AN ADAPTIVE POLICY TO EMPLOY SHARPNESS-AWARE MINIMIZATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Sharpness-aware minimization (SAM), which searches for flat minima by min-max optimization, has been shown to be useful in improving model generalization. However, since each SAM update requires computing two gradients, its computational cost and training time are both doubled compared to standard empirical risk minimization (ERM). Recent state-of-the-arts reduce the fraction of SAM updates and thus accelerate SAM by switching between SAM and ERM updates randomly or periodically. In this paper, we design an adaptive policy to employ SAM based on the loss landscape geometry. Two efficient algorithms, AE-SAM and AE-LookSAM, are proposed. We theoretically show that AE-SAM has the same convergence rate as SAM. Experimental results on various datasets and architectures demonstrate the efficiency and effectiveness of the proposed method.

1 INTRODUCTION

Despite great success in many applications (He et al., 2016; Zagoruyko & Komodakis, 2016; Han et al., 2017), deep networks are often over-parameterized and capable of memorizing all training data. The training loss landscape is complex and nonconvex with many local minima of different generalization abilities. Many studies have investigated the relationship between geometry of the loss surface and generalization performance (Hochreiter & Schmidhuber, 1994; McAllester, 1999; Keskar et al., 2017; Neyshabur et al., 2017; Jiang et al., 2020), and found that flatter minima generalize better than sharper minima (Dziugaite & Roy, 2017; Petzka et al., 2021; Chaudhari et al., 2017; Keskar et al., 2017; Jiang et al., 2020).

Sharpness-aware minimization (SAM) (Foret et al., 2021) is the current state-of-the-art to seek flat minima by solving a min-max optimization problem. In the SAM algorithm, each update consists of *two* forward-backward computations: one for computing the perturbation and the other for computing the actual update direction. Since these two computations are not parallelizable, SAM doubles the computational overhead as well as the training time compared to empirical risk minimization (ERM).

Several algorithms (Du et al., 2022a; Zhao et al., 2022b; Liu et al., 2022) have been proposed to improve SAM's efficiency. ESAM (Du et al., 2022a) uses fewer samples to compute the gradients and updates fewer parameters, but each update still requires two gradient computations. Thus, ESAM does not alleviate the bottleneck of training speed. Instead of using the SAM update at every iteration, recent state-of-the-arts (Zhao et al., 2022b; Liu et al., 2022) proposed to use SAM randomly or periodically. Specifically, SS-SAM (Zhao et al., 2022b) selects SAM or ERM according to a Bernoulli trial, while LookSAM (Liu et al., 2022) employs SAM at every k step. Though efficiency, random or periodic policy of using SAM is suboptimal as it is not geometry-aware. Intuitively, the SAM update is more necessary for sharp regions than for flat regions.

In this paper, we propose an adaptive policy to employ SAM based on the geometry of the loss landscape. The SAM update is used when the model is in sharp regions, while the ERM update is used in flat regions for reducing the fraction of SAM updates. To measure sharpness, we use the squared stochastic gradient norm and model it as normal distribution. We estimate the distribution parameters by exponential moving average. Experimental results on standard benchmarks demonstrate the superiority of the proposed policy.

Our contributions are summarized as follows: (i) We propose an adaptive policy to employ SAM based on the loss landscape geometry. (ii) We propose an efficient algorithm, called AE-SAM, to

reduce the fraction of SAM updates. We also theoretically study its convergence rates with both batch and stochastic gradient descent. (iii) The proposed policy is general and can be combined with any SAM variant. In this paper, we integrate it with LookSAM and propose AE-LookSAM. (iv) Experimental results on various datasets and network architectures, with and without label noise, verify the superiority of AE-SAM and AE-LookSAM over existing baselines.

Notations. Vectors (e.g., $\mathbf{x}$) and matrices (e.g., $\mathbf{X}$) are denoted by lowercase and uppercase boldface letters, respectively. For a vector $\mathbf{x}$, its ℓ_2-norm is $\|\mathbf{x}\|$. $\mathcal{N}(\mu; \sigma^2)$ is the univariate normal distribution with mean μ and variance σ^2. $\mathrm{diag}(\mathbf{x})$ constructs a diagonal matrix with $\mathbf{x}$ on the diagonal. Let $\mathbb{I}_A(x)$ be an indicator function, i.e., $\mathbb{I}_A(x) = 1$ if $x \in A$ otherwise 0.

2 Related Work

We are given a training set $\mathcal{D}$ with i.i.d. samples $\{(\mathbf{x}_i, y_i) : i = 1, \ldots, n\}$ drawn from a data distribution, Let $f(\mathbf{x}; \mathbf{w})$ be a model parameterized by $\mathbf{w}$. Its empirical risk on $\mathcal{D}$ is $\mathcal{L}(\mathcal{D}; \mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(\mathbf{x}_i; \mathbf{w}), y_i)$, where $\ell(\cdot, \cdot)$ is a loss (e.g., cross-entropy loss for classification). Model training aims to learn a model from the training data that generalizes well on test data.

Generalization and Flat Minima. The connection between model generalization and loss landscape geometry has been theoretically and empirically studied in (Keskar et al., 2017; Dziugaite & Roy, 2017; Jiang et al., 2020). Recently, Jiang et al. (2020) conducted large-scale experiments and found that sharpness-based measures (flatness) are related to generalization of minimizers. Although flatness can be characterized by Hessian eigenvalues (Keskar et al., 2017; Dinh et al., 2017), dealing with Hessian explicitly is computationally prohibitive. To address this issue, practical algorithms propose to seek flat minima by injecting noise in the optimizers (Zhu et al., 2019; Zhou et al., 2019; Orvieto et al., 2022; Bisla et al., 2022), introducing regularization (Chaudhari et al., 2017; Zhao et al., 2022a; Du et al., 2022b), averaging model weights during training (Izmailov et al., 2018; He et al., 2019; Cha et al., 2021), or sharpness-aware minimization (SAM) (Foret et al., 2021; Kwon et al., 2021; Zhuang et al., 2022; Kim et al., 2022).

SAM. The state-of-the-art SAM (Foret et al., 2021) and its variants (Kwon et al., 2021; Zhuang et al., 2022; Kim et al., 2022) search for flat minima by solving the min-max optimization problem:

$$\min_{\mathbf{w}} \max_{\|\boldsymbol{\epsilon}\| \le \rho} \mathcal{L}(\mathcal{D}; \mathbf{w} + \boldsymbol{\epsilon}), \tag{1}$$

where $\rho > 0$ is the radius of perturbation. The above can be rewritten as $\min_{\mathbf{w}} \mathcal{L}(\mathcal{D}; \mathbf{w}) + \mathcal{R}(\mathcal{D}; \mathbf{w})$, where $\mathcal{R}(\mathcal{D}; \mathbf{w}) \equiv \max_{\|\boldsymbol{\epsilon}\| \le \rho} \mathcal{L}(\mathcal{D}; \mathbf{w} + \boldsymbol{\epsilon}) - \mathcal{L}(\mathcal{D}; \mathbf{w})$ is a regularizer that penalizes sharp minimizers (Foret et al., 2021). As solving the inner maximization in Eq.(1) exactly is computationally infeasible for nonconvex optimization, SAM approximately solves it by the first-order Taylor approximation and its update rule is

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \nabla \mathcal{L}(\mathcal{B}_t; \mathbf{w}_t + \rho_t \nabla \mathcal{L}(\mathcal{B}_t; \mathbf{w}_t)), \tag{2}$$

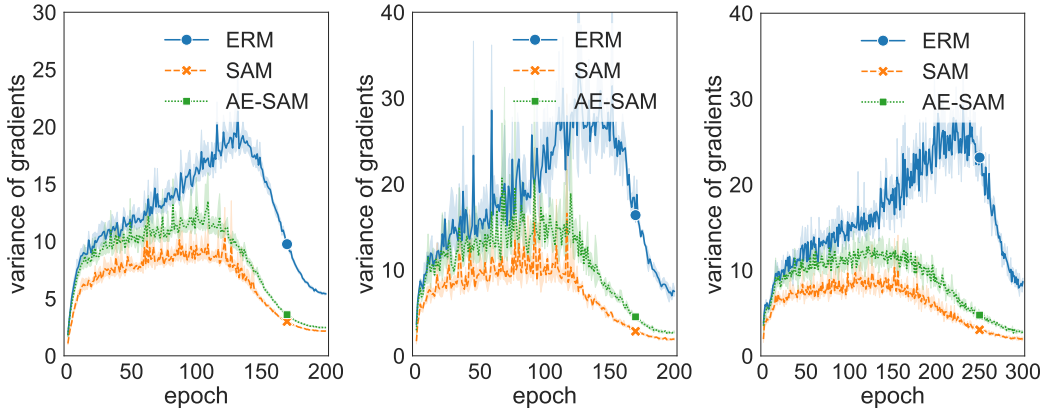
where $\mathcal{B}_t$ is a mini-batch of data, η is the step size, and $\rho_t = \frac{\rho}{\|\nabla \mathcal{L}(\mathcal{B}_t; \mathbf{w}_t)\|}$. Although SAM has shown to be effective in improving the generalization of deep networks, a major drawback is that each update in Eq.(2) requires *two* forward-backward calculations. Specifically, SAM first calculates the gradient of $\mathcal{L}(\mathcal{B}_t; \mathbf{w})$ at $\mathbf{w}_t$ to obtain the perturbation, then calculates the gradient of $\mathcal{L}(\mathcal{B}_t; \mathbf{w})$ at $\mathbf{w}_t + \rho_t \nabla \mathcal{L}(\mathcal{B}_t; \mathbf{w}_t)$ to obtain the update direction for $\mathbf{w}_t$. As a result, SAM doubles the computational overhead compared to ERM.

Efficient Variants of SAM. Several algorithms have been proposed to accelerate the SAM algorithm. ESAM (Du et al., 2022a) uses fewer samples to compute the gradients and only updates part of parameters in the second step, but still requires to compute most of the gradients. Another direction is to reduce the number of SAM updates used in the training process. SS-SAM (Zhao et al., 2022b) randomly selects SAM or ERM update according to a Bernoulli trial, while LookSAM (Liu et al., 2022) employ SAM at every k iteration. Unlike the random or periodic policy of employing SAM, the proposed policy is adaptive to the loss landscape.

3 Method

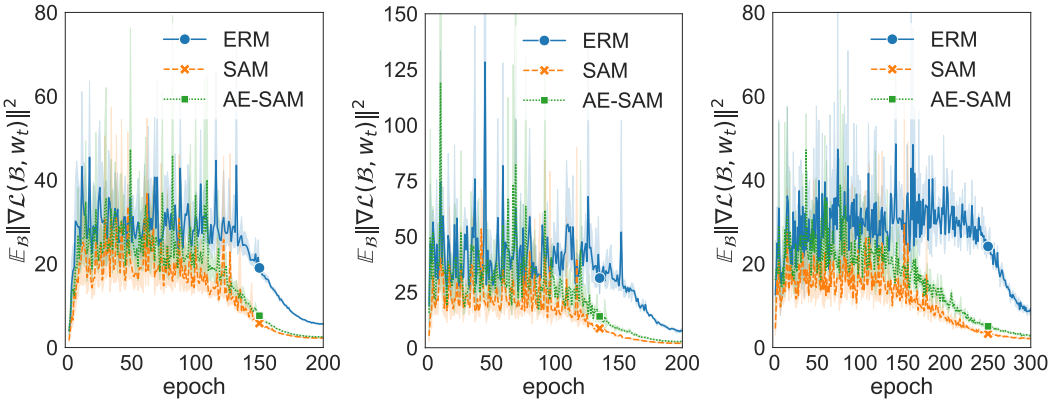
In this section, we propose an adaptive policy to employ the SAM update. The general idea is to use ERM when $\mathbf{w}_t$ is already in a flat region, and use SAM only when the loss landscape is locally sharp.

We start by introducing a sharpness measure (Section 3.1), then propose an adaptive policy based on the sharpness measure (Section 3.2). Next, we propose two algorithms (AE-SAM and AE-LookSAM) and study the convergence of AE-SAM.



(a) *ResNet-18*. (b) *WRN-28-10*. (c) *PyramidNet-110*.

Figure 1: Variance of gradients on *CIFAR-100*. Best viewed in color.



(a) *ResNet-18*. (b) *WRN-28-10*. (c) *PyramidNet-110*.

Figure 2: Squared stochastic gradient norms $\mathbb{E}_{\mathcal{B}}\|\nabla\mathcal{L}(\mathcal{B};\mathbf{w}_t)\|^2$ on *CIFAR-100*. Best viewed in color.

