# CR-SAM: Curvature Regularized Sharpness-Aware Minimization

**Tao Wu**[1], **Tie Luo**[1*], **Donald C. Wunsch II**[2]

[1]Department of Computer Science, Missouri University of Science and Technology
[2]Department of Electrical and Computer Engineering, Missouri University of Science and Technology
{wuta, tluo, dwunsch}@mst.edu

## Abstract

The capacity to generalize to future unseen data stands as one of the utmost crucial attributes of deep neural networks. Sharpness-Aware Minimization (SAM) aims to enhance the generalizability by minimizing worst-case loss using one-step gradient ascent as an approximation. However, as training progresses, the non-linearity of the loss landscape increases, rendering one-step gradient ascent less effective. On the other hand, multi-step gradient ascent will incur higher training cost. In this paper, we introduce a normalized Hessian trace to accurately measure the curvature of loss landscape on *both* training and test sets. In particular, to counter excessive non-linearity of loss landscape, we propose Curvature Regularized SAM (CR-SAM), integrating the normalized Hessian trace as a SAM regularizer. Additionally, we present an efficient way to compute the trace via finite differences with parallelism. Our theoretical analysis based on PAC-Bayes bounds establishes the regularizer's efficacy in reducing generalization error. Empirical evaluation on CIFAR and ImageNet datasets shows that CR-SAM consistently enhances classification performance for ResNet and Vision Transformer (ViT) models across various datasets. Our code is available at https://github.com/TrustAIoT/CR-SAM.

## Introduction

Over the past decade, rapid advancements in deep neural networks (DNNs) have significantly reshaped various pattern recognition domains including computer vision (He et al. 2016), speech recognition (Oord et al. 2018), and natural language processing (Kenton and Toutanova 2019). However, the success of DNNs hinges on their capacity to generalize—how well they would perform on new, unseen data. With their intricate multilayer structures and non-linear characteristics, modern DNNs possess highly non-convex loss landscapes that remain only partially understood. Prior landscape analysis has linked flat local minima to better generalization (Hochreiter and Schmidhuber 1997; Keskar et al. 2016; Dziugaite and Roy 2017; Neyshabur et al. 2017; Jiang et al. 2020). In particular, (Jiang et al. 2020) conducted a comprehensive empirical study on various generalization metrics, revealing that measures based on sharpness exhibit the highest correlation with generalization performance. Recently, (Foret et al. 2021) introduced Sharpness-Aware Minimization (SAM), an efficient technique for minimizing loss landscape sharpness. This method has proven highly effective in enhancing DNN generalization across diverse scenarios. Given SAM's remarkable success and the significance of DNN generalization, a substantial body of subsequent research has emerged (Liu et al. 2022a; Du et al. 2022a,b; Jiang et al. 2023; Liu et al. 2022b).

Specifically, the SAM approach formulates the optimization of neural networks as a minimax problem, where it aims to minimize the maximum loss within a small radius $\rho$ around the parameter $\boldsymbol{w}$. Given that the inner maximization problem is NP-hard, SAM employs a practical sharpness calculation method that utilizes one-step gradient ascent as an approximation. However, our experimentation reveals a notable decline in the accuracy of this one-step approximation as training progresses (see Fig. 1). This phenomenon likely stems from the heightened non-linearity within the loss landscape during later stages of training. Our further investigation highlights a limitation in conventional curvature measures like the Hessian trace and the top eigenvalue of the Hessian matrix. These measures diminish as training advances, incorrectly suggesting reduced curvature and overlooking the actual non-linear characteristics.

Consequently, we posit that the escalating non-linearity in SAM training undermines the precision of approximating and effectiveness of mitigating sharpness. Building upon these insights, we introduce the concept of a *normalized Hessian trace*. This novel metric serves as a dependable indicator of loss landscape non-linearity and behaves consistently across training and testing datasets. Guided by this metric, we propose *Curvature Regularized SAM* (CR-SAM), a novel regularization approach for SAM training. CR-SAM incorporates the normalized Hessian trace to counteract excessive non-linearity effectively.

To calculate the normalized Hessian trace, we present a computationally efficient strategy based on finite differences (FD). This approach enables parallel execution without additional computational burden. Through both theoretical analysis and empirical evaluation, we demonstrate that CR-SAM training converges towards flatter minima, resulting in substantially enhanced generalization performance.

Our main contributions can be summarized as follows:

*Corresponding author.

- We identify that the one-step gradient ascent approximation becomes less effective during the later stages of SAM training. In response, we introduce normalized Hessian trace, a metric that can accurately and consistently characterize the non-linearity of neural network loss landscapes.
- We propose CR-SAM, a novel algorithm that infuses curvature minimization into SAM and thereby enhance the generalizability of deep neural networks. For scalable computation, we devise an efficient technique to approximate the Hessian trace using finite differences (FD). This technique involves only independent function evaluations and can be executed in parallel without additional overhead. Moreover, we also theoretically show the efficacy of CR-SAM in reducing generalization error, leveraging PAC-Bayes bounds.
- Our comprehensive evaluation of CR-SAM spans a diverse range of contemporary DNN architectures. The empirical findings affirm that CR-SAM consistently outperforms both SAM and SGD in terms of improving model generalizability, across multiple datasets including CIFAR10/100 and ImageNet-1k/-C/-R.

## Background and Related Work

Empirical risk minimization (ERM) is a fundamental principle in machine learning for model training on observed data. Given a training dataset $\mathcal{S} = \{(x_i, y_i)\}_{i=1}^n$ drawn i.i.d. from an underlying unknown distribution $\mathcal{D}$, we denote by $f(x; \boldsymbol{w})$ a deep neural network model with trainable parameters $\boldsymbol{w} \in \mathbb{R}^p$, where a differentiable loss function w.r.t. an input $x_i$ is given by $\ell\left(f\left(x_i; \boldsymbol{w}\right), y_i\right)$ and is taken to be the cross entropy loss in this paper. The *empirical loss* can be written as $L_{\mathcal{S}}(\boldsymbol{w}) = \frac{1}{n}\sum_{i=1}^n \ell\left(f\left(x_i; \boldsymbol{w}\right), y_i\right)$ whereas the *population loss* is defined as $L_{\mathcal{D}}(\boldsymbol{w}) = \mathbb{E}_{(x,y)\sim D}[\ell\left(f\left(x; \boldsymbol{w}\right), y\right)]$. The generalization error is defined as the difference between $L_{\mathcal{D}}(\boldsymbol{w})$ and $L_{\mathcal{S}}(\boldsymbol{w})$, i.e., $e(f) = L_{\mathcal{D}}(\boldsymbol{w}) - L_{\mathcal{S}}(\boldsymbol{w})$.

