moving average (EMA) and $\mathbf{g}_t(\mathbf{x}_t)$ itself. The EMA \mathbf{m}_t is accumulated as

$$\mathbf{m}_t = \alpha \mathbf{m}_{t-1} + (1 - \alpha) \mathbf{g}_t(\mathbf{x}_t) \quad (7a)$$

where $0 < \alpha < 1$ is a hyperparameter. Vector \mathbf{m}_t serves as an estimate of $\nabla f(\mathbf{x}_t)$, which is then leveraged to estimate

$$\hat{\sigma}_t^2 = (\mathbf{m}_t - \mathbf{g}_t(\mathbf{x}_t))^2. \quad (7b)$$

With $\hat{\Sigma}_t := \text{diag}(\hat{\sigma}_t^2)$, infoSAM obtains its ϵ_t via

$$\epsilon_t = \underset{\|\epsilon\| \leq \rho}{\text{argmax}} \langle \hat{\Sigma}_t^{-1} \mathbf{g}_t(\mathbf{x}_t), \epsilon \rangle = \rho \frac{\hat{\Sigma}_t^{-1} \mathbf{g}_t(\mathbf{x}_t)}{\|\hat{\Sigma}_t^{-1} \mathbf{g}_t(\mathbf{x}_t)\|}. \quad (7c)$$

The step-by-step implementation of infoSAM is summarized in Alg. 1. It is also worth noting that infoSAM can be used jointly with CP methods such as ASAM and FisherSAM, which has been added to our future research agenda.

V. NUMERICAL TESTS

Here we test infoSAM’s numerical efficiency. Implementation details are deferred to App. E.

A. CIFAR10 and CIFAR100

The evaluation starts with image classification on benchmarks CIFAR10 and CIFAR100 [27]. The backbone architectures are convolutional neural networks including ResNet-18 [28], DenseNet-121 [29], WideResNet-28-10 [30], and PyramidNet-110 [31]. Besides infoSAM, we also test stochastic gradient descent (SGD), SAM, and ASAM as baselines.

The test accuracies are gathered in Tab. III. The proposed infoSAM achieves the highest accuracy in all model setups, validating that AMD can be alleviated through proper preconditioning. The results also suggest that CP can be delicate when ρ is not chosen properly. As discussed after Thm. 1,

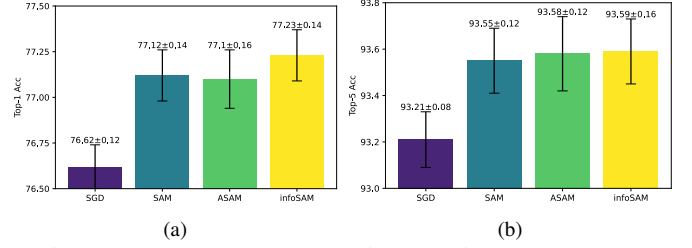


Fig. 1: (a) Top-1 and (b) top-5 accuracies on ImageNet.

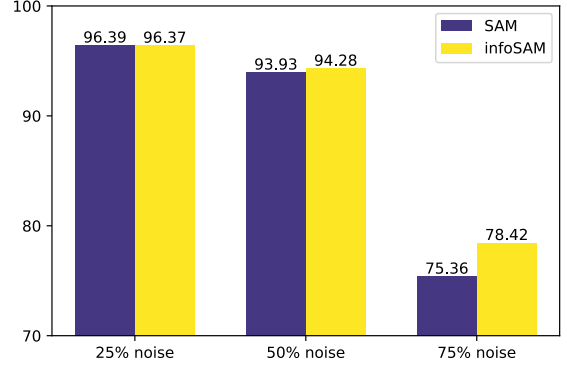


Fig. 2: Performance under different levels of label noise.

CP approaches such as ASAM rely on a large ρ to achieve comparable performance over SAM. This matches the results in Tab. III, where ASAM underperforms SAM when adopting the same ρ , and only slightly improves over SGD. This demonstrates that CP has to be used cautiously, and further justifies our preference of OP for tackling the AMD issue.