3.1 Sharpness Measure

Though sharpness can be characterized by the eigenvalues of Hessian (Keskar et al., 2017; Dinh et al., 2017), it is expensive to compute. A widely-used approximation is based on the gradient magnitude $\mathrm{diag}([\nabla\mathcal{L}(\mathcal{B};\mathbf{w}_t)]^2)$ (Bottou et al., 2018; Khan et al., 2018), where $[\mathbf{v}]^2$ denotes the elementwise square of $\mathbf{v}$. As $\|\nabla\mathcal{L}(\mathcal{B};\mathbf{w}_t)\|^2$ equals the trace of $\mathrm{diag}([\nabla\mathcal{L}(\mathcal{B};\mathbf{w}_t)]^2)$, it is reasonable to choose $\|\nabla\mathcal{L}(\mathcal{B};\mathbf{w}_t)\|^2$ as a sharpness measure.

$\|\nabla\mathcal{L}(\mathcal{B};\mathbf{w}_t)\|^2$ is also related to another sharpness measure, namely, the variance of gradients, which is defined as

$$\mathsf{Var}(\nabla\mathcal{L}(\mathcal{B};\mathbf{w})) \equiv \mathbb{E}_{\mathcal{B}}\|\nabla\mathcal{L}(\mathcal{B};\mathbf{w}) - \nabla\mathcal{L}(\mathcal{D};\mathbf{w})\|^2 = \mathbb{E}_{\mathcal{B}}\|\nabla\mathcal{L}(\mathcal{B};\mathbf{w})\|^2 - \|\nabla\mathcal{L}(\mathcal{D};\mathbf{w})\|^2. \quad (3)$$

With appropriate smoothness assumptions on the loss $\mathcal{L}$, both SAM and ERM can be shown theoretically to converge to critical points of $\mathcal{L}(\mathcal{D};\mathbf{w})$ (i.e., $\nabla\mathcal{L}(\mathcal{D};\mathbf{w}) = 0$) (Reddi et al., 2016; Andriushchenko & Flammarion, 2022). Thus, it follows from Eq.(3) that the squared stochastic gradient norm $\mathbb{E}_{\mathcal{B}}\|\nabla\mathcal{L}(\mathcal{B};\mathbf{w})\|^2$ indicates the variance of gradients. Jiang et al. (2020) conducted extensive experiments and empirically shows that the variance of gradients is positively correlated with the generalization gap. In other words, the smaller the $\mathsf{Var}(\nabla\mathcal{L}(\mathcal{B};\mathbf{w}))$, the better generalization is the model with parameter $\mathbf{w}$. This finding also explains why SAM generalizes better than ERM. Figure 1 shows the variance of gradients w.r.t. the number of epochs using SAM and ERM on

CIFAR-100 with various network architectures (experimental details are in Section 4.1). As can be seen, SAM always has a much smaller variance than ERM. Figure 2 shows the squared stochastic gradient norm w.r.t. the number of epochs on *CIFAR-100*. As can be seen, SAM achieves a much smaller $\mathbb{E}_{\mathcal{B}}\|\nabla\mathcal{L}(\mathcal{B};\mathbf{w}_t)\|^2$ than ERM.

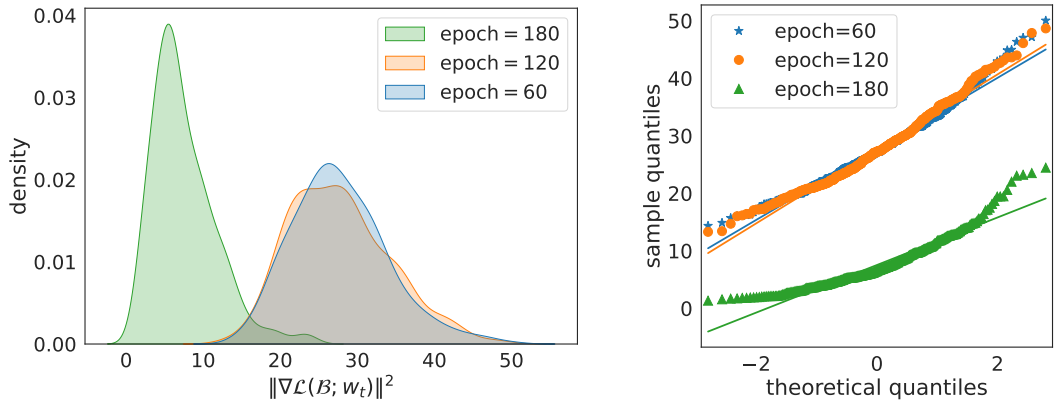
3.2 Adaptive Policy to Employ SAM

As $\mathbb{E}_{\mathcal{B}}\|\nabla\mathcal{L}(\mathcal{B};\mathbf{w}_t)\|^2$ changes over t (Figure 2), it needs to be estimated at every iteration. One can sample a large number of mini-batches and compute the mean of the corresponding stochastic gradient norms. However, this can be computationally expensive as the computational overhead is substantial. To address this problem, we model $\|\nabla\mathcal{L}(\mathcal{B};\mathbf{w}_t)\|^2$ with a distribution and estimate the distribution parameters in an online manner. Figure 3(a) shows $\|\nabla\mathcal{L}(\mathcal{B};\mathbf{w}_t)\|^2$ of 400 mini-batches at different training stages (epoch $= 60$, 120, and 180) on *CIFAR-100* using *ResNet-18*. As can be seen, the distribution follows a Bell curve. Figure 3(b) shows the corresponding quantile-quantile (Q-Q) plot, which is usually used to assess normality: the closer the curve to the line, the distribution is closer to the normal distribution. Both Figures 3(a) and 3(b) suggest that $\|\nabla\mathcal{L}(\mathcal{B};\mathbf{w}_t)\|^2$ can be modeled with a normal distribution $\mathcal{N}(\mu_t, \sigma_t^2)$. We estimate the mean and variance by an exponential moving average (EMA):

$$\mu_t = \delta\mu_{t-1} + (1-\delta)\|\nabla\mathcal{L}(\mathcal{B}_t;\mathbf{w}_t)\|^2, \tag{4}$$

$$\sigma_t^2 = \delta\sigma_{t-1}^2 + (1-\delta)(\|\nabla\mathcal{L}(\mathcal{B}_t;\mathbf{w}_t)\|^2 - \mu_t)^2, \tag{5}$$

where $\delta \in (0,1)$ controls the forgetting rate. Empirically, we use $\delta = 0.9$. Since $\nabla\mathcal{L}(\mathcal{B}_t;\mathbf{w}_t)$ is already available, EMA does not involve any additional gradient calculations (the cost for the norm operator is negligible).



(a) Distributions of stochastic gradient norms.

(b) Q-Q plots of stochastic gradient norms.

Figure 3: Stochastic gradient norms $\{\|\mathcal{L}(\mathcal{B};\mathbf{w}_t)\|^2 : \mathcal{B} \sim \mathcal{D}\}$ of *ResNet-18* on *CIFAR-100* are approximately normally distributed. Best viewed in color.

Using μ_t and σ_t^2, we employ SAM only at iterations where $\|\nabla\mathcal{L}(\mathcal{B}_t;\mathbf{w}_t)\|^2$ is relatively large. Specifically, when $\|\nabla\mathcal{L}(\mathcal{B}_t;\mathbf{w}_t)\|^2 \geq \mu_t + c_t\sigma_t$ (where c_t is a threshold), SAM is used; otherwise, ERM is used. When using the SAM update is more desirable, c_t is small (or even negative). When $c_t \to -\infty$, it reduces to SAM; when $c_t \to \infty$, it becomes ERM. Note that during the early training stage, the model is still underfitting and $\mathbf{w}_t$ is far from the region of final convergence. Thus, minimizing the empirical loss is more important than seeking a locally flat region. Andriushchenko & Flammarion (2022) also empirically observe that the SAM update is more effective in boosting performance towards the end of training. We therefore design a schedule that linearly decreases c_t from λ_2 to λ_1 (which are pre-set values): $c_t = g_{\lambda_1,\lambda_2}(t) \equiv \frac{t}{T}\lambda_1 + \left(1 - \frac{t}{T}\right)\lambda_2$, where T is the total number of iterations. The whole procedure, called Adaptive policy to Employ SAM (**AE-SAM**), is summarized in Algorithm 1.

AE-LookSAM. The proposed adaptive policy can be combined with any SAM variant. Here, we consider integrating it with LookSAM (Liu et al., 2022). When $\|\nabla\mathcal{L}(\mathcal{B}_t;\mathbf{w}_t)\|^2 \geq \mu_t + c_t\sigma_t$, SAM is used and the update direction for $\mathbf{w}_t$ is decomposed into two orthogonal directions as in LookSAM:

(i) the ERM update direction to reduce training loss, and (ii) the direction that biases the model to a flat region. When $\|\nabla\mathcal{L}(\mathcal{B}_t;\mathbf{w}_t)\|^2 < \mu_t + c_t\sigma_t$, ERM is performed and the second direction of previous SAM update is reused to compose an approximate SAM direction. The procedure, called AE-LookSAM, is also shown in Algorithm 1.

Algorithm 1 AE-SAM and AE-LookSAM.

Require: training set $\mathcal{D}$, stepsize η, radius ρ; λ_1 and λ_2 for $g_{\lambda_1,\lambda_2}(t)$; $\mathbf{w}_0$, $\mu_{-1}=0$, $\sigma_{-1}^2 = e^{-10}$, and α for AE-LookSAM;

for $t = 0, \ldots, T-1$ **do**
  sample a mini-batch data $\mathcal{B}_t$ from $\mathcal{D}$;
  compute $\mathbf{g} = \nabla\mathcal{L}(\mathcal{B}_t;\mathbf{w}_t)$;
  update μ_t by (4) and σ_t^2 by (5);
  compute $c_t = g_{\lambda_1,\lambda_2}(t)$;
  if $\|\nabla\mathcal{L}(\mathcal{B}_t;\mathbf{w}_t)\|^2 \geq \mu_t + c_t\sigma_t$ **then**
    $\mathbf{g}_s = \nabla\mathcal{L}(\mathcal{B}_t;\mathbf{w}_t + \rho\nabla\mathcal{L}(\mathcal{B}_t;\mathbf{w}_t))$;
    if AE-LookSAM: decompose $\mathbf{g}_s$ as $\mathbf{g}_v = \mathbf{g}_s - \frac{\mathbf{g}^\top\mathbf{g}_s}{\|\mathbf{g}\|^2}\mathbf{g}$;
  else:
    if AE-SAM: $\mathbf{g}_s = \mathbf{g}$;
    if AE-LookSAM: $\mathbf{g}_s = \mathbf{g} + \alpha\frac{\|\mathbf{g}\|}{\|\mathbf{g}_v\|}\mathbf{g}_v$;
  end if
  $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta\mathbf{g}_s$;
end for
return $\mathbf{w}_T$.

3.3 Convergence Analysis

In this section, we study the convergence of any algorithm that mixes SAM and ERM. Due to the mixing scheme, analyzing convergence is more challenging compared with that of SAM. Let ξ_t be an indicator of whether SAM or ERM is used at iteration t (i.e., $\xi_t = 1$ for SAM, and 0 for ERM). For example, $\xi_t = \mathbb{I}_{\{\mathbf{w}:\|\nabla\mathcal{L}(\mathcal{B}_t;\mathbf{w})\|^2 \geq \mu_t + c_t\sigma_t\}}(\mathbf{w}_t)$ for the proposed AE-SAM, and ξ_t is sampled from a Bernoulli distribution for SS-SAM (Zhao et al., 2022b).

The following assumptions on smoothness and bounded variance of stochastic gradients are standard in the literature on non-convex optimization (Ghadimi & Lan, 2013; Reddi et al., 2016) and SAM (Andriushchenko & Flammarion, 2022; Abbas et al., 2022; Qu et al., 2022). All proofs are in Appendix A.

Assumption 3.1 (Smoothness)**.** $\mathcal{L}(\mathcal{D};\mathbf{w})$ is β-smooth in $\mathbf{w}$, i.e., $\|\nabla\mathcal{L}(\mathcal{D};\mathbf{w}) - \nabla\mathcal{L}(\mathcal{D};\mathbf{v})\| \leq \beta\|\mathbf{w}-\mathbf{v}\|$.

Assumption 3.2 (Bounded variance of stochastic gradients)**.** $\mathbb{E}_{(\mathbf{x}_i,y_i)\sim\mathcal{D}}\|\nabla\ell(f(\mathbf{x}_i;\mathbf{w}),y_i) - \nabla\mathcal{L}(\mathcal{D};\mathbf{w})\|^2 \leq \sigma^2$.

Theorem 3.3. *Let b be the mini-batch size. If $\eta = \frac{1}{4\beta\sqrt{T}}$ and $\rho = 1/T^{\frac{1}{4}}$, algorithms mixing SAM and ERM satisfy*

$$\min_{0\leq t\leq T-1}\mathbb{E}\|\nabla\mathcal{L}(\mathcal{D};\mathbf{w}_t)\|^2 \leq \frac{32\beta\left(\mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_0) - \mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_T)\right)}{\sqrt{T}\,(7-6\zeta)} + \frac{(1+\zeta+5\beta^2\zeta)\sigma^2}{b\sqrt{T}\,(7-6\zeta)}, \tag{6}$$

where $\zeta = \frac{1}{T}\sum_{t=0}^{T-1}\xi_t \in [0,1]$.