### SAM and Variants

Sharpness-Aware Minimization (SAM) (Foret et al. 2021) is a novel optimization algorithm that directs the search for model parameters within flat regions. Training DNNs with this method has demonstrated remarkable efficacy in enhancing generalization, especially on transformers. SAM introduces a new objective that aims to minimize the maximum loss in the vicinity of weight $\boldsymbol{w}$ within a radius $\rho$:

$$\min_{\boldsymbol{w}} L^{\mathrm{SAM}}(\boldsymbol{w}) \;\text{ where }\; L^{\mathrm{SAM}}(\boldsymbol{w}) = \max_{\|\boldsymbol{v}\|_2 \leq 1} L_{\mathcal{S}}(\boldsymbol{w} + \rho \boldsymbol{v}).$$

Through the minimization of the SAM objective, the neural network's weights undergo updates that shift them towards a smoother loss landscape. As a result, the model's generalization performance is improved. To ensure practical feasibility, SAM adopts two approximations: (1) employs one-step gradient ascent to approximate the inner maximization; (2) simplifies gradient calculation by omitting the second and higher-order terms, i.e.,

$$\nabla L^{\mathrm{SAM}}(\boldsymbol{w}) \approx \nabla L_{\mathcal{S}}\left(\boldsymbol{w} + \rho \frac{\nabla L_{\mathcal{S}}(\boldsymbol{w})}{\|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2}\right). \quad (1)$$

Nevertheless, behind the empirical successes of SAM in training computer vision models (Foret et al. 2021; Chen, Hsieh, and Gong 2022) and natural language processing models (Bahri, Mobahi, and Tay 2021), there are two inherent limitations.

Firstly, SAM introduces a twofold computational overhead to the base optimizer (e.g., SGD) due to the inner maximization process. In response, recent solutions such as LookSAM (Liu et al. 2022a), Efficient SAM (ESAM) (Du et al. 2022a), Sparse SAM (SSAM) (Mi et al. 2022), Sharpness-Aware Training for Free (SAF) (Du et al. 2022b), and Adaptive policy SAM (AE-SAM) (Jiang et al. 2023) have emerged, which propose various strategies to reduce the added overhead.

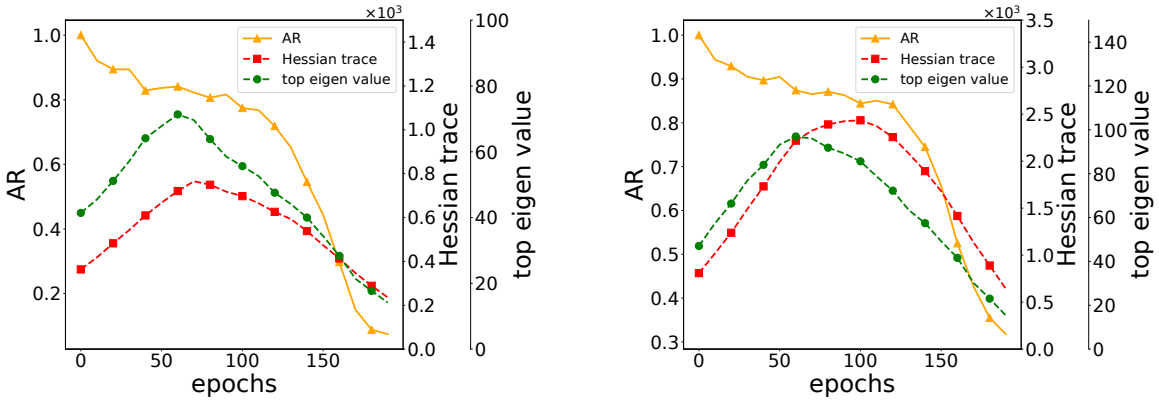
Secondly, the high non-linearity of DNNs' loss landscapes means that relying solely on one-step ascent may not consistently lead to an accurate approximation of the maximum loss. To address this issue, Random SAM (R-SAM) (Liu et al. 2022b) introduced random smoothing to the loss landscape, but its reliance on heuristic methods without strong theoretical underpinnings limits its justification. Empirically, we have included R-SAM in our baseline comparisons, demonstrating our method's superior performance (as seen in Table 2).

Beyond these two limitations, our proposed approach to enhance computational efficiency remains orthogonal to the various SAM variants mentioned above. It can be seamlessly integrated with them, further amplifying efficiency gains.

### Regularization Methods for Generalization

The work (Wilson and Izmailov 2020) contends that model generalization hinges primarily on two traits: the model's *support* and its *inductive biases*. Given the broad applicability of modern DNNs to various datasets, the inductive biases is the remaining crucial factor for guiding a model towards the true data distribution. From a Bayesian standpoint, inductive bias can be viewed as a prior distribution over the parameter space. Classical $\ell_1$ and $\ell_2$ regularization, for instance, correspond to Laplacian and Gaussian prior distributions respectively. In practice, one can employ regularization techniques to instill intended inductive biases, thereby enhancing model generalization. Such regularization can be applied to three core components of modern deep learning models: data, model architecture, and optimization.

**Data-based regularization** involves transforming raw data or generating augmented data to combat overfitting. Methods like label smoothing (Szegedy et al. 2016), Cutout (DeVries and Taylor 2017), Mixup (Zhang et al. 2017), and RandAugment (Cubuk et al. 2020) fall under this category. **Model-based regularization** aids feature extraction and includes techniques such as dropout (Srivastava et al. 2014), skip connections (He et al. 2016), and batch normalization (Ioffe and Szegedy 2015). Lastly, **optimization-based regularization** imparts desired properties like sparsity or complexity into the model. Common methods include weight decay (Krogh and Hertz 1991), gradient norm penalty (Drucker and Le Cun 1992; Zhao, Zhang, and Hu 2022), Jacobian regularization (Sokolić et al. 2017), and confidence penalty (Pereyra et al. 2017). Our proposed cur-




(a) ResNet-18 on CIFAR10

(b) ResNet-18 on CIFAR100

Figure 1: The evolution of approximation ratio (AR), Hessian trace and top eigenvalue of Hessian (the two Y axes on the right) during SAM training on CIFAR10 and CIFAR100 datasets. The continuously decreasing AR indicates an enlarging curvature whereas both of the Hessian-based curvature metrics (which are expected to continuously increase) fail to capture the true curvature of model loss landscape.

vature regularizer in this work aligns with the optimization-based strategies, fostering flatter loss landscapes.