B. ImageNet

Next, we investigate the performance of infoSAM on large-scale experiments by training a ResNet-50 [28] on ImageNet [32]. Fig. 1 plots the top-1 and top-5 accuracy of tested algorithms. It can be observed that infoSAM has the best top-1 as well as top-5 accuracies. Again, the CP-based ASAM does not catch up with SAM when using the same ρ .

C. Label noise

SAM is known to exhibit robustness against large label noise in the training set [12]. Since the loss landscape can be heavily perturbed, it is expected that infoSAM outperforms SAM. In our experiments, we consider the classical noisy-label setting, where a fraction of the training labels are randomly flipped, whereas the test set remains clean. A ResNet-18 [28] is trained on CIFAR10 with label noise levels {25%, 50%, 75%}. It can be seen from Fig. 2 that infoSAM markedly improves SAM in high-level label noise.

VI. CONCLUSIONS

We developed a preconditioning-based SAM framework that provides: i) unifying convergence analysis of SAM variants; ii) valuable insights of experimental results; and, iii) guidelines to develop novel SAM algorithms. Within this framework, infoSAM can tackle the AMD challenge of SAM, and thus improves generalization across various benchmarks.

REFERENCES

- [1] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, “Bert: Pre-training of deep bidirectional transformers for language understanding,” *arXiv:1810.04805*, 2018.
- [2] Y. Gong, Y.-A. Chung, and J. Glass, “Ast: Audio spectrogram transformer,” *arXiv preprint arXiv:2104.01778*, 2021.
- [3] X. Chen, C.-J. Hsieh, and B. Gong, “When vision transformers outperform resnets without pre-training or strong data augmentations,” in *Proc. Int. Conf. Learning Representation*, 2022.
- [4] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, “Understanding deep learning (still) requires rethinking generalization,” *Communications of the ACM*, vol. 64, no. 3, pp. 107–115, 2021.
- [5] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: a simple way to prevent neural networks from overfitting,” *J. Mach. Learn. Res.*, vol. 15, pp. 1929–1958, 2014.
- [6] E. D. Cubuk, B. Zoph, D. Mane, V. Vasudevan, and Q. V. Le, “Autoaugment: Learning augmentation strategies from data,” in *Proc. Conf. Computer Vision and Pattern Recognition*, 2019, pp. 113–123.
- [7] R. Balestriero, L. Bottou, and Y. LeCun, “The effects of regularization and data augmentation are class dependent,” *arXiv preprint arXiv:2204.03632*, 2022.
- [8] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang, “On large-batch training for deep learning: Generalization gap and sharp minima,” in *Proc. Int. Conf. Learning Representation*, 2016.
- [9] M. Andriushchenko and N. Flammarion, “Towards understanding sharpness-aware minimization,” in *Proc. Int. Conf. Machine Learning*, 2022, pp. 639–668.
- [10] K. Wen, T. Ma, and Z. hiyuan Li, “How does sharpness-aware minimization minimizes sharpness,” in *Proc. Int. Conf. Learning Representation*, 2023.
- [11] Y. Jiang, B. Neyshabur, D. Krishnan, H. Mobahi, and S. Bengio, “Fantastic generalization measures and where to find them,” *arXiv:1912.02178*, 2019.
- [12] P. Foret, A. Kleiner, H. Mobahi, and B. Neyshabur, “Sharpness-aware minimization for efficiently improving generalization,” in *Proc. Int. Conf. Learning Representation*, 2021.
- [13] J. Kwon, J. Kim, H. Park, and I. K. Choi, “ASAM: Adaptive sharpness-aware minimization for scale-invariant learning of deep neural networks,” in *Proc. Int. Conf. Machine Learning*, vol. 139, 2021, pp. 5905–5914.