Note that ζ is the fraction of SAM updates, and a larger ζ leads to a larger upper bound in Eq.(6). The above theorem can also be used on SAM (which corresponds to $\zeta = 1$), and is shown in Corollary A.2 of Appendix A.1.

4 EXPERIMENTS

In this section, we evaluate the proposed AE-SAM and AE-LookSAM on several standard benchmarks. As the SAM update doubles the computational overhead compared to the ERM update, the training speed is mainly determined by how often the SAM update is used. Hence, we evaluate efficiency by measuring the fraction of SAM updates used:

$$\%\text{SAM} \equiv 100 \times \#\{\text{iterations using SAM}\}/T,$$

where T, the total number of iterations, is the same for all methods.

4.1 *CIFAR-10* AND *CIFAR-100*

Setup. In this section, experiments are conducted on the *CIFAR-10* and *CIFAR-100* datasets (Krizhevsky & Hinton, 2009) using three network architectures: *ResNet-18* (He et al., 2016), *WideResNet-28-10* (denoted *WRN-28-10*) (Zagoruyko & Komodakis, 2016), and *PyramidNet-110* (Han et al., 2017).

Following the setup in (Liu et al., 2022; Foret et al., 2021), we use batch size 128, initial learning rate of 0.1, cosine learning rate schedule, SGD optimizer with momentum 0.9 and weight decay 0.0001. The number of training epochs is 300 for *PyramidNet-110*, and 200 for *ResNet-18* and *WideResNet-28-10*. 10% of the training set is used as the validation set. We perform grid search on the radius ρ over $\{0.01, 0.02, 0.05, 0.1, 0.2, 0.5\}$ as in Foret et al. (2021). Similarly, α is selected by grid search over $\{0.1, 0.3, 0.6, 0.9\}$. For the c_t schedule $g_{\lambda_1,\lambda_2}(t)$, $\lambda_1 = -1$ and $\lambda_2 = 1$ for AE-SAM; $\lambda_1 = 0$ and $\lambda_2 = 2$ for AE-LookSAM.

Baselines. The proposed AE-SAM and AE-LookSAM are compared with the following baselines: (i) ERM; (ii) SAM (Foret et al., 2021); and its more efficient variants including (iii) ESAM (Du et al., 2022a) which uses part of the weights to compute the perturbation and part of the samples to compute the SAM update direction. These two techniques can reduce the computational cost, but may not always accelerate SAM, particularly in parallel training (Li et al., 2020); (iv) SS-SAM (Zhao et al., 2022b), which randomly selects SAM or ERM according to a Bernoulli trial with success probability 0.5. This is the scheme with the best performance in (Zhao et al., 2022b); (v) LookSAM (Liu et al., 2022) which uses SAM at every $k = 5$ step. The experiment is repeated five times with different random seeds.

Results. Table 1 shows the testing accuracy and fraction of SAM updates (%SAM), in which methods are grouped based on %SAM. As can be seen, AE-SAM has higher accuracy than SAM while using only 50% of SAM updates. SS-SAM and AE-SAM have comparable %SAM (about 50%), and AE-SAM achieves higher accuracy than SS-SAM (which is statistically significant based on the pairwise t-test at 95% significance level). Finally, LookSAM and AE-LookSAM have comparable %SAM (about 20%), and AE-LookSAM also has higher accuracy than LookSAM. These improvements confirm the adaptive policy is better.

4.2 *ImageNet*

Setup. In this section, we perform experiments on the *ImageNet* (Russakovsky et al., 2015), which contains 1000 classes and 1.28 million images. The *ResNet-50* (He et al., 2016) is used. Following the setup in Du et al. (2022a), we train the network for 90 epochs using a SGD optimizer with momentum 0.9, weight decay 0.0001, initial learning rate 0.1, cosine learning rate schedule, and batch size 512. As in (Foret et al., 2021; Du et al., 2022a), $\rho = 0.05$. For the c_t schedule $g_{\lambda_1,\lambda_2}(t)$, $\lambda_1 = -1$ and $\lambda_2 = 1$ for AE-SAM; $\lambda_1 = 0$ and $\lambda_2 = 2$ for AE-LookSAM. $k = 5$ is used for LookSAM. Experiments are repeated with three different random seeds.

Results. Table 2 shows the testing accuracy and fraction of SAM updates. As can be seen, with only half of the iterations using SAM, AE-SAM achieves comparable (or even better) performance as SAM. Compared with LookSAM, AE-LookSAM has better performance (which is also statistically significant), verifying the proposed adaptive policy is more effective than LookSAM's periodic policy.

Table 1: Means and standard deviations of testing accuracy and fraction of SAM updates (%SAM) on *CIFAR-10* and *CIFAR-100*. Methods are grouped based on %SAM. The highest accuracy in each group is underlined; while the highest accuracy for each network architecture (across all groups) is in bold.

		CIFAR-10		*CIFAR-100*	
		Accuracy	%SAM	Accuracy	%SAM
ResNet-18	ERM	95.41 ±0.03	0.0 ±0.0	78.17 ±0.05	0.0 ±0.0
	SAM (Foret et al., 2021)	96.52 ±0.12	100.0 ±0.0	80.17 ±0.15	100.0 ±0.0
	ESAM (Du et al., 2022a)	96.56 ±0.08	100.0 ±0.0	80.41 ±0.10	100.0 ±0.0
	SS-SAM (Zhao et al., 2022b)	96.40 ±0.16	50.0 ±0.0	80.10 ±0.16	50.0 ±0.0
	AE-SAM	**96.63** ±0.04	50.1 ±0.1	**80.48** ±0.11	49.8 ±0.0
	LookSAM (Liu et al., 2022)	96.32 ±0.12	20.0 ±0.0	79.89 ±0.29	20.0 ±0.0
	AE-LookSAM	96.56 ±0.21	20.0 ±0.1	80.29 ±0.37	20.0 ±0.0
WRN-28-10	ERM	96.34 ±0.12	0.0 ±0.0	81.56 ±0.14	0.0 ±0.0
	SAM (Foret et al., 2021)	97.27 ±0.11	100.0 ±0.0	83.42 ±0.05	100.0 ±0.0
	ESAM (Du et al., 2022a)	97.29 ±0.11	100.0 ±0.0	**84.51** ±0.02	100.0 ±0.0
	SS-SAM (Zhao et al., 2022b)	97.09 ±0.11	50.0 ±0.0	82.89 ±0.02	50.0 ±0.0
	AE-SAM	**97.30** ±0.10	49.5 ±0.1	**84.51** ±0.11	49.6 ±0.0
	LookSAM (Liu et al., 2022)	97.02 ±0.12	20.0 ±0.0	83.70 ±0.12	20.0 ±0.0
	AE-LookSAM	97.15 ±0.08	20.0 ±0.0	83.92 ±0.07	20.2 ±0.0
PyramidNet-110	ERM	96.62 ±0.10	0.0 ±0.0	81.89 ±0.15	0.0 ±0.0
	SAM (Foret et al., 2021)	97.30 ±0.10	100.0 ±0.0	84.46 ±0.05	100.0 ±0.0
	ESAM (Du et al., 2022a)	97.42 ±0.10	100.0 ±0.0	85.39 ±0.05	100.0 ±0.0
	SS-SAM (Zhao et al., 2022b)	97.22 ±0.10	50.0 ±0.0	84.90 ±0.05	50.0 ±0.0
	AE-SAM	**97.45** ±0.09	49.9 ±0.0	**85.58** ±0.10	49.8 ±0.1
	LookSAM (Liu et al., 2022)	97.10 ±0.11	20.0 ±0.0	84.01 ±0.06	20.0 ±0.0
	AE-LookSAM	97.22 ±0.11	20.3 ±0.0	84.80 ±0.13	20.2 ±0.1

Table 2: Means and standard deviations of testing accuracy and fraction of SAM updates (%SAM) on *ImageNet* using *ResNet-50*. Methods are grouped based on %SAM. The highest accuracy in each group is underlined; while the highest across all groups is in bold.

	Accuracy	%SAM
ERM	77.11 ±0.14	0.0 ±0.0
SAM (Foret et al., 2021)	**77.47** ±0.12	100.0 ±0.0
ESAM (Du et al., 2022a)	77.25 ±0.75	100.0 ±0.0
SS-SAM (Zhao et al., 2022b)	77.38 ±0.06	50.0 ±0.0
AE-SAM	77.43 ±0.06	49.4 ±0.0
LookSAM (Liu et al., 2022)	77.13 ±0.09	20.0 ±0.0
AE-LookSAM	77.29 ±0.08	20.3 ±0.0

4.3 Robustness to Label Noise

Setup. In this section, we study whether the more-efficient SAM variants will affect its robustness to training label noise. Following the setup in Foret et al. (2021), we conduct experiments on a corrupted version of *CIFAR-10*, with some of its training labels randomly flipped (while its testing set is kept clean). The *ResNet-18* and *ResNet-32* networks are used. They are trained for 200 epochs using SGD with momentum 0.9, weight decay 0.0001, batch size 128, initial learning rate 0.1, and cosine

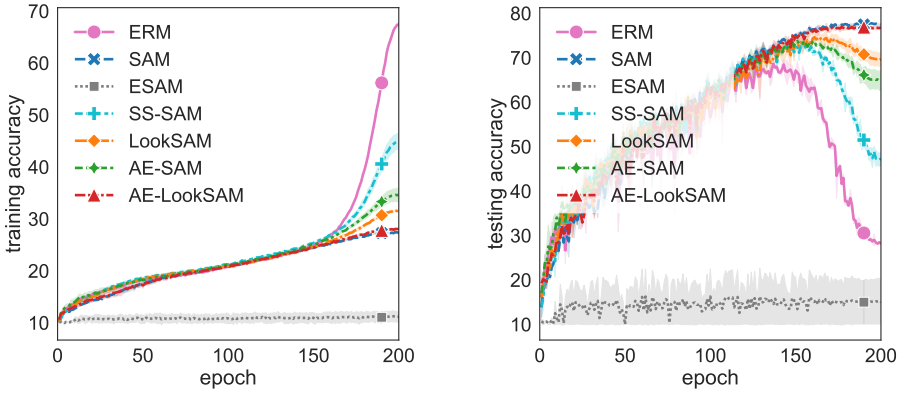
learning rate schedule. For LookSAM, the SAM update is used every $k = 2$ steps.[1] For AE-SAM and AE-LookSAM, we set $\lambda_1 = -1$ and $\lambda_2 = 1$ in their c_t schedules $g_{\lambda_1,\lambda_2}(t)$, such that their fractions of SAM updates (approximately 50%) are comparable with SS-SAM and LookSAM. Experiments are repeated with five different random seeds.

Results. Table 3 shows the testing accuracy and fraction of SAM updates[2]. As can be seen, AE-LookSAM achieves comparable performance with SAM but is faster as only half of the iterations use the SAM update. Compared with ESAM, SS-SAM and LookSAM, AE-LookSAM performs better. The improvement is particularly noticeable at the higher noise levels (e.g., 80%).

Table 3: Testing accuracy and fraction of SAM updates on *CIFAR-10* with different levels of label noise. The best accuracy is in bold and the second best is underlined.

		noise $= 20\%$		noise $= 40\%$		noise $= 60\%$		noise $= 80\%$	
		accuracy	%SAM	accuracy	%SAM	accuracy	%SAM	accuracy	%SAM
ResNet-18	ERM	87.92	0.0	70.82	0.0	49.61	0.0	28.23	0.0
	SAM (Foret et al., 2021)	**94.80**	100.0	<u>91.50</u>	100.0	**88.15**	100.0	**77.40**	100.0
	ESAM (Du et al., 2022a)	94.19	100.0	91.46	100.0	81.30	100.0	15.00	100.0
	SS-SAM (Zhao et al., 2022b)	90.62	50.0	77.84	50.0	61.18	50.0	47.32	50.0
	LookSAM (Liu et al., 2022)	92.72	50.0	88.04	50.0	72.26	50.0	69.72	50.0
	AE-SAM	92.84	50.0	84.17	50.0	73.54	49.9	65.00	50.0
	AE-LookSAM	<u>94.34</u>	49.9	**91.58**	50.0	<u>87.85</u>	50.0	<u>76.90</u>	50.0
ResNet-32	ERM	87.43	0.0	70.82	0.0	46.26	0.0	29.00	0.0
	SAM (Foret et al., 2021)	**95.08**	100.0	91.01	100.0	**88.90**	100.0	**77.32**	100.0
	ESAM (Du et al., 2022a)	93.42	100.0	<u>91.63</u>	100.0	82.73	100.0	10.09	100.0
	SS-SAM (Zhao et al., 2022b)	89.63	50.0	74.17	50.0	58.40	50.0	59.53	50.0
	LookSAM (Liu et al., 2022)	92.49	50.0	86.56	50.0	63.35	50.0	68.01	50.0
	AE-SAM	92.87	50.0	82.85	50.0	71.50	50.0	65.43	50.3
	AE-LookSAM	<u>94.70</u>	50.0	**91.80**	50.0	<u>88.22</u>	50.0	<u>77.03</u>	49.8

Figure 4 shows the training and testing accuracies with number of epochs at a noise level of 80% using *ResNet-18*. Results for other noise levels and *ResNet-32* are shown in Appendix B.2. As can be seen, SAM is robust to the label noise, while ERM and SS-SAM heavily suffer from overfitting. AE-SAM and LookSAM can alleviate the overfitting problem to a certain extent. AE-LookSAM, by combining the adaptive policy with LookSAM, achieves the same high level of robustness as SAM.