### Flat Minima

Recent research into loss surface geometry underscores the strong correlation between generalization and the flatness of minima reached by DNN parameters. Among various mathematical definitions of flatness, including $\epsilon$-sharpness (Keskar et al. 2016), PAC-Bayes measure (Jiang et al. 2020), Fisher Rao Norm (Liang et al. 2019), and entropy measures (Pereyra et al. 2017; Chaudhari et al. 2019), notable ones include Hessian-based metrics like Frobenius norm (Wu, Wang, and Su 2022a,b), trace of the Hessian (Dinh et al. 2017), largest eigenvalue of the Hessian (Kaur, Cohen, and Lipton 2023), and effective dimensionality of the Hessian (Maddox, Benton, and Wilson 2020). In this work, our focus is on exploring the Hessian trace and its connection to generalization. Akin to our objective, (Liu, Yu, and Lin 2023) also proposes Hessian trace regularization for DNNs. However, (Liu, Yu, and Lin 2023) utilizes the computationally demanding Hutchinson method (Avron and Toledo 2011) with dropout as an unbiased estimator for Hessian trace. In contrast, our method employs finite difference (FD), offering greater computational efficiency and numerical stability. Moreover, our rationale and regularization approach significantly differ from (Liu, Yu, and Lin 2023).

## Methodology

### Our Empirical Findings about SAM Training

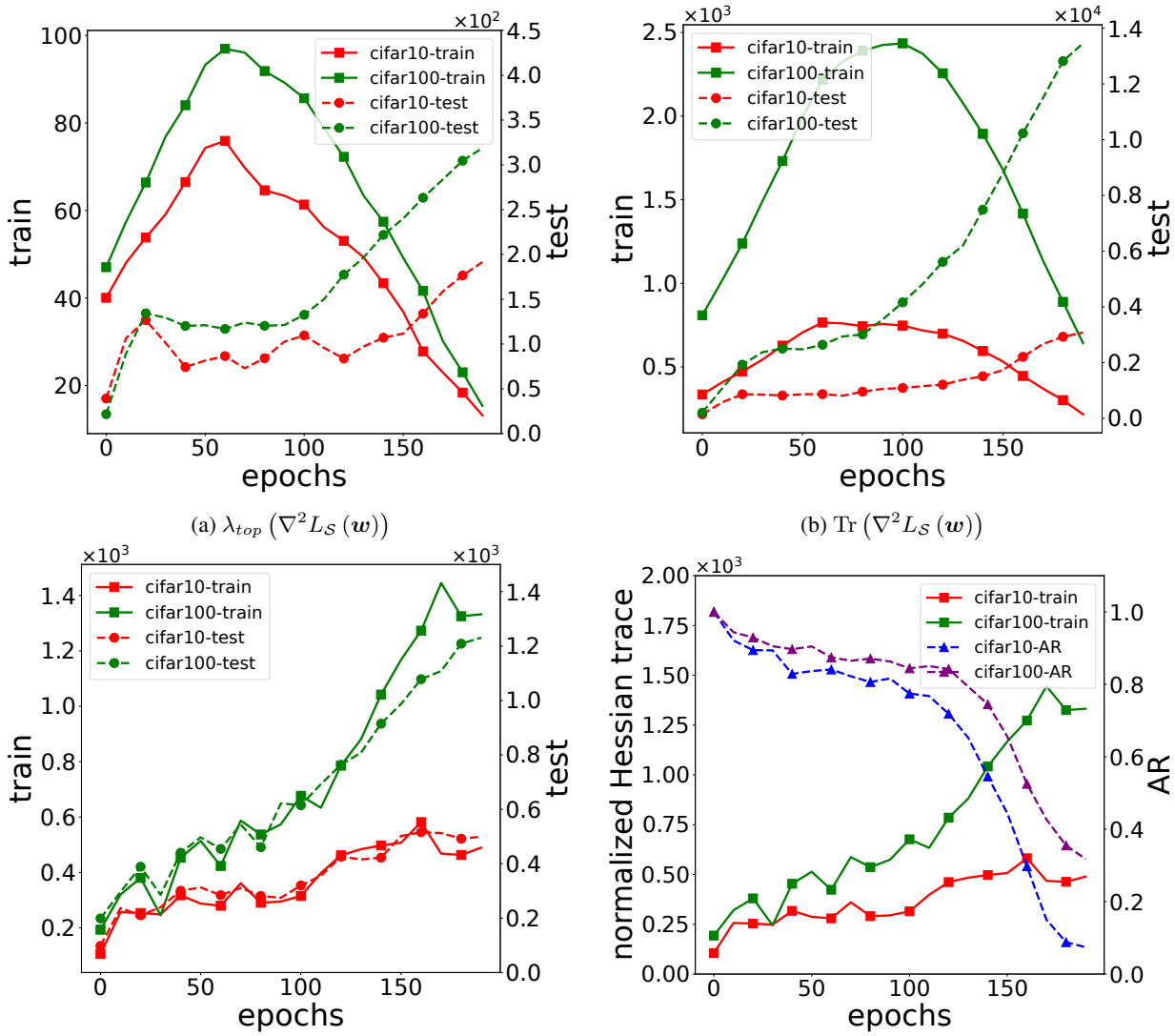
**1) Declining accuracy of one-step approximation.** The optimal solution to the inner maximization in SAM's objective is intractable, which led SAM to resort to an approximation using one-step gradient ascent. However, we found that this approximation's accuracy diminishes progressively as training advances. To show this, we introduce the *approximation ratio (AR)* for sharpness, approximated by one-step gradient ascent, defined as:

$$\mathrm{AR} = \mathbb{E}_{(x,y)\sim D}\left[\frac{\ell\left(f\left(x;\boldsymbol{w}+\boldsymbol{\delta}\right),y\right)-\ell\left(f\left(x;\boldsymbol{w}\right),y\right)}{\ell\left(f\left(x;\boldsymbol{w}+\boldsymbol{\delta}^{*}\right),y\right)-\ell\left(f\left(x;\boldsymbol{w}\right),y\right)}\right] \tag{2}$$

where $\boldsymbol{\delta}$ represents one-step gradient ascent perturbation, and $\boldsymbol{\delta}^*$ denotes the optimal perturbation. An AR closer to 1 indicates a better approximation. Given the infeasibility of obtaining the optimal $\boldsymbol{\delta}^*$, we employ the perturbation from a 20-step gradient ascent as $\boldsymbol{\delta}^*$ and approximate its expectation by sampling 5000 data points from the training set and calculating their average. Our assessment of AR through multiple experiments, illustrated in Fig. 1, reveals its progression during training. Notably, the one-step ascent approximation for sharpness demonstrates diminishing accuracy as training unfolds, with a significant decline in the later stages. This suggests an increasing curvature of the loss landscape as training advances. In the realm of DNNs, the curvature of a function at a specific point is commonly assessed through the Hessian matrix calculated at that point. However, the dependence on gradient scale make Hessian metrics fail to measure the curvature precisely. Specifically, models near convergence of training exhibit smaller gradient norms and inherently correspond to reduced Hessian norms, but does not imply a more linear model.