- [14] M. Kim, D. Li, S. X. Hu, and T. M. Hospedales, “Fisher SAM: Information geometry and sharpness aware minimisation,” in *Proc. Int. Conf. Machine Learning*, 2022, pp. 11 148–11 161.
- [15] J. Zhuang, B. Gong, L. Yuan, Y. Cui, H. Adam, N. Dvornek, S. Tatikonda, J. Duncan, and T. Liu, “Surrogate gap minimization improves sharpness-aware training,” in *Proc. Int. Conf. Learning Representation*, 2022.
- [16] Y. Zhao, H. Zhang, and X. Hu, “Penalizing gradient norm for efficiently improving generalization in deep learning,” in *Proc. Int. Conf. Machine Learning*, 2022, pp. 26 982–26 992.
- [17] B. Li, L. Zhang, and N. He, “Implicit regularization of sharpness-aware minimization for scale-invariant problems,” in *Proc. Adv. Neural Info. Processing Systems*, 2024.
- [18] P. Mi, L. Shen, T. Ren, Y. Zhou, X. Sun, R. Ji, and D. Tao, “Make sharpness-aware minimization stronger: A sparsified perturbation approach,” in *Proc. Adv. Neural Info. Processing Systems*, 2022.
- [19] W. Jiang, H. Yang, Y. Zhang, and J. Kwok, “An adaptive policy to employ sharpness-aware minimization,” *arXiv preprint arXiv:2304.14647*, 2023.
- [20] Y. Zhao, H. Zhang, and X. Hu, “SS-SAM: Stochastic scheduled sharpness-aware minimization for efficiently training deep neural networks,” *arXiv:2203.09962*, 2022.
- [21] B. Li and G. Giannakis, “Enhancing sharpness-aware optimization through variance suppression,” in *Proc. Advances in Neural Info. Process. Syst.*, vol. 36, 2023, pp. 70 861–70 879.
- [22] S. Ghadimi and G. Lan, “Stochastic first-and zeroth-order methods for nonconvex stochastic programming,” *SIAM Journal on Optimization*, vol. 23, no. 4, pp. 2341–2368, 2013.
- [23] L. Bottou, F. E. Curtis, and J. Nocedal, “Optimization methods for large-scale machine learning,” *SIAM Review*, vol. 60, no. 2, pp. 223–311, 2018.
- [24] L. Dinh, R. Pascanu, S. Bengio, and Y. Bengio, “Sharp minima can generalize for deep nets,” in *Proc. Int. Conf. Machine Learning*, 2017, pp. 1019–1028.
- [25] A. Argyriou, R. Foygel, and N. Srebro, “Sparse prediction with the k -support norm,” in *Proc. Advances in Neural Info. Process. Syst.*, 2012, pp. 1457–1465.
- [26] J. Zhuang, T. Tang, Y. Ding, S. C. Tatikonda, N. Dvornek, X. Papademetris, and J. Duncan, “AdaBelief optimizer: Adapting stepsizes by the belief in observed gradients,” in *Proc. Adv. Neural Info. Processing Systems*, vol. 33, 2020, pp. 18 795–18 806.
- [27] A. Krizhevsky, G. Hinton *et al.*, “Learning multiple layers of features from tiny images,” 2009.
- [28] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” in *Proc. Conf. Computer Vision and Pattern Recognition*, June 2016.
- [29] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, “Densely connected convolutional networks,” in *Proc. Conf. Computer Vision and Pattern Recognition*, July 2017.
- [30] S. Zagoruyko, “Wide residual networks,” *arXiv preprint arXiv:1605.07146*, 2016.
- [31] D. Han, J. Kim, and J. Kim, “Deep pyramidal residual networks,” in *Proc. Conf. Computer Vision and Pattern Recognition*, July 2017.
- [32] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, “ImageNet: A large-scale hierarchical image database,” in *Proc. Conf. Computer Vision and Pattern Recognition*, 2009, pp. 248–255.
- [33] H. He, G. Huang, and Y. Yuan, “Asymmetric valleys: Beyond sharp and flat local minima,” in *Proc. Adv. Neural Info. Processing Systems*, vol. 32, 2019, pp. 2549–2560.
- [34] T. Devries and G. W. Taylor, “Improved regularization of convolutional neural networks with cutout,” vol. abs/1708.04552, 2017.