(a) Training accuracy. (b) Testing accuracy.

Figure 4: Accuracies with number of training epochs on *CIFAR-10* (with 80% noisy labels) using *ResNet-18*. Best viewed in color.

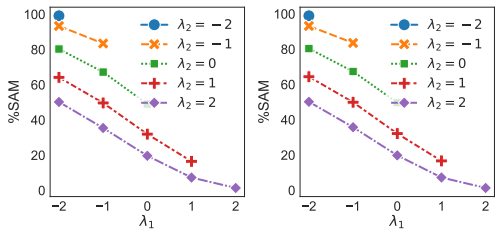
[1]The performance of LookSAM can be sensitive to the value of k. In this experiment, setting $k = 2$ leads to the best performance. Details can be found in Appendix B.2.

[2]The full table, with standard deviation on the results, is shown in Table 4 of Appendix B.2.

4.4 EFFECTS OF λ_1 AND λ_2

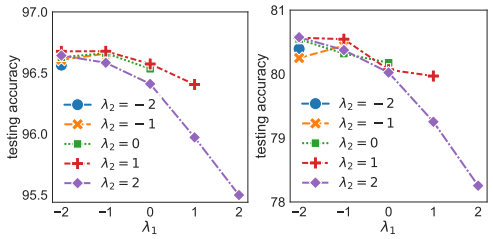
In this experiment, we study the effects of λ_1 and λ_2 on AE-SAM. A similar on AE-LookSAM is shown in Appendix B.2. We use the setup as in Section 4.1, where λ_1 and λ_2 (with $\lambda_1 \leq \lambda_2$) are chosen from $\{0, \pm 1, \pm 2\}$.

Figure 5 shows the effect on the fraction of SAM updates. For a fixed λ_2, increasing λ_1 increases the threshold c_t, and the condition $\|\nabla \mathcal{L}(\mathcal{B}_t; \mathbf{w}_t)\|^2 \geq \mu_t + c_t \sigma_t$ becomes more difficult to satisfy. Thus, as can be seen, the fraction of SAM updates is reduced. The same applies when λ_2 increases. Figure 6 shows the effect on the testing accuracy. For $\lambda_2 \in \{0, 1, 2\}$ (particularly, $\lambda_2 = 2$), increasing λ_1 leads to a reduction in testing accuracy as SAM updates are fewer. For $\lambda_2 \in \{-2, -1\}$, their accuracies are similar as the SAM update is used frequently (%SAM $\geq 80\%$ as shown in Figure 5).



(a) *CIFAR-10*. (b) *CIFAR-100*.

Figure 5: Effects of λ_1 and λ_2 on fraction of SAM updates using *ResNet-18*.

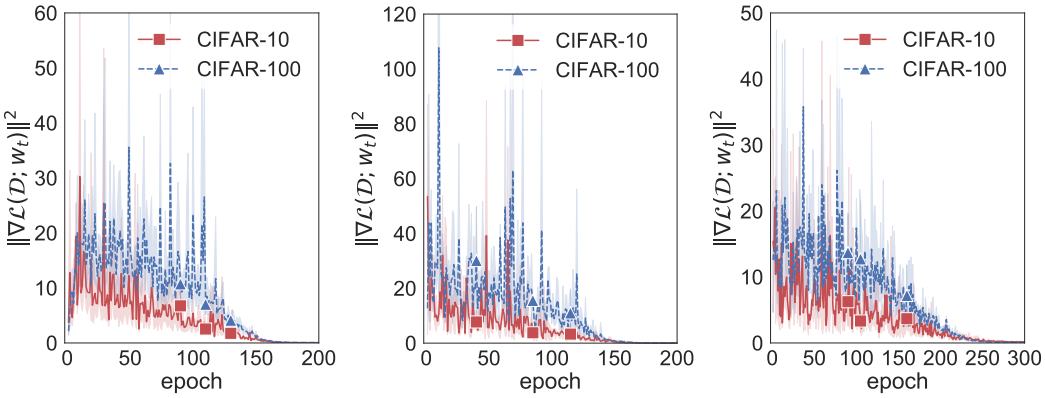


(a) *CIFAR-10*. (b) *CIFAR-100*.

Figure 6: Effects of λ_1 and λ_2 on testing accuracy using *ResNet-18*.

4.5 CONVERGENCE

In this experiment, we validate whether $\mathbf{w}_t$'s, where t is the number of epochs, obtained from AE-SAM can reach critical points of $\mathcal{L}(\mathcal{D}; \mathbf{w})$, as suggested in Theorem 3.3. From the experimental results in Section 4.1, Figure 7 shows $\|\nabla \mathcal{L}(\mathcal{D}; \mathbf{w}_t)\|^2$ w.r.t. t. As can be seen, in all settings, $\|\nabla \mathcal{L}(\mathcal{D}; \mathbf{w}_t)\|^2$ converges to 0. Figure 8 in Appendix B.1 also verifies the convergence of AE-SAM's training loss on *CIFAR-10* and *CIFAR-100*.



(a) *ResNet-18*. (b) *WRN-28-10*. (c) *PyramidNet-110*.

Figure 7: Squared gradient norms of AE-SAM with number of epochs. Best viewed in color.

5 CONCLUSION

In this paper, we proposed an adaptive policy to employ SAM based on the loss landscape geometry. Using the policy, we proposed an efficient algorithm (called AE-SAM) to reduce the fraction of SAM updates during training. We theoretically and empirically analyzed the convergence of AE-SAM. Experimental results on a number of datasets and network architectures verify the effectiveness of the proposed policy. Moreover, the adaptive policy is general and can be combined with SAM variant, as demonstrated by the success of AE-LookSAM.

REFERENCES

Momin Abbas, Quan Xiao, Lisha Chen, Pin-Yu Chen, and Tianyi Chen. Sharp-MAML: Sharpness-aware model-agnostic meta learning. In *International Conference on Machine Learning*, 2022.

Maksym Andriushchenko and Nicolas Flammarion. Towards understanding sharpness-aware minimization. In *International Conference on Machine Learning*, 2022.

Devansh Bisla, Jing Wang, and Anna Choromanska. Low-pass filtering SGD for recovering flat optima in the deep learning optimization landscape. In *International Conference on Artificial Intelligence and Statistics*, 2022.

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

Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. SWAD: Domain generalization by seeking flat minima. In *Neural Information Processing Systems*, 2021.

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

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

Jiawei Du, Hanshu Yan, Jiashi Feng, Joey Tianyi Zhou, Liangli Zhen, Rick Siow Mong Goh, and Vincent Tan. Efficient sharpness-aware minimization for improved training of neural networks. In *International Conference on Learning Representations*, 2022a.

Jiawei Du, Daquan Zhou, Jiashi Feng, Vincent YF Tan, and Joey Tianyi Zhou. Sharpness-aware training for free. In *Neural Information Processing Systems*, 2022b.

Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In *Uncertainty in Artificial Intelligence*, 2017.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 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.

Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Haowei He, Gao Huang, and Yang Yuan. Asymmetric valleys: Beyond sharp and flat local minima. In *Neural Information Processing Systems*, 2019.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

Sepp Hochreiter and Jürgen Schmidhuber. Simplifying neural nets by discovering flat minima. In *Neural Information Processing Systems*, 1994.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *Uncertainty in Artificial Intelligence*, 2018.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *International Conference on Learning Representations*, 2020.

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 *International Conference on Learning Representations*, 2017.

Mohammad Khan, Didrik Nielsen, Voot Tangkaratt, Wu Lin, Yarin Gal, and Akash Srivastava. Fast and scalable Bayesian deep learning by weight-perturbation in adam. In *International Conference on Machine Learning*, 2018.

Minyoung Kim, Da Li, Shell X Hu, and Timothy Hospedales. Fisher SAM: Information geometry and sharpness aware minimisation. In *International Conference on Machine Learning*, 2022.

Yunyong Ko, Dongwon Lee, and Sang-Wook Kim. Not all layers are equal: A layer-wise adaptive approach toward large-scale DNN training. In *ACM Web Conference*, 2022.

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 *International Conference on Machine Learning*, 2021.

Shen Li, Yanli Zhao, Rohan Varma, Omkar Salpekar, Pieter Noordhuis, Teng Li, Adam Paszke, Jeff Smith, Brian Vaughan, and Pritam Damania. PyTorch distributed: experiences on accelerating data parallel training. In *Proceedings of the VLDB Endowment*, 2020.

Yong Liu, Siqi Mai, Xiangning Chen, Cho-Jui Hsieh, and Yang You. Towards efficient and scalable sharpness-aware minimization. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2022.

David A McAllester. PAC-Bayesian model averaging. In *Annual Conference on Computational Learning Theory*, 1999.

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

Antonio Orvieto, Hans Kersting, Frank Proske, Francis Bach, and Aurelien Lucchi. Anticorrelated noise injection for improved generalization. In *International Conference on Machine Learning*, 2022.

Henning Petzka, Michael Kamp, Linara Adilova, Cristian Sminchisescu, and Mario Boley. Relative flatness and generalization. In *Neural Information Processing Systems*, 2021.

Zhe Qu, Xingyu Li, Rui Duan, Yao Liu, Bo Tang, and Zhuo Lu. Generalized federated learning via sharpness aware minimization. In *International Conference on Machine Learning*, 2022.

Sashank J Reddi, Ahmed Hefny, Suvrit Sra, Barnabas Poczos, and Alex Smola. Stochastic variance reduction for nonconvex optimization. In *International Conference on Machine Learning*, pp. 314–323, 2016.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *British Machine Vision Conference*, 2016.

Yang Zhao, Hao Zhang, and Xiuyuan Hu. Penalizing gradient norm for efficiently improving generalization in deep learning. In *International Conference on Machine Learning*, 2022a.

Yang Zhao, Hao Zhang, and Xiuyuan Hu. SS-SAM: Stochastic scheduled sharpness-aware minimization for efficiently training deep neural networks. Preprint arXiv:2203.09962, 2022b.

Mo Zhou, Tianyi Liu, Yan Li, Dachao Lin, Enlu Zhou, and Tuo Zhao. Toward understanding the importance of noise in training neural networks. In *International Conference on Machine Learning*, 2019.

Zhanxing Zhu, Jingfeng Wu, Bing Yu, Lei Wu, and Jinwen Ma. The anisotropic noise in stochastic gradient descent: Its behavior of escaping from sharp minima and regularization effects. In *International Conference on Machine Learning*, 2019.

Juntang Zhuang, Boqing Gong, Liangzhe Yuan, Yin Cui, Hartwig Adam, Nicha C Dvornek, James S. Duncan, and Ting Liu. Surrogate gap minimization improves sharpness-aware training. In *International Conference on Learning Representations*, 2022.

A PROOFS

A.1 PROOF OF THEOREM 3.3

Theorem 3.3. *Let b be the mini-batch size. If $\eta = \frac{1}{4\beta\sqrt{T}}$ and $\rho = 1/T^{\frac{1}{4}}$, algorithms mixing SAM and ERM satisfy*

$$\min_{0\le t\le T-1} \mathbb{E}\|\nabla\mathcal{L}(\mathcal{D};\mathbf{w}_t)\|^2 \le \frac{32\beta\left(\mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_0)-\mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_T)\right)}{\sqrt{T}\left(7-6\zeta\right)} + \frac{(1+\zeta+5\beta^2\zeta)\sigma^2}{b\sqrt{T}\left(7-6\zeta\right)}, \tag{7}$$

where $\zeta = \frac{1}{T}\sum_{t=0}^{T-1}\xi_t \in [0,1]$.

Lemma A.1 (Andriushchenko & Flammarion (2022))**.** *Under Assumptions 3.1 and 3.2 for all t and $\rho > 0$, we have*

$$\mathbb{E}\nabla\mathcal{L}(\mathcal{B}_t;\mathbf{w}+\rho\nabla\mathcal{L}(\mathcal{B}_t;\mathbf{w}))^\top\nabla\mathcal{L}(\mathcal{D};\mathbf{w}) \ge \left(\frac{1}{2}-\rho\beta\right)\|\nabla\mathcal{L}(\mathcal{D};\mathbf{w})\|^2 - \frac{\beta^2\rho^2\sigma^2}{2b}. \tag{8}$$

Proof. Let $\mathbf{g}_t \equiv \frac{1}{b}\sum_{(\mathbf{x}_i,y_i)\in\mathcal{B}_t}\nabla\ell(f(\mathbf{x}_i;\mathbf{w}_t),y_i)$, $\mathbf{h}_t \equiv \frac{1}{b}\sum_{(\mathbf{x}_i,y_i)\in\mathcal{B}_t}\nabla\ell(f(\mathbf{x}_i;\mathbf{w}_t+\rho\mathbf{g}_t),y_i)$, and $\hat{\mathbf{g}}_t \equiv \nabla\mathcal{L}(\mathcal{D};\mathbf{w}_t)$.