We show the evolution of conventional curvature metrics like Hessian trace and the top eigenvalue of the Hessian in Fig. 1, both metrics increase initially and then decrease, which fail to capture true loss landscape curvature since AR's consistent decline implies a higher curvature. This phenomenon also verify their dependence on the scaling of model gradients; as gradients decrease near convergence, Hessian-based curvature metrics like Hessian trace and top eigenvalue of the Hessian also decrease.

The degrading effectiveness of the one-step gradient ascent approximation can be theoretically confirmed through a Taylor expansion. The sharpness optimized by SAM in prac-




(a) $\lambda_{top}\left(\nabla^2 L_{\mathcal{S}}\left(\boldsymbol{w}\right)\right)$

(b) $\mathrm{Tr}\left(\nabla^2 L_{\mathcal{S}}\left(\boldsymbol{w}\right)\right)$

(c) $\mathcal{C}(\boldsymbol{w})$

(d) $\mathcal{C}(\boldsymbol{w})$ *v.s.* AR

Figure 2: Evolution of the three curvature metrics (indicated in the captions of subfig.(a)(b)(c)) during SAM training of ResNet-18 on CIFAR-10 and CIFAR-100. In (a)(b)(c), the left/right Y axes denote the metric values on training/test sets, also corresponding to solid/dashed lines. Subfigs. (a) (b) show that the top Hessian eigenvalue and Hessian trace exhibit large discrepancy on train and test sets where values calculated on test set can be 50x more than those on training set. Subfig (c) shows that our proposed normalized Hessian trace shows consistent trends which implies that it well captures the true model geometry. Finally, subfig (d) illustrates that the normalized Hessian trace also reflects (inversely) the phenomenon of decreasing approximation ratio (AR) since they both indicate a growing curvature throughout training.

tice is represented as:

$$R^{\mathrm{SAM}}(\boldsymbol{w}) = L_{\mathcal{S}}\left(\boldsymbol{w} + \rho\frac{\nabla L_{\mathcal{S}}(\boldsymbol{w})}{\|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2}\right) - L_{\mathcal{S}}(\boldsymbol{w}) = \rho\|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2 + O\left(\rho^2\right) \tag{3}$$

Eq. (3) highlights that as training nears convergence, the gradient $\nabla L_{\mathcal{S}}(\boldsymbol{w})$ tends toward 0, causing $R^{\mathrm{SAM}}(\boldsymbol{w})$ to approach 0 as well. Consequently, sharpness ceases to be effectively captured, and SAM training mirrors standard training behavior.

**2) A new metric for accurate curvature characterization.** Our initial observation underscores the limitations of the top Hessian eigenvalue and Hessian trace in capturing loss landscape curvature during SAM training. These metrics suffer from sensitivity to gradient scaling, prompting the need for a more precise curvature characterization. To address this challenge, we introduce a novel curvature metric, *normalized Hessian trace*, defined as follows:

$$\mathcal{C}(\boldsymbol{w}) = \frac{\mathrm{Tr}\left(\nabla^2 L_{\mathcal{S}}\left(\boldsymbol{w}\right)\right)}{\|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2} \tag{4}$$

This metric exhibits continual growth during SAM training, indicating increasing curvature. This behavior aligns well with the decreasing AR of one-step gradient ascent, as depicted in Fig. 2. An additional advantage of the normalized Hessian trace is its consistent trends and values across both training and test sets. In contrast, plain $\mathrm{Tr}\left(\nabla^2 L_{\mathcal{S}}\left(\boldsymbol{w}\right)\right)$ display inconsistent behaviors between these sets, as evidenced in Fig. 2 (a,b,c). This discrepancy questions the viability of solely utilizing Hessian trace or the top Hessian eigenvalue for DNN regularization based on training data.

## Curvature Regularized Sharpness-Aware Minimization (CR-SAM)

For the sake of generalization, it is preferable to steer clear of excessive non-linearity in deep learning models, as it implies highly non-convex loss surfaces. On such models, the challenge of flattening minima (which improves generalization) becomes considerably harder, potentially exceeding the capabilities of gradient-based optimizers. In this context, our proposed normalized Hessian trace (4) can be employed to train deep models with more manageable loss landscapes. However, a direct minimization of $\mathcal{C}(\boldsymbol{w})$ would lead to an elevation in the gradient norm $|\nabla L_{\mathcal{S}}(\boldsymbol{w})|_2$, which could adversely affect generalization (Zhao, Zhang, and Hu 2022). Therefore, we propose to optimize $\mathrm{Tr}\left(\nabla^2 L_{\mathcal{S}}\left(\boldsymbol{w}\right)\right)$ and $\|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2$ separately. Specifically, we penalize both $\mathrm{Tr}\left(\nabla^2 L_{\mathcal{S}}\left(\boldsymbol{w}\right)\right)$ and $\|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2$ but with different extent such that they jointly lead to a smaller $\mathcal{C}(\boldsymbol{w})$. Thus, we introduce our proposed curvature regularizer as:

$$R_c(\boldsymbol{w}) = \alpha \log \mathrm{Tr}\left(\nabla^2 L_{\mathcal{S}}\left(\boldsymbol{w}\right)\right) + \beta \log \|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2 \quad (5)$$

where $\alpha > \beta > 0$ such that the numerator of $\mathcal{C}(\boldsymbol{w})$ is penalized more than the denominator. This regularizer is equivalent to $\alpha \log \mathcal{C}(\boldsymbol{w}) + (\alpha+\beta)\log \|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2$, which is a combination of normalized Hessian trace with gradient norm penalty regularizer. Our regularization strategy can also be justified by analyzing the sharpness:

$$\begin{aligned} R^{\mathrm{True}}(\boldsymbol{w}) &= \max_{\|\boldsymbol{v}\|_2 \le 1} L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v}) - L_{\mathcal{S}}(\boldsymbol{w}) \\ &= \max_{\|\boldsymbol{v}\|_2 \le 1} \left(\rho \boldsymbol{v}^\top \nabla L_{\mathcal{S}}(\boldsymbol{w}) + \frac{\rho^2}{2}\boldsymbol{v}^\top \nabla^2 L_{\mathcal{S}}(\boldsymbol{w})\boldsymbol{v} + O\left(\rho^3\right)\right) \end{aligned}$$

We can see that $\max_{\|\boldsymbol{v}\|_2 \le 1} \rho \boldsymbol{v}^\top \nabla L_{\mathcal{S}}(\boldsymbol{w}) = \rho \|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2$ (cf. (1)). Under the condition that $\boldsymbol{v} \sim N(0, I)$, we have $\mathbb{E}_{\boldsymbol{v}\sim N(0,I)} \boldsymbol{v}^\top \nabla^2 L_{\mathcal{S}}(\boldsymbol{w})\boldsymbol{v} = \mathrm{Tr}\left(\nabla^2 L_{\mathcal{S}}\left(\boldsymbol{w}\right)\right)$ for the second term. However, the first-order term $\|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2$ vanishes at the local minimizers of the loss $L$, and thus the second-order term will become prominent and hence be penalized. Therefore, introducing our regularizer will have the effect of penalizing both the Hessian trace and the gradient norm and thereby reduce the sharpness of a loss landscape.

Informed by our heuristic and theoretical analysis above, our CR-SAM optimizes the following objective:

$$\begin{aligned} &\min_{\boldsymbol{w}} L^{\text{CR-SAM}}(\boldsymbol{w}) \\ &\text{where } L^{\text{CR-SAM}}(\boldsymbol{w}) = L^{\text{SAM}}(\boldsymbol{w}) + R_c(\boldsymbol{w}) \end{aligned} \quad (6)$$

## Solving Computational Efficiency

Computing the Hessian trace as in $R_c(\boldsymbol{w})$ for very large matrices is computationally intensive, especially for modern over-parameterized DNNs with millions of parameters. To address this issue, we first propose a stochastic estimators for $R_c(\boldsymbol{w})$:

$$\begin{aligned} R_c(\boldsymbol{w}) &= \alpha \log \mathrm{Tr}\left(\nabla^2 L_{\mathcal{S}}\left(\boldsymbol{w}\right)\right) + \beta \log \|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2 \\ &= \mathbb{E}_{\boldsymbol{v}\sim N(0,I)}\left[\alpha \log \boldsymbol{v}^\top \nabla^2 L_{\mathcal{S}}(\boldsymbol{w})\boldsymbol{v} + \beta \log \boldsymbol{v}^\top \nabla L_{\mathcal{S}}(\boldsymbol{w})\right] \end{aligned}$$

which reduces Hessian trace computation to averages of Hessian-vector products. However, the complexity of computing the Hessian-vector products in the above estimator is still high for optimizers in large scale problems. Hence, we further propose an approximation based on finite difference (FD) which not only reduces the computational complexity, but also makes the computation *parallelizable*.

**Theorem 1.** *If $L_{\mathcal{S}}(\boldsymbol{w})$ is 2-times-differentiable at $\boldsymbol{w}$, with $\boldsymbol{v} \sim N(0, I)$ , by finite difference we have*

$$\begin{cases} \boldsymbol{v}^\top \nabla L_{\mathcal{S}}(\boldsymbol{w}) = \frac{1}{2\rho}(L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v}) - L_{\mathcal{S}}(\boldsymbol{w}-\rho\boldsymbol{v})) + o\left(\epsilon^2\right); \\ \boldsymbol{v}^\top \nabla^2 L_{\mathcal{S}}(\boldsymbol{w})\boldsymbol{v} = \dfrac{1}{\rho^2}(L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v}) + L_{\mathcal{S}}(\boldsymbol{w}-\rho\boldsymbol{v}) \\ \qquad\qquad - 2L_{\mathcal{S}}(\boldsymbol{w})) + o\left(\epsilon^3\right). \end{cases}$$

By Theorem 1, we can instantiate $R_c(\boldsymbol{w})$ as:

$$\begin{aligned} R_c(\boldsymbol{w}) &= \mathbb{E}_{\boldsymbol{v}\sim N(0,I)}\Big[\alpha \log\left(L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v}) + L_{\mathcal{S}}(\boldsymbol{w}-\rho\boldsymbol{v})\right. \\ &\left. - 2L_{\mathcal{S}}(\boldsymbol{w})\right) + \beta \log\left(L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v}) - L_{\mathcal{S}}(\boldsymbol{w}-\rho\boldsymbol{v})\right)\Big] \\ &+ const. \end{aligned} \quad (7)$$

The above formulation involves an expectation over $\boldsymbol{v}$, which uniformly penalizes expected curvature across all directions. Previous studies (Fawzi et al. 2018; Moosavi-Dezfooli et al. 2019) highlight that gradient directions represent high-curvature directions. Hence, we choose to optimize over perturbations solely along gradient directions, approximating $R_c(\boldsymbol{w})$ by considering $\boldsymbol{v} = \nabla L_{\mathcal{S}}(\boldsymbol{w})$. Additionally, the terms $L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v})$ and $L_{\mathcal{S}}(\boldsymbol{w}-\rho\boldsymbol{v})$ can be computed in parallel as shown in Fig. 3.

We offer a meaningful interpretation of the finite difference regularizer (7): The second term within $R_c(\boldsymbol{w})$, i.e., $[L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v}) - L_{\mathcal{S}}(\boldsymbol{w}-\rho\boldsymbol{v})]$, resembles the surrogate gap $[L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v}) - L_{\mathcal{S}}(\boldsymbol{w})]$ as introduced in (Zhuang et al. 2022). However, unlike solely focusing on optimizing the ridge (locally worst-case perturbation) within the $\rho$-bounded neighborhood around the current parameter vector, our proposed regularizer also delves into the valley (locally best-case perturbation) of the DNN loss landscape, with their loss discrepancies similarly constrained by $R_c(\boldsymbol{w})$. Additionally, by expressing the first term within $R_c(\boldsymbol{w})$ as $[L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v}) - L_{\mathcal{S}}(\boldsymbol{w})] - [L_{\mathcal{S}}(\boldsymbol{w}) - L_{\mathcal{S}}(\boldsymbol{w}-\rho\boldsymbol{v})]$, our approach encourages minimizing the disparity between the worst-case perturbed sharpness and the best-case perturbed sharpness. In essence, our strategy jointly optimizes the worst-case and best-case perturbations within the parameter space neighborhood, promoting a smoother, flatter loss landscape with fewer excessive wavy ridges and valleys.

The full pseudo-code of our CR-SAM training is given in Algorithm 1.