By Taylor expansion and $\mathcal{L}(\mathcal{D};\mathbf{w})$ is β-smooth, we have

$$\begin{aligned}
&\mathcal{L}(\mathcal{D};\mathbf{w}_{t+1})\\
\le&\mathcal{L}(\mathcal{D};\mathbf{w}_t)+\hat{\mathbf{g}}_t^\top(\mathbf{w}_{t+1}-\mathbf{w}_t)+\frac{\beta}{2}\|\mathbf{w}_{t+1}-\mathbf{w}_t\|^2\\
\le&\mathcal{L}(\mathcal{D};\mathbf{w}_t)-\eta\hat{\mathbf{g}}_t^\top\left((1-\xi_t)\mathbf{g}_t+\xi_t\mathbf{h}_t\right)+\frac{\beta\eta^2}{2}\|(1-\xi_t)\mathbf{g}_t+\xi_t\mathbf{h}_t\|^2\\
=&\mathcal{L}(\mathcal{D};\mathbf{w}_t)-\eta(1-\xi_t)\hat{\mathbf{g}}_t^\top\mathbf{g}_t-\eta\xi_t\hat{\mathbf{g}}_t^\top\mathbf{h}_t+\frac{\beta\eta^2}{2}\left((1-\xi_t)\|\mathbf{g}_t\|^2+\xi_t\|\mathbf{h}_t\|^2+\underbrace{2\xi_t(1-\xi_t)\mathbf{g}_t^\top\mathbf{h}_t}_{=0}\right) && (9)\\
=&\mathcal{L}(\mathcal{D};\mathbf{w}_t)-\eta(1-\xi_t)\hat{\mathbf{g}}_t^\top\mathbf{g}_t-\eta\xi_t\hat{\mathbf{g}}_t^\top\mathbf{h}_t+\frac{\beta\eta^2}{2}\left((1-\xi_t)\|\mathbf{g}_t\|^2+\xi_t\|\mathbf{h}_t\|^2\right), && (10)
\end{aligned}$$

where we have used $\xi_t(1-\xi_t)=0$ as $\xi_t\in\{0,1\}$, $\xi_t^2=\xi_t$, and $(1-\xi_t)^2=1-\xi_t$ to obtain Eq.(9). Taking expectation w.r.t. $\mathbf{w}_t$ on both sides of (10), we have

$$\mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_{t+1})\le\mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_t)-\eta(1-\xi_t)\mathbb{E}\|\hat{\mathbf{g}}_t\|^2-\eta\xi_t\mathbb{E}\hat{\mathbf{g}}_t^\top\mathbf{h}_t+\frac{\beta\eta^2(1-\xi_t)}{2}\mathbb{E}\|\mathbf{g}_t\|^2+\frac{\beta\eta^2\xi_t}{2}\mathbb{E}\|\mathbf{h}_t\|^2. \tag{11}$$

Claim 1: $\mathbb{E}\|\mathbf{g}_t\|^2=\mathbb{E}\|\mathbf{g}_t-\hat{\mathbf{g}}_t\|^2+\mathbb{E}\|\hat{\mathbf{g}}_t\|^2=\frac{\sigma^2}{b}+\mathbb{E}\|\hat{\mathbf{g}}_t\|^2$, which follows from Assumption 3.2.

Claim 2: $\mathbb{E}\|\mathbf{h}_t\|^2 \leq 2(1+\rho^2\beta^2)\frac{\sigma^2}{b} - (1-2\rho^2\beta^2)\mathbb{E}\|\hat{\mathbf{g}}_t\|^2 + 2\mathbb{E}\hat{\mathbf{g}}_t^\top \mathbf{h}_t$, which is derived as follows:

$$\begin{aligned}
\mathbb{E}\|\mathbf{h}_t\|^2 &= \mathbb{E}\|\mathbf{h}_t - \hat{\mathbf{g}}_t\|^2 - \mathbb{E}\|\hat{\mathbf{g}}_t\|^2 + 2\mathbb{E}\hat{\mathbf{g}}_t^\top \mathbf{h}_t \\
&= 2\mathbb{E}\|\mathbf{h}_t - \mathbf{g}_t\|^2 + 2\mathbb{E}\|\mathbf{g}_t - \hat{\mathbf{g}}_t\|^2 - \mathbb{E}\|\hat{\mathbf{g}}_t\|^2 + 2\mathbb{E}\hat{\mathbf{g}}_t^\top \mathbf{h}_t \\
&\leq 2\rho^2\beta^2\mathbb{E}\|\mathbf{g}_t\|^2 + \frac{2\sigma^2}{b} - \mathbb{E}\|\hat{\mathbf{g}}_t\|^2 + 2\mathbb{E}\hat{\mathbf{g}}_t^\top \mathbf{h}_t && (12) \\
&\leq 2\rho^2\beta^2\left(\frac{\sigma^2}{b} + \mathbb{E}\|\hat{\mathbf{g}}_t\|^2\right) + \frac{2\sigma^2}{b} - \mathbb{E}\|\hat{\mathbf{g}}_t\|^2 + 2\mathbb{E}\hat{\mathbf{g}}_t^\top \mathbf{h}_t && (13) \\
&= 2(1+\rho^2\beta^2)\frac{\sigma^2}{b} - (1-2\rho^2\beta^2)\mathbb{E}\|\hat{\mathbf{g}}_t\|^2 + 2\mathbb{E}\hat{\mathbf{g}}_t^\top \mathbf{h}_t, && (14)
\end{aligned}$$

where Eq.(12) follows from $\|\mathbf{h}_t - \mathbf{g}_t\| \leq \rho\beta\|\mathbf{g}_t\|$ and Assumption 3.2, Eq.(13) follows from the above Claim 1.

Substituting Claims 1 and 2 into (11), we obtain

$$\begin{aligned}
&\mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_{t+1}) \\
&\leq \mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_t) - \eta\left(1-\xi_t\right)\mathbb{E}\|\hat{\mathbf{g}}_t\|^2 - \eta\xi_t\mathbb{E}\hat{\mathbf{g}}_t^\top \mathbf{h}_t + \frac{\beta\eta^2(1-\xi_t)}{2}\left(\frac{\sigma^2}{b} + \mathbb{E}\|\hat{\mathbf{g}}_t\|^2\right) \\
&\quad + \frac{\beta\eta^2\xi_t}{2}\left(2(1+\rho^2\beta^2)\frac{\sigma^2}{b} - (1-2\rho^2\beta^2)\mathbb{E}\|\hat{\mathbf{g}}_t\|^2 + 2\mathbb{E}\hat{\mathbf{g}}_t^\top \mathbf{h}_t\right) && (15) \\
&= \mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_t) - \eta\left(1-\xi_t - \frac{\beta\eta(1-\xi_t)}{2} + \frac{\beta\eta\xi_t(1-2\rho^2\beta^2)}{2}\right)\mathbb{E}\|\hat{\mathbf{g}}_t\|^2 - \eta\xi_t\left(1-\eta\beta\right)\mathbb{E}\hat{\mathbf{g}}_t^\top \mathbf{h}_t \\
&\quad + \left(\frac{\beta\eta^2(1-\xi_t)}{2} + \beta\eta^2\xi_t(1+\rho^2\beta^2)\right)\frac{\sigma^2}{b} \\
&\leq \mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_t) - \eta\left(1-\xi_t - \frac{\beta\eta(1-\xi_t)}{2} + \frac{\beta\eta\xi_t(1-2\rho^2\beta^2)}{2} + \xi_t\left(1-\eta\beta\right)\left(\frac{1}{2} - \rho\beta\right)\right)\mathbb{E}\|\hat{\mathbf{g}}_t\|^2 \\
&\quad + \left(\frac{\beta\eta^2(1-\xi_t)}{2} + \beta\eta^2\xi_t(1+\rho^2\beta^2) + \eta\xi_t\left(1-\eta\beta\right)\frac{\beta^2\rho^2}{2}\right)\frac{\sigma^2}{b} && (16) \\
&\leq \mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_t) - \eta\left(1 - (1+\beta\eta-2\rho\beta)\frac{\xi_t}{2} - \frac{\beta\eta}{2}\right)\mathbb{E}\|\hat{\mathbf{g}}_t\|^2 \\
&\quad + \left(\eta + \xi_t(\eta + 2\eta\rho^2\beta^2 + \beta\rho^2 - \eta\beta^2\rho^2)\right)\frac{\eta\beta\sigma^2}{2b}, && (17)
\end{aligned}$$

where Eq.(15) follows from Claims 1 and 2, Eq.(16) follows from Lemma A.1 and $1-\eta\beta > 0$. As $\eta < \frac{1}{4\beta}$, we have $1+\beta\eta-2\rho\beta \leq 3/2$ and $\beta\eta < 1/4$, thus, $1-(1+\beta\eta-2\rho\beta)\frac{\xi_t}{2} - \frac{\beta\eta}{2} > 0$.

Summing over t on both sides of Eq.(17) and rearranging, we obtain

$$\begin{aligned}
\min_{0\leq t\leq T-1}\mathbb{E}\|\hat{\mathbf{g}}_t\|^2 &\leq \frac{\mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_0) - \mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_T)}{\eta\sum_{t=0}^{T-1}\left(1-(1+\beta\eta-2\rho\beta)\frac{\xi_t}{2} - \frac{\beta\eta}{2}\right)} \\
&\quad + \frac{\sum_{t=0}^{T-1}\left(\eta + \xi_t(\eta + \eta\rho^2\beta^2 + \beta\rho^2)\right)}{\sum_{t=0}^{T-1}\left(1-(1+\beta\eta-2\rho\beta)\frac{\xi_t}{2} - \frac{\beta\eta}{2}\right)}\frac{\beta\sigma^2}{2b} \\
&= \frac{\mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_0) - \mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_T)}{T\eta(1-\frac{\gamma\zeta}{2}-\frac{\beta\eta}{2})} + \frac{T(\eta+\eta\kappa\zeta+\beta\rho^2\zeta)\beta\sigma^2}{2bT(1-\frac{\gamma\zeta}{2}-\frac{\beta\eta}{2})} && (18) \\
&= \frac{\mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_0) - \mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_T)}{T\eta(1-\frac{\gamma\zeta}{2}-\frac{\beta\eta}{2})} + \frac{(1+\kappa\zeta+4\beta^2\zeta)\eta\beta\sigma^2}{2b(1-\frac{\gamma\zeta}{2}-\frac{\beta\eta}{2})} \\
&= \frac{\mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_0) - \mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_T)}{T\eta(1-\frac{\gamma\zeta}{2}-\frac{\beta\eta}{2})} + \frac{(1+\kappa\zeta+4\beta^2\zeta)\sigma^2}{8b\sqrt{T}(1-\frac{\gamma\zeta}{2}-\frac{\beta\eta}{2})} && (19) \\
&\leq \frac{32\beta\left(\mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_0) - \mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_T)\right)}{\sqrt{T}\left(7-6\zeta\right)} + \frac{(1+\zeta+5\beta^2\zeta)\sigma^2}{b\sqrt{T}\left(7-6\zeta\right)}, && (20)
\end{aligned}$$

where $\gamma = 1 + \beta\eta - 2\rho\beta \leq 3/2$, $\kappa = 1 + \rho^2\beta^2$, $\rho^2 = 1/\sqrt{T}$, and $\zeta = \frac{1}{T}\sum_{t=0}^{T-1}\xi_t \in [0,1]$. We thus finish the proof. □

Corollary A.2. *Let b be the mini-batch size. If $\eta = \frac{1}{4\beta\sqrt{T}}$ and $\rho = 1/T^{\frac{1}{4}}$, SAM (Foret et al., 2021) satisfies*

$$\min_{0\leq t\leq T-1} \mathbb{E}\|\nabla\mathcal{L}(\mathcal{D};\mathbf{w}_t)\|^2 \leq \frac{32\beta\left(\mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_0) - \mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_T)\right)}{\sqrt{T}} + \frac{(2+5\beta^2)\sigma^2}{b\sqrt{T}}. \tag{21}$$

Corollary A.3. *Let b be the mini-batch size. If $\eta = \frac{\sqrt{b}}{4\beta\sqrt{T}}$ and $\rho = 1/T^{\frac{1}{4}}$, algorithms mixing SAM and ERM satisfy*

$$\min_{0\leq t\leq T-1} \mathbb{E}\|\nabla\mathcal{L}(\mathcal{D};\mathbf{w}_t)\|^2 \leq \frac{32\beta(\mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_0) - \mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_T))}{\sqrt{Tb}(7-6\zeta)} + \frac{(1+\zeta+5\beta^2\zeta)\sigma^2}{\sqrt{Tb}(7-6\zeta)}, \tag{22}$$

where $\zeta = \frac{1}{T}\sum_{t=0}^{T-1}\xi_t \in [0,1]$.