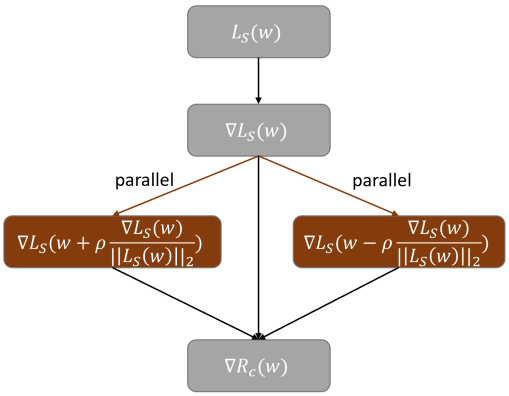



Figure 3: Computing the gradient of $R_c(\boldsymbol{w})$. The two gradient steps are independent of each other and can be perfectly parallelized.

Algorithm 1: Training with CR-SAM

**Input:** Training set $\mathcal{S}$; DNN model $f(x; \boldsymbol{w})$; Loss funtion $\ell\left(f\left(x_i; \boldsymbol{w}\right), y_i\right)$; Batch size $B$; Learning rate $\eta$; Perturbation size $\rho$; regularizer coefficients $\alpha$ and $\beta$
**Output:** model trained by CR-SAM

Parameter initialization $\boldsymbol{w}_0$.
**while** *not converged* **do**
  Sample batch $\mathcal{B} = \{(x_i, y_i)\}_{i=0}^{B}$ from $\mathcal{S}$;
  Compute $\boldsymbol{v} = \frac{\nabla L_{\mathcal{S}}(\boldsymbol{w})}{\|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2}$;
  Compute $L_{\mathcal{S}}(\boldsymbol{w} + \rho \boldsymbol{v})$ and $L_{\mathcal{S}}(\boldsymbol{w} - \rho \boldsymbol{v})$ ;
  Compute $\nabla R_c(\boldsymbol{w})$ per equation 7;
  $\boldsymbol{w}_{t+1} = \boldsymbol{w}_t - \eta(\nabla \mathcal{L}\left(\boldsymbol{w}_t\right) + R_c\left(\boldsymbol{w}_t\right))$;
**end while**
**return** $\boldsymbol{w}_t$

## Experiments

To assess CR-SAM, we conduct thorough experiments on prominent image classification benchmark datasets: CIFAR-10/CIFAR-100 and ImageNet-1k/-C/-R. Our evaluation encompasses a wide array of network architectures, including ResNet, WideResNet, PyramidNet, and Vision Transformer (ViT), in conjunction with diverse data augmentation techniques. These experiments are implemented using PyTorch and executed on Nvidia A100 and V100 GPUs.

### Training from Scratch on CIFAR-10 / CIFAR-100

**Setup.** In this section, we evaluate CR-SAM using the CIFAR-10/100 datasets (Krizhevsky, Hinton et al. 2009). Our evaluation encompasses a diverse selection of widely-used DNN architectures with varying depths and widths. Specifically, we employ ResNet-18 (He et al. 2016), ResNet-50 (He et al. 2016), Wide ResNet-28-10 (WRN-28-10) (Zagoruyko and Komodakis 2016), and PyramidNet-110 (Han, Kim, and Kim 2017), along with a range of data augmentation techniques, including basic augmentations (horizontal flip, padding by four pixels, and random crop) (Foret et al. 2021), Cutout (DeVries and Taylor 2017), and AutoAugment (Cubuk et al. 2018), to ensure a comprehensive assessment.

Following the setup in (Liu et al. 2022b; Du et al. 2022b), we train all models from scratch for 200 epochs, using batch size 128 and employing a cosine learning rate schedule. We conduct grid search to determine the optimal learning rate, weight decay, perturbation magnitude ($\rho$), coefficient ($\alpha$ and $\beta$) values that yield the highest test accuracy. To ensure a fair comparison, we run each experiment three times with different random seeds. Further details of the setup are provided in Appendix.

**Results.** Refer to Table **??** for a comprehensive overview. CR-SAM consistently outperforms both vanilla SAM and SGD across all configurations on both CIFAR-10 and CIFAR-100 datasets. Notable improvements are observed, such as a 1.11% enhancement on CIFAR-100 with ResNet-18 employing cutout augmentation and a 1.30% boost on CIFAR-100 with WRN-28-10 using basic augmentation.

Furthermore, we empirically observe that CR-SAM exhibits a faster convergence rate in comparison to vanilla SAM (details in Appendix). This accelerated convergence could be attributed to CR-SAM's ability to mitigate excessive curvature, ultimately reducing optimization complexity and facilitating swifter arrival at local minima.

### Training from Scratch on ImageNet-1k/-C/-R

**Setup.** This section details our evaluation on the ImageNet dataset (Deng et al. 2009), containing 1.28 million images across 1000 classes. We assess the performance of ResNet (He et al. 2016) and Vision Transformer (ViT) (Dosovitskiy et al. 2020) architectures. Evaluation is extended to out-of-distribution data, namely ImageNet-C (Hendrycks and Dietterich 2019) and ImageNet-R (Hendrycks et al. 2021). ResNet50, ResNet101, ViT-S/32, and ViT-B/32 are evaluated with Inception-style preprocessing.

For ResNet models, SGD serves as the base optimizer. We follow the setup in (Du et al. 2022a), training ResNet50 and ResNet101 with batch size 512 for 90 epochs. The initial learning rate is set to 0.1, progressively decayed using a cosine schedule. For ViT models, we adopt AdamW (Loshchilov and Hutter 2019) as the base optimizer with parameters $\beta_1 = 0.9$ and $\beta_2 = 0.999$. ViTs are trained with batch size 512 for 300 epochs. Refer to the Appendix for further training specifics.

**Results.** Summarized in Table 2, our results indicate substantial accuracy improvements across various DNN models, including ResNet and ViT, on the ImageNet dataset. Notably, CR-SAM's performance surpasses that of SAM by 1.16% for ResNet-50 and by 1.77% for ViT-B/32. These findings underscore the efficacy of our CR-SAM approach.

### Model Geometry Analysis

CR-SAM aims to reduce the normalized trace of the Hessian to promote flatter minima. Empirical validation of CR-SAM's ability to locate optima with lower curvature is presented through model geometry comparisons among mod-

| | | CIFAR-10 | | | CIFAR-100 | | |
|---|---|---|---|---|---|---|---|
| Model | Aug | SGD | SAM | CR-SAM | SGD | SAM | CR-SAM |
| ResNet-18 | Basic | $95.29_{\pm 0.16}$ | $96.46_{\pm 0.18}$ | $\mathbf{96.95}_{\pm 0.13}$ | $78.34_{\pm 0.22}$ | $79.81_{\pm 0.18}$ | $\mathbf{80.76}_{\pm 0.21}$ |
| | Cutout | $95.96_{\pm 0.13}$ | $96.55_{\pm 0.15}$ | $\mathbf{97.01}_{\pm 0.21}$ | $79.23_{\pm 0.13}$ | $80.15_{\pm 0.17}$ | $\mathbf{81.26}_{\pm 0.19}$ |
| | AA | $96.33_{\pm 0.15}$ | $96.75_{\pm 0.18}$ | $\mathbf{97.27}_{\pm 0.12}$ | $79.05_{\pm 0.17}$ | $81.26_{\pm 0.21}$ | $\mathbf{82.11}_{\pm 0.22}$ |
| ResNet-101 | Basic | $96.35_{\pm 0.12}$ | $96.51_{\pm 0.16}$ | $\mathbf{97.14}_{\pm 0.11}$ | $80.54_{\pm 0.13}$ | $82.11_{\pm 0.12}$ | $\mathbf{83.03}_{\pm 0.17}$ |
| | Cutout | $96.56_{\pm 0.18}$ | $96.95_{\pm 0.13}$ | $\mathbf{97.51}_{\pm 0.24}$ | $81.26_{\pm 0.21}$ | $82.39_{\pm 0.27}$ | $\mathbf{83.46}_{\pm 0.16}$ |
| | AA | $96.78_{\pm 0.14}$ | $97.11_{\pm 0.16}$ | $\mathbf{97.76}_{\pm 0.16}$ | $81.83_{\pm 0.37}$ | $83.25_{\pm 0.47}$ | $\mathbf{84.19}_{\pm 0.23}$ |
| WRN-28-10 | Basic | $95.89_{\pm 0.21}$ | $96.81_{\pm 0.26}$ | $\mathbf{97.36}_{\pm 0.15}$ | $81.84_{\pm 0.13}$ | $83.15_{\pm 0.14}$ | $\mathbf{84.45}_{\pm 0.09}$ |
| | Cutout | $96.89_{\pm 0.07}$ | $97.55_{\pm 0.16}$ | $\mathbf{97.98}_{\pm 0.21}$ | $81.96_{\pm 0.40}$ | $83.47_{\pm 0.15}$ | $\mathbf{84.48}_{\pm 0.13}$ |
| | AA | $96.93_{\pm 0.12}$ | $97.59_{\pm 0.06}$ | $\mathbf{97.94}_{\pm 0.08}$ | $82.16_{\pm 0.11}$ | $83.69_{\pm 0.26}$ | $\mathbf{84.74}_{\pm 0.21}$ |
| PyramidNet-110 | Basic | $96.27_{\pm 0.13}$ | $97.34_{\pm 0.13}$ | $\mathbf{97.89}_{\pm 0.08}$ | $83.27_{\pm 0.12}$ | $84.89_{\pm 0.09}$ | $\mathbf{85.68}_{\pm 0.14}$ |
| | Cutout | $96.79_{\pm 0.13}$ | $97.61_{\pm 0.21}$ | $\mathbf{98.08}_{\pm 0.11}$ | $83.43_{\pm 0.21}$ | $84.97_{\pm 0.17}$ | $\mathbf{85.86}_{\pm 0.21}$ |
| | AA | $96.97_{\pm 0.08}$ | $97.81_{\pm 0.13}$ | $\mathbf{98.26}_{\pm 0.11}$ | $84.59_{\pm 0.08}$ | $85.76_{\pm 0.23}$ | $\mathbf{86.58}_{\pm 0.14}$ |

Table 1: Results on CIFAR-10 and CIFAR-100. The base optimizer for SAM and CR-SAM is SGD with Momentum (SGD+M).

| Model | Datasets | Vanilla | SAM | R-SAM | CR-SAM |
|---|---|---|---|---|---|
| ResNet-50 | IN-1k | 75.94 | 76.48 | 76.89 | **77.64** |
| | IN-C | 43.64 | 46.03 | 46.19 | **46.94** |
| | IN-R | 21.93 | 23.13 | 22.89 | **23.48** |
| ResNet-101 | IN-1k | 77.81 | 78.64 | 78.71 | **79.12** |
| | IN-C | 48.56 | 51.27 | 51.35 | **51.87** |
| | IN-R | 24.38 | 25.89 | 25.91 | **26.37** |
| ViT-S/32 | IN-1k | 68.40 | 70.23 | 70.39 | **71.68** |
| | IN-C | 43.21 | 45.78 | 45.92 | **46.46** |
| | IN-R | 19.04 | 21.12 | 21.35 | **21.98** |
| ViT-B/32 | IN-1k | 71.25 | 73.51 | 74.06 | **75.28** |
| | IN-C | 44.37 | 46.98 | 47.28 | **48.12** |
| | IN-R | 23.12 | 24.31 | 24.53 | **25.04** |

Table 2: Results on ImageNet, the base optimizer for ResNets and ViTs are SGD+M and AdamW, respectively.

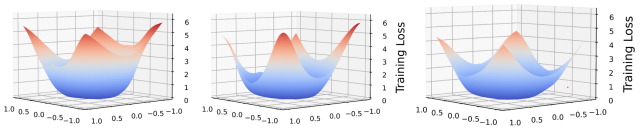
els trained by SGD, SAM, and CR-SAM (see Table 3). Our analysis is based on ResNet-18 trained on CIFAR-100 for 200 epochs using the three optimization methods. Hutchinson's method (Avron and Toledo 2011; Yao et al. 2020) is utilized to compute the Hessian trace, with values obtained from the test set across three independent runs. Notably, the results reveal that CR-SAM significantly reduces both gradient norms and Hessian traces throughout training in contrast to SGD and SAM. This reduction contributes to a smaller normalized Hessian trace, affirming the effectiveness of our proposed regularization strategy.

| **Optimizer** | $\|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2$ | $\mathrm{Tr}\left(\nabla^2 L_{\mathcal{S}}(\boldsymbol{w})\right)$ | $\mathcal{C}(\boldsymbol{w})$ |
|---|---|---|---|
| SGD | 19.97 $\pm 0.52$ | 32673 $\pm 1497$ | 1674 $\pm 78$ |
| SAM | 11.51 $\pm 0.31$ | 14176 $\pm 327$ | 1193 $\pm 59$ |
| CR-SAM | **8.26** $\pm 0.19$ | **7968** $\pm 145$ | **884** $\pm 23$ |

Table 3: Model geometry of ResNet-18 models trained with SGD, SAM and CR-SAM, values are computed on test set.