Proof. It follows from (18) that

$$\begin{aligned}
\min_{0\leq t\leq T-1} \mathbb{E}\|\hat{\mathbf{g}}_t\|^2 &\leq \frac{\mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_0) - \mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_T)}{T\eta(1-\frac{\gamma\zeta}{2}-\frac{\beta\eta}{2})} + \frac{\eta\beta(1+\kappa\zeta+4\beta^2\zeta)\sigma^2}{2b(1-\frac{\gamma\zeta}{2}-\frac{\beta\eta}{2})} && (23)\\
&\leq \frac{4\beta(\mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_0) - \mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_T))}{\sqrt{Tb}(\frac{7}{8}-\frac{3}{4}\zeta)} + \frac{(1+\zeta+5\beta^2\zeta)\sigma^2}{8\sqrt{Tb}(\frac{7}{8}-\frac{3\zeta}{4})} && (24)\\
&= \frac{32\beta(\mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_0) - \mathbb{E}\mathcal{L}(\mathcal{D};\mathbf{w}_T))}{\sqrt{Tb}(7-6\zeta)} + \frac{(1+\zeta+5\beta^2\zeta)\sigma^2}{\sqrt{Tb}(7-6\zeta)}. && (25)
\end{aligned}$$

□

A.2 Convergence of full-batch gradient descent for AE-SAM

Theorem A.4. *Under Assumption 3.1, with full-batch gradient descent, if $\rho < \frac{1}{2\beta}$ and $\eta < \frac{1}{\beta}$, algorithms mixing SAM and ERM satisfy*

$$\min_{0\leq t\leq T-1} \|\nabla\mathcal{L}(\mathcal{D};\mathbf{w}_t)\|^2 \leq \frac{\mathcal{L}(\mathcal{D};\mathbf{w}_0) - \mathcal{L}(\mathcal{D};\mathbf{w}_T)}{T\eta\left(1-\frac{\beta\eta}{2}-\beta\rho\zeta\right)}, \tag{26}$$

where $\zeta = \frac{1}{T}\sum_{t=0}^{T-1}\xi_t \in [0,1]$.

Lemma A.5 (Lemma 7 in Andriushchenko & Flammarion (2022))**.** *Let $\mathcal{L}(\mathcal{D};\mathbf{w})$ be a β-smooth function. For any $\rho > 0$, we have*

$$\nabla\mathcal{L}(\mathcal{D};\mathbf{w})^\top\nabla\mathcal{L}(\mathcal{D};\mathbf{w}+\rho\nabla\mathcal{L}(\mathcal{D};\mathbf{w})) \geq (1-\rho\beta)\|\nabla\mathcal{L}(\mathcal{D};\mathbf{w})\|^2. \tag{27}$$

Proof of Theorem A.4. Let $\mathbf{g}_t \equiv \nabla\mathcal{L}(\mathcal{D};\mathbf{w}_t)$ and $\mathbf{h}_t \equiv \nabla\mathcal{L}(\mathcal{D};\mathbf{w}_t+\rho\nabla\mathcal{L}(\mathcal{D};\mathbf{w}_t))$ be the update direction of ERM and SAM, respectively. By Taylor expansion and $\mathcal{L}(\mathcal{D};\mathbf{w})$ is β-smooth, we have

$$\begin{aligned}
&\mathcal{L}(\mathcal{D};\mathbf{w}_{t+1})\\
&\leq \mathcal{L}(\mathcal{D};\mathbf{w}_t) + \mathbf{g}_t^\top(\mathbf{w}_{t+1}-\mathbf{w}_t) + \frac{\beta}{2}\|\mathbf{w}_{t+1}-\mathbf{w}_t\|^2\\
&\leq \mathcal{L}(\mathcal{D};\mathbf{w}_t) - \eta\mathbf{g}_t^\top\left((1-\xi_t)\mathbf{g}_t + \xi_t\mathbf{h}_t\right) + \frac{\beta\eta^2}{2}\|(1-\xi_t)\mathbf{g}_t + \xi_t\mathbf{h}_t\|^2\\
&= \mathcal{L}(\mathcal{D};\mathbf{w}_t) - \eta(1-\xi_t)\|\mathbf{g}_t\|^2 - \eta\xi_t\mathbf{g}_t^\top\mathbf{h}_t + \frac{\beta\eta^2}{2}\left((1-\xi_t)\|\mathbf{g}_t\|^2 + \xi_t\|\mathbf{h}_t\|^2 + \underbrace{2\xi_t(1-\xi_t)\mathbf{g}_t^\top\mathbf{h}_t}_{=0}\right) && (28)\\
&= \mathcal{L}(\mathcal{D};\mathbf{w}_t) - \eta\left(1-\xi_t-\frac{\beta\eta(1-\xi_t)}{2}\right)\|\mathbf{g}_t\|^2 + \frac{\beta\eta^2\xi_t}{2}\|\mathbf{h}_t\|^2 - \eta\xi_t\mathbf{g}_t^\top\mathbf{h}_t, && (29)
\end{aligned}$$

where we have used $\xi_t(1-\xi_t)=0$ as $\xi_t \in \{0,1\}$, $\xi_t^2=\xi_t$, and $(1-\xi_t)^2=1-\xi_t$ to obtain Eq.(28). As $\|\mathbf{h}_t\|^2=\|\mathbf{h}_t-\mathbf{g}_t\|^2-\|\mathbf{g}_t\|^2+2\mathbf{g}_t^\top\mathbf{h}_t$, it follows from Eq. (29) that

$$
\begin{aligned}
&\mathcal{L}(\mathcal{D};\mathbf{w}_{t+1})\\
=&\mathcal{L}(\mathcal{D};\mathbf{w}_t)-\eta\left(1-\xi_t-\frac{\beta\eta(1-\xi_t)}{2}\right)\|\mathbf{g}_t\|^2+\frac{\beta\eta^2\xi_t}{2}\left(\|\mathbf{h}_t-\mathbf{g}_t\|^2-\|\mathbf{g}_t\|^2+2\mathbf{g}_t^\top\mathbf{h}_t\right)-\eta\xi_t\mathbf{g}_t^\top\mathbf{h}_t\\
\leq&\mathcal{L}(\mathcal{D};\mathbf{w}_t)-\eta\left(1-\xi_t-\frac{\beta\eta(1-\xi_t)}{2}+\frac{\beta\eta\xi_t}{2}\right)\|\mathbf{g}_t\|^2+\frac{\beta\eta^2\xi_t}{2}\|\mathbf{h}_t-\mathbf{g}_t\|^2-\eta(1-\beta\eta)\xi_t\mathbf{g}_t^\top\mathbf{h}_t\\
\leq&\mathcal{L}(\mathcal{D};\mathbf{w}_t)-\eta\left(1-\xi_t-\frac{\beta\eta(1-\xi_t)}{2}+\frac{\beta\eta\xi_t}{2}\right)\|\mathbf{g}_t\|^2+\frac{\beta^3\eta^2\rho^2\xi_t}{2}\|\mathbf{g}_t\|^2-\eta(1-\beta\eta)\xi_t\mathbf{g}_t^\top\mathbf{h}_t \quad (30)\\
=&\mathcal{L}(\mathcal{D};\mathbf{w}_t)-\eta\left(1-\xi_t-\frac{\beta\eta(1-\xi_t)}{2}+\frac{\beta\eta\xi_t}{2}+\frac{\beta^3\eta\rho^2\xi_t}{2}+(1-\beta\eta)(1-\beta\rho)\xi_t\right)\|\mathbf{g}_t\|^2 \quad (31)\\
=&\mathcal{L}(\mathcal{D};\mathbf{w}_t)-\eta\left(1-\frac{\beta\eta(1-\xi_t)}{2}+\frac{\beta\eta\xi_t}{2}+\frac{\beta^3\eta\xi_t\rho^2}{2}-\beta\eta\xi_t-\beta\rho\xi_t+\beta^2\eta\rho\xi_t\right)\|\mathbf{g}_t\|^2\\
\leq&\mathcal{L}(\mathcal{D};\mathbf{w}_t)-\eta\left(1-\frac{\beta\eta}{2}-\beta\rho\xi_t\right)\|\mathbf{g}_t\|^2, \quad (32)
\end{aligned}
$$

where we have used $\|\mathbf{h}_t-\mathbf{g}_t\|^2=\|\nabla\mathcal{L}(\mathcal{D};\mathbf{w}_t+\rho\nabla\mathcal{L}(\mathcal{D};\mathbf{w}_t))-\nabla\mathcal{L}(\mathcal{D};\mathbf{w}_t)\|^2 \leq \beta^2\rho^2\|\nabla\mathcal{L}(\mathcal{D};\mathbf{w}_t)\|^2=\beta^2\rho^2\|\mathbf{g}_t\|^2$ to obtain (30), and Lemma A.5 to obtain (31).

Summing over t from $t=0$ to $T-1$ on both sides of Eq.(32) and rearranging, we have

$$
\sum_{t=0}^{T-1}\eta\left(1-\frac{\beta\eta}{2}-\beta\rho\xi_t\right)\|\mathbf{g}_t\|^2 \leq \mathcal{L}(\mathcal{D};\mathbf{w}_0)-\mathcal{L}(\mathcal{D};\mathbf{w}_T). \quad (33)
$$

As $\rho<\frac{1}{2\beta}$ and $\eta<\frac{1}{\beta}$, it follows that $1-\frac{\beta\eta}{2}-\beta\rho\xi_t>0$ for all t. Thus, Eq. (33) implies

$$
\min_{0\leq t\leq T-1}\|\mathbf{g}_t\|^2 \leq \frac{\mathcal{L}(\mathbf{w}_0)-\mathcal{L}(\mathcal{D};\mathbf{w}_T)}{\sum_{t=0}^{T-1}\eta\left(1-\frac{\beta\eta}{2}-\xi_t\beta\rho\right)}=\frac{\mathcal{L}(\mathcal{D};\mathbf{w}_0)-\mathcal{L}(\mathcal{D};\mathbf{w}_T)}{T\eta\left(1-\frac{\beta\eta}{2}-\beta\rho\zeta\right)} \quad (34)
$$

where $\zeta=\frac{1}{T}\sum_{t=0}^{T-1}\xi_t\in[0,1]$ and we finish the proof. □

B Additional Experiments

B.1 Additional Experimental Results on *CIFAR-10* and *CIFAR-100*

Figure 8 shows the convergence of AE-SAM's training loss on *CIFAR-10* and *CIFAR-100* datasets. As can be seen, AE-SAM achieves convergence in various network architectures.

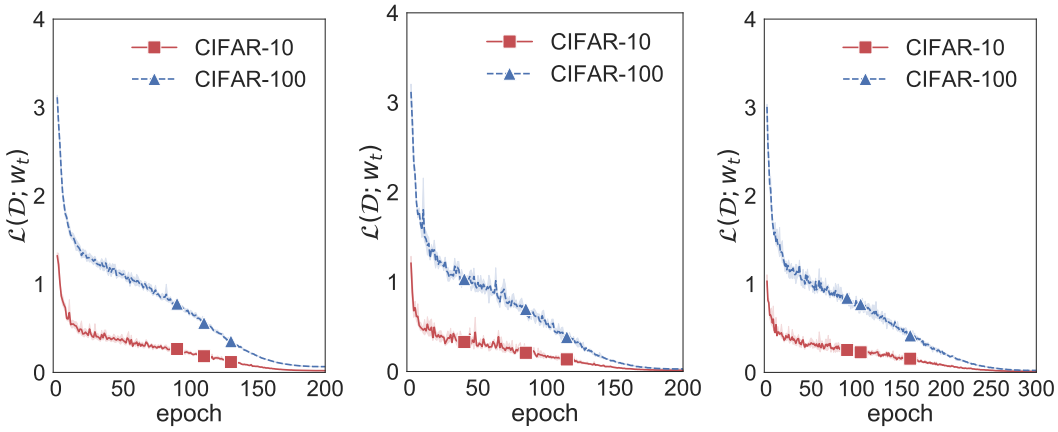
B.2 Additional Experimental Results on *CIFAR-10* with Label Noise

Table 4 reports the testing accuracy and fraction of SAM updates on *CIFAR-10* with label noise.

Effects of k on LookSAM. We conduct an additional experiment on *CIFAR-10* with label noise to show that the performance of LookSAM is sensitive to the choice of k. Table 5 reports the testing accuracy and fraction of SAM updates using LookSAM with different $k\in\{2,3,4,5\}$ for *ResNet-18*. As can be seen, $k=2$ yields better performance than $k=3,4$, or 5, particularly in the higher noisy levels (e.g., 80%).

Effects of λ_1 and λ_2 on AE-LookSAM. We study the effects of λ_1 and λ_2 on AE-LookSAM. We conduct an experiment on *CIFAR-10* with label noise (80% noisy level), where AE-LookSAM achieves comparable performance.