## Visualization of Landscapes

We visualize the flatness of minima obtained using CR-SAM by plotting loss landscapes of PyramidNet110 trained with SGD, SAM, and CR-SAM on CIFAR-100 for 200 epochs. Employing the visualization techniques from (Li et al. 2018), we depict loss values along two randomly sampled orthogonal Gaussian perturbations around local minima. As depicted in Fig. 4, the visualization illustrates that CR-SAM yields flatter minima compared to SGD and SAM.




(a) SGD (b) SAM (c) CR-SAM

Figure 4: CR-SAM yields flatter loss landscape.

## Conclusion

In this paper, we identify the limitations of the one-step gradient ascent in SAM's inner maximization during training due to the excessive non-linearity of the loss landscape. In addition, existing curvature metrics lack the ability to precisely capture the loss function geometry. To address these issues, we introduce normalized Hessian trace, which offers consistent and accurate characterization of loss function curvature on both training and test data. Building upon this metric, we present CR-SAM, a novel training approach for enhancing neural network generalizability by regularizing our proposed curvature metric. Additionally, to mitigate the overhead of computing the Hessian trace, we incorporate a parallelizable finite difference method. Our comprehensive experiments that span a wide variety of model architectures across popular image classification datasets including CIFAR10/100 and ImageNet-1k/-C/-R, affirm the effectiveness of our proposed CR-SAM training strategy.

## Acknowledgements

This work was supported in part by the National Science Foundation (NSF) under Grant No. 2008878, and in part by the Air Force Research Laboratory (AFRL) and the Lifelong Learning Machines program by DARPA/MTO under Contract No. FA8650-18-C-7831. The research was also sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-22-2-0209.

## References

Avron, H.; and Toledo, S. 2011. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *Journal of the ACM (JACM)*, 58(2): 1–34.

Bahri, D.; Mobahi, H.; and Tay, Y. 2021. Sharpness-aware minimization improves language model generalization. *arXiv preprint arXiv:2110.08529*.

Chaudhari, P.; Choromanska, A.; Soatto, S.; LeCun, Y.; Baldassi, C.; Borgs, C.; Chayes, J.; Sagun, L.; and Zecchina, R. 2019. Entropy-sgd: Biasing gradient descent into wide valleys. *Journal of Statistical Mechanics: Theory and Experiment*, 2019(12): 124018.

Chen, X.; Hsieh, C.-J.; and Gong, B. 2022. When Vision Transformers Outperform ResNets without Pre-training or Strong Data Augmentations. In *International Conference on Learning Representations*.

Cubuk, E. D.; Zoph, B.; Mane, D.; Vasudevan, V.; and Le, Q. V. 2018. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*.

Cubuk, E. D.; Zoph, B.; Shlens, J.; and Le, Q. V. 2020. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, 702–703.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

DeVries, T.; and Taylor, G. W. 2017. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*.

Dinh, L.; Pascanu, R.; Bengio, S.; and Bengio, Y. 2017. Sharp minima can generalize for deep nets. In *International Conference on Machine Learning*, 1019–1028. PMLR.

Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*.

Drucker, H.; and Le Cun, Y. 1992. Improving generalization performance using double backpropagation. *IEEE transactions on neural networks*, 3(6): 991–997.

Du, J.; Yan, H.; Feng, J.; Zhou, J. T.; Zhen, L.; Goh, R. S. M.; and Tan, V. 2022a. Efficient Sharpness-aware Minimization for Improved Training of Neural Networks. In *International Conference on Learning Representations*.

Du, J.; Zhou, D.; Feng, J.; Tan, V.; and Zhou, J. T. 2022b. Sharpness-aware training for free. *Advances in Neural Information Processing Systems*, 35: 23439–23451.

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

Fawzi, A.; Moosavi-Dezfooli, S.-M.; Frossard, P.; and Soatto, S. 2018. Empirical study of the topology and geometry of deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3762–3770.

Foret, P.; Kleiner, A.; Mobahi, H.; and Neyshabur, B. 2021. Sharpness-aware Minimization for Efficiently Improving Generalization. In *International Conference on Learning Representations*.

Han, D.; Kim, J.; and Kim, J. 2017. Deep pyramidal residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5927–5935.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

Hendrycks, D.; Basart, S.; Mu, N.; Kadavath, S.; Wang, F.; Dorundo, E.; Desai, R.; Zhu, T.; Parajuli, S.; Guo, M.; et al. 2021. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 8340–8349.

Hendrycks, D.; and Dietterich, T. 2019. Benchmarking Neural Network Robustness to Common Corruptions and Perturbations. In *International Conference on Learning Representations*.

Hochreiter, S.; and Schmidhuber, J. 1997. Flat minima. *Neural computation*, 9(1): 1–42.

Ioffe, S.; and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, 448–456. pmlr.

Jiang, W.; Yang, H.; Zhang, Y.; and Kwok, J. 2023. An Adaptive Policy to Employ Sharpness-Aware Minimization. In *The Eleventh International Conference on Learning Representations*.

Jiang, Y.; Neyshabur, B.; Mobahi, H.; Krishnan, D.; and Bengio, S. 2020. Fantastic Generalization Measures and Where to Find Them. In *International Conference on Learning Representations*.

Kaur, S.; Cohen, J.; and Lipton, Z. C. 2023. On the maximum hessian eigenvalue and generalization. In *Proceedings on*, 51–65. PMLR.

Kenton, J. D. M.-W. C.; and Toutanova, L. K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of NAACL-HLT*, 4171–4186.

Keskar, N. S.; Mudigere, D.; Nocedal, J.; Smelyanskiy, M.; and Tang, P. T. P. 2016. On large-batch training for

deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images.

Krogh, A.; and Hertz, J. 1991. A simple weight decay can improve generalization. *Advances in neural information processing systems*, 4.

Li, H.; Xu, Z.; Taylor, G.; Studer, C.; and Goldstein, T. 2018. Visualizing the loss landscape of neural nets. *Advances in neural information processing systems*, 31.

Liang, T.; Poggio, T.; Rakhlin, A.; and Stokes, J. 2019. Fisher-rao metric, geometry, and complexity of neural networks. In *The 22nd international conference on artificial intelligence and statistics*, 888–896. PMLR.

Liu, Y.; Mai, S.; Chen, X.; Hsieh, C.-J.; and You, Y. 2022a. Towards efficient and scalable sharpness-aware minimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