Figure 9 shows the effects of λ_1 and λ_2 on fraction of SAM updates. As can be seen, again, for a fixed λ_2, increasing λ_1 always reduces the fraction of SAM updates.



(a) *ResNet-18*. (b) *WRN-28-10*. (c) *PyramidNet-110*.

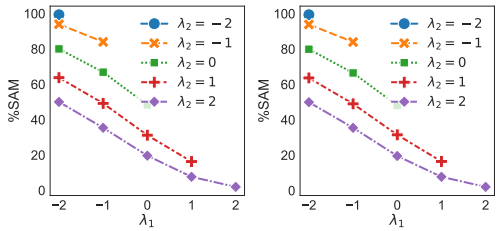
Figure 8: Training loss of AE-SAM with number of epochs on *CIFAR-10* and *CIFAR-100*. Best viewed in color.

Table 4: Testing accuracy and fraction of SAM updates on *CIFAR-10* with different levels of label noise. The best accuracy is in bold and the second best is underlined.

		noise = 20%		noise = 40%		noise = 60%		noise = 80%	
		Accuracy	%SAM	Accuracy	%SAM	Accuracy	%SAM	Accuracy	%SAM
ResNet-18	ERM	87.92 ±0.02	0.0 ±0.0	70.82 ±0.33	0.0 ±0.0	49.61 ±0.39	0.0 ±0.0	28.23 ±0.40	0.0 ±0.0
	SAM (Foret et al., 2021)	**94.80** ±0.05	100.0 ±0.0	91.50 ±0.22	100.0 ±0.0	**88.15** ±0.23	100.0 ±0.0	**77.40** ±0.21	100.0 ±0.0
	ESAM (Du et al., 2022a)	94.19 ±0.10	100.0 ±0.0	91.46 ±0.49	100.0 ±0.0	81.29 ±0.69	100.0 ±0.0	15.00 ±4.89	100.0 ±0.0
	SS-SAM (Zhao et al., 2022b)	90.62 ±0.37	50.0 ±0.0	77.84 ±0.56	50.0 ±0.0	61.18 ±0.87	50.0 ±0.0	47.32 ±1.50	50.0 ±0.0
	AE-SAM	92.84 ±0.25	50.0 ±0.1	84.17 ±0.53	50.0 ±0.0	73.54 ±0.50	49.9 ±0.0	65.00 ±2.25	50.0 ±0.0
	LookSAM (Liu et al., 2022)	92.72 ±0.18	50.0 ±0.0	88.04 ±0.40	50.0 ±0.0	72.26 ±1.75	50.0 ±0.0	69.72 ±1.52	50.0 ±0.0
	AE-LookSAM	94.34 ±0.29	49.9 ±0.0	**91.58** ±0.54	50.0 ±0.0	87.85 ±0.23	50.0 ±0.0	76.90 ±0.32	50.0 ±0.0
ResNet-32	ERM	87.43 ±0.0	0.0 ±0.00	70.82 ±0.98	0.0 ±0.0	46.26 ±0.18	0.0 ±0.0	29.00 ±1.79	0.0 ±0.0
	SAM (Foret et al., 2021)	**95.08** ±0.23	100.0 ±0.0	91.01 ±0.41	100.0 ±0.0	**88.90** ±0.39	100.0 ±0.0	**77.32** ±0.12	100.0 ±0.0
	ESAM (Du et al., 2022a)	93.42 ±0.50	100.0 ±0.0	91.63 ±0.29	100.0 ±0.0	82.73 ±1.21	100.0 ±0.0	10.09 ±0.10	100.0 ±0.0
	SS-SAM (Zhao et al., 2022b)	89.63 ±0.26	50.0 ±0.0	74.17 ±0.47	50.0 ±0.0	58.40 ±2.95	50.0 ±0.0	59.53 ±1.63	50.0 ±0.0
	AE-SAM	92.87 ±0.17	50.0 ±0.0	82.85 ±2.16	50.0 ±0.0	71.50 ±0.74	50.0 ±0.0	65.43 ±3.19	50.3 ±0.0
	LookSAM (Liu et al., 2022)	92.49 ±0.05	50.0 ±0.0	86.56 ±0.92	50.0 ±0.0	63.35 ±0.48	50.0 ±0.0	68.01 ±5.37	50.0 ±0.0
	AE-LookSAM	94.70 ±0.10	50.0 ±0.0	**91.80** ±0.87	50.0 ±0.0	88.22 ±0.27	50.0 ±0.0	77.03 ±0.16	49.8 ±0.1

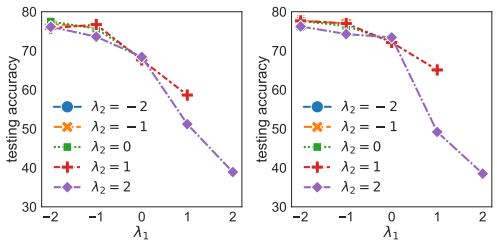
Table 5: Effects of k in LookSAM on *CIFAR-10* with different levels of label noise using *ResNet-18*.

	noise = 20%		noise = 40%		noise = 60%		noise = 80%	
	accuracy	%SAM	accuracy	%SAM	accuracy	%SAM	accuracy	%SAM
LookSAM ($k = 5$)	88.57	20.0	73.90	20.0	56.80	20.0	51.82	20.0
LookSAM ($k = 4$)	89.00	25.0	74.12	25.0	58.17	25.0	52.28	25.0
LookSAM ($k = 3$)	89.07	33.3	75.38	33.3	63.79	33.3	53.87	33.3
LookSAM ($k = 2$)	**92.72**	50.0	**88.04**	50.0	**72.26**	50.0	**69.72**	50.0



(a) *ResNet-18*. (b) *ResNet-32*.

Figure 9: Effects of λ_1 and λ_2 on fraction of SAM updates on *CIFAR-10* with label noise (noisy level of 80%).

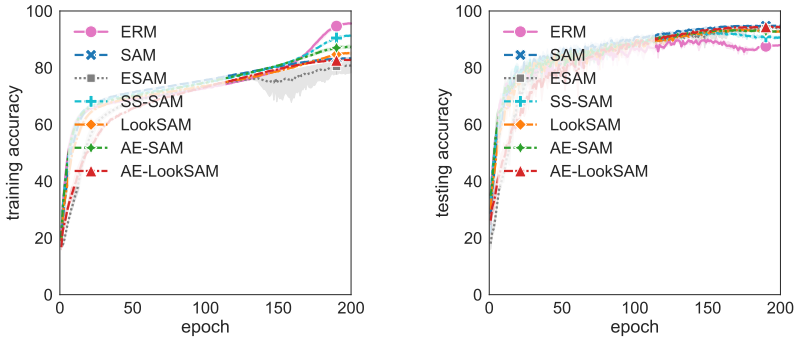


(a) *ResNet-18*. (b) *ResNet-32*.

Figure 10: Effects of λ_1 and λ_2 on testing accuracy on *CIFAR-10* with label noise (noisy level of 80%).

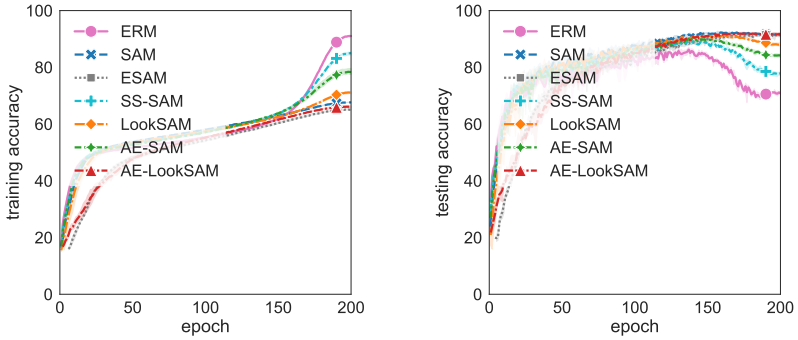
Figure 10 shows the effects of λ_1 and λ_2 on testing accuracy of AE-SAM. We can see that for a fixed $\lambda_2 \in \{0, 1, 2\}$ (particularly $\lambda_2 = 2$), increasing λ_1 always leads a reduction in testing accuracy.

Curves of training and testing accuracy. For *ResNet-18*, Figures 11, 12, and 13 show the curves of accuracies at noise levels of 20%, 40%, and 60%, respectively; For *ResNet-32*, Figures 14, 15, , 16, and 17 show the curves of accuracies at noise levels of 20%, 40%, 60%, and 80%, respectively. As can be seen, in all settings, AE-LookSAM is robust to labels noise like SAM.



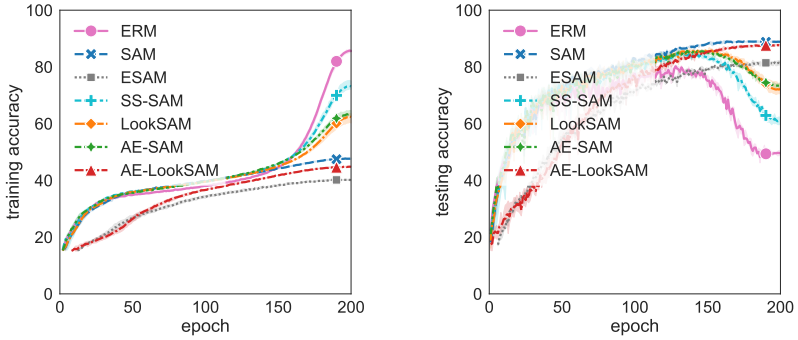
(a) Training. (b) Testing.

Figure 11: Accuracies with number of epochs on CIFAR-10 with 20% noisy level using *ResNet-18*.



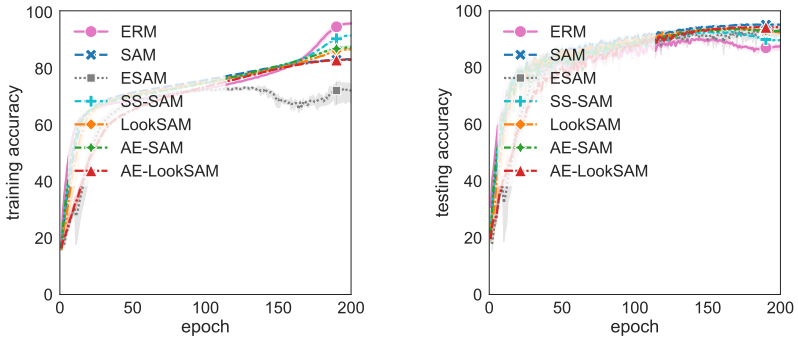
(a) Training. (b) Testing.

Figure 12: Accuracies with number of epochs on CIFAR-10 with 40% noisy level using *ResNet-18*.



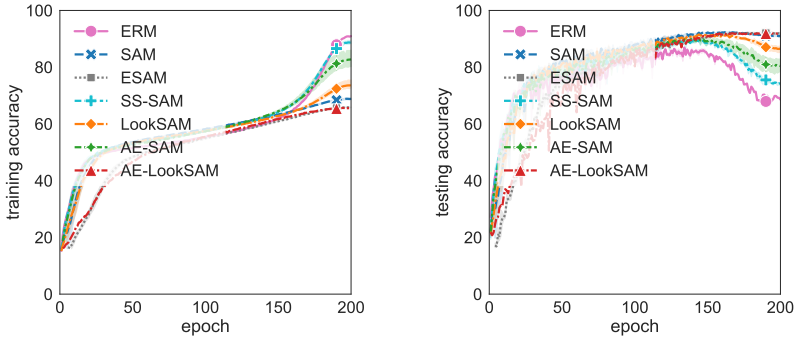
(a) Training. (b) Testing.

Figure 13: Accuracies with number of epochs on CIFAR-10 with 60% noisy level using *ResNet-18*.



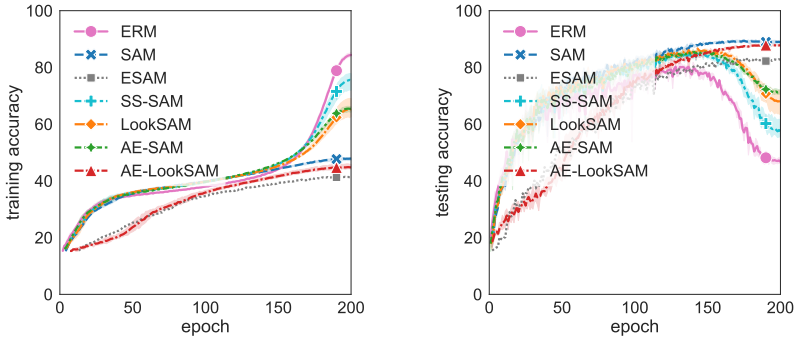
(a) Training. (b) Testing.

Figure 14: Accuracies with number of epochs on CIFAR-10 with 20% noisy level using *ResNet-32*.



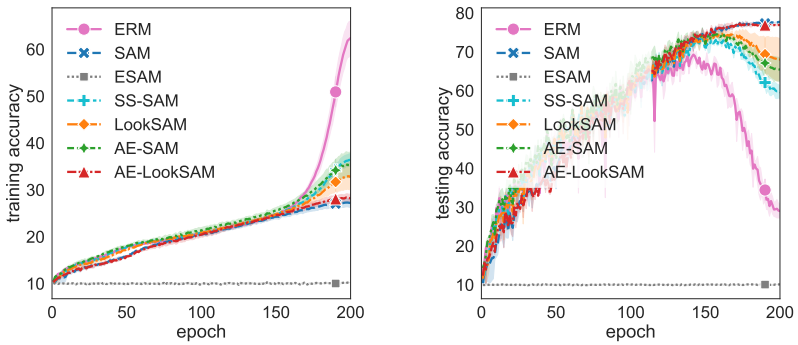
(a) Training. (b) Testing.

Figure 15: Accuracies with number of epochs on CIFAR-10 with 40% noisy level using *ResNet-32*.



(a) Training. (b) Testing.

Figure 16: Accuracies with number of epochs on CIFAR-10 with 60% noisy level using *ResNet-32*.

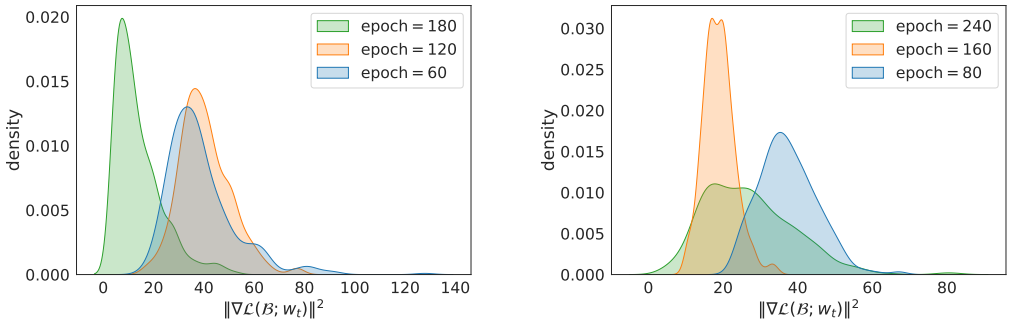


(a) Training. (b) Testing.

Figure 17: Accuracies with number of epochs on CIFAR-10 with 80% noisy level using *ResNet-32*.

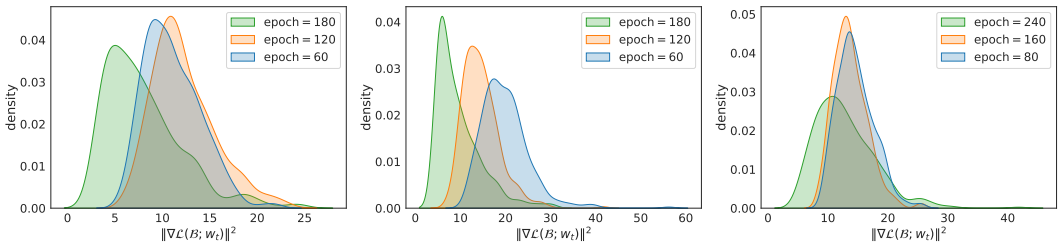
B.3 DISTRIBUTION OF STOCHASTIC GRADIENT NORM

Besides Figure 3(a) for *CIFAR-100* using *ResNet-18*, we empirically verify that the distribution of stochastic gradient norms follows a Bell curve for other network architectures and dataset. Figure 18 shows the distribution for *WRN-28-10* and *PyramidNet-110* on *CIFAR-100*, while Figure 19 shows the distributions for *CIFAR-10*. As can be seen, in all settings, the distribution follows a Bell curve.



(a) *WRN-28-10*. (b) *PyramidNet-110*.

Figure 18: Distributions of stochastic gradient norms on *CIFAR-100*.



(a) *ResNet-18*. (b) *WRN-28-10*. (c) *PyramidNet-110*.

Figure 19: Distributions of stochastic gradient norms on *CIFAR-10*.

Table 6 shows the effects of perturbance radius ρ on the testing accuracy of AE-SAM. As can be seen, AE-SAM is insensitive over a large range of ρ (e.g., $\rho \in \{0.05, 0.1\}$). In experiments, $\rho = 0.05$ and 0.1 are chosen for *CIFAR-10* and *CIFAR-100*, respectively, as is common in the SAM literature (e.g., SAM and ESAM).

Table 6: Effect of ρ on AE-SAM using *ResNet-18*. The highest accuracy is in bold.

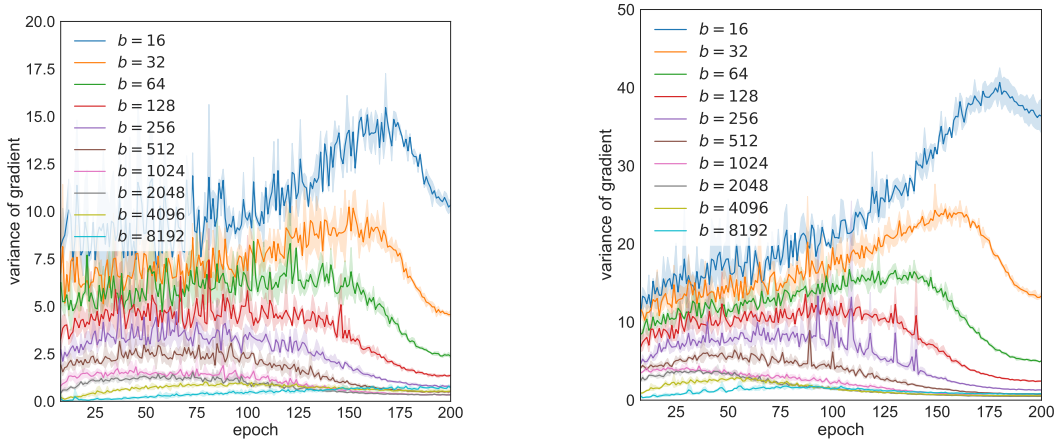
	CIFAR-10		*CIFAR-100*	
	Accuracy	%SAM	Accuracy	%SAM
0.00001	95.82 ±0.26	49.7 ±0.1	79.23 ±0.31	49.9 ±0.0
0.0001	96.03 ±0.16	50.1 ±0.2	79.02 ±0.20	50.1 ±0.1
0.001	96.21 ±0.03	50.0 ±0.0	79.36 ±0.19	49.8 ±0.0
0.01	96.31 ±0.04	49.8 ±0.1	79.55 ±0.08	49.7 ±0.0
0.02	96.44 ±0.03	50.0 ±0.1	79.72 ±0.54	49.8 ±0.0
0.05	**96.63** ±0.04	50.1 ±0.1	80.37 ±0.19	49.9 ±0.0
0.1	96.53 ±0.05	50.0 ±0.1	**80.48** ±0.11	49.8 ±0.0
0.2	96.45 ±0.17	49.8 ±0.1	80.46 ±0.26	49.9 ±0.0
0.5	95.52 ±0.20	49.6 ±0.2	80.28 ±0.22	49.8 ±0.1
1.0	93.98 ±0.19	49.8 ±0.0	80.11 ±0.06	50.0 ±0.1
2.0	91.71 ±0.21	50.3 ±0.2	78.56 ±0.19	50.1 ±0.2
5.0	83.19 ±0.67	50.1 ±0.1	71.37 ±0.62	49.3 ±0.2

B.4 EFFECTS OF BATCH SIZE

We conduct experiments to study the effects of batch size on AE-SAM. Figure 20 show the curves of variance of gradient on *CIFAR-10* and *CIFAR-100* with different batch sizes. As shown, variances are related to batch sizes, which is consistent with the observation in Ko et al. (2022). Table 7 shows the testing accuracy. As can be seen, AE-SAM is insensitive over a large range of batch size, and $b = 128$ is a good choice and is used in the experiments.

Table 7: Effect of batch size on *AE-SAM* using *ResNet-18*. The highest accuracy is in bold.

	CIFAR-10		*CIFAR-100*	
	Accuracy	%SAM	Accuracy	%SAM
16	94.99 ±0.22	49.3 ±0.2	77.40 ±0.56	49.2 ±0.2
32	95.98 ±0.11	49.3 ±0.2	79.88 ±0.13	49.2 ±0.1
64	96.57 ±0.10	49.4 ±0.2	80.23 ±0.30	49.5 ±0.0
128	**96.63** ±0.04	50.1 ±0.1	**80.48** ±0.11	49.8 ±0.0
256	96.47 ±0.11	49.1 ±0.1	79.71 ±0.13	50.2 ±0.2
512	95.93 ±0.08	49.8 ±0.1	78.63 ±0.29	49.9 ±0.2
1024	95.08 ±0.25	50.0 ±0.3	77.18 ±0.13	50.1 ±0.1
2048	94.24 ±0.48	49.3 ±0.0	75.78 ±0.18	49.7 ±0.1
4096	91.35 ±0.74	49.6 ±0.2	72.59 ±0.32	49.5 ±0.2
8192	89.19 ±1.71	50.1 ±0.1	67.83 ±0.64	49.6 ±0.3



(a) *CIFAR-10*. (b) *CIFAR-100*.

Figure 20: Variance of gradient with epoch on *CIFAR* datasets using *ResNet-18*. Best viewed in color.

B.5 RESULTS ON VISION TRANSFORMER

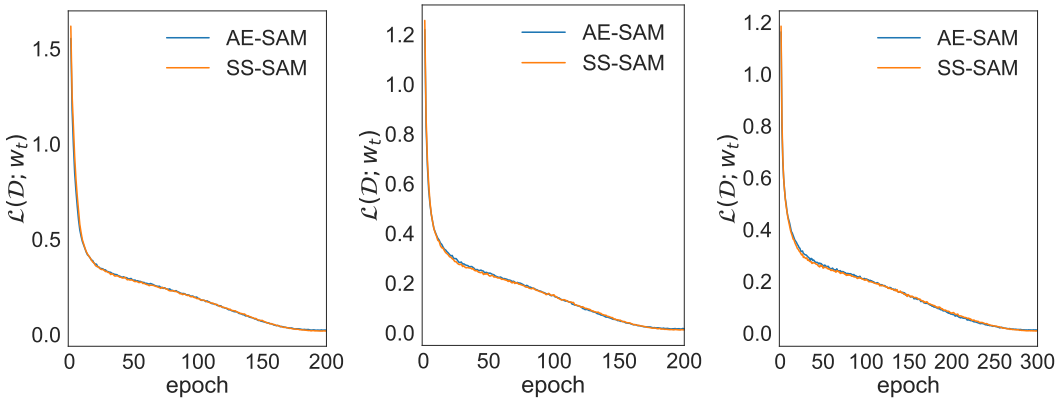
We conduct experiments on vision transformer using the setting (*ViT-S16* and heavy data augmentation) in Zhao et al. (2022a). Table 8 shows the testing accuracy. As can be seen, AE-SAM performs better than SAM while using only 50% of SAM updates. SS-SAM and AE-SAM have comparable %SAM (about 50%), and AE-SAM achieves higher accuracy. As for LookSAM and AE-LookSAM, both use about 20% of SAM updates but AE-LookSAM is better. These improvements confirm the effectiveness of the adaptive policy for *ViT-S16*.

Table 8: Accuracy on *CIFAR-10* and *CIFAR-100* using *ViT-S16*. The highest accuracy in each group is underlined; while the highest accuracy (across all groups) is in bold.

	CIFAR-10		*CIFAR-100*	
	Accuracy	%SAM	Accuracy	%SAM
ERM	86.69 ±0.11	0.0 ±0.0	62.42 ±0.22	0.0 ±0.0
SAM	87.37 ±0.09	100.0 ±0.0	63.23 ±0.25	100.0 ±0.0
ESAM	84.27 ±0.11	100.0 ±0.0	62.11 ±0.15	100.0 ±0.0
Zhao et al. (2022a)	87.71 ±0.19	100.0 ±0.0	63.41 ±0.22	100.0 ±0.0
SS-SAM	87.38 ±0.14	50.0 ±0.0	63.18 ±0.19	50.0 ±0.0
AE-SAM	**87.77** ±0.13	49.7 ±0.1	63.68 ±0.23	49.5 ±0.2
LookSAM	87.12 ±0.20	20.0 ±0.0	63.52 ±0.19	20.0 ±0.0
AE-LookSAM	87.32 ±0.11	20.2 ±0.2	**64.16** ±0.23	20.3 ±0.2

B.6 Convergence curves of AE-SAM and SS-SAM

Figure 21 shows the training losses w.r.t. number of epochs for AE-SAM and SS-SAM. As can be seen, AE-SAM and SS-SAM converge at a comparable speed, which agrees with Theorem 3.3 as both of them have comparable fractions of SAM updates (Table 1).



(a) *ResNet-18*. (b) *WRN-28-10*. (c) *PyramidNet-110*.

Figure 21: Convergence curves on *CIFAR-10*. Best viewed in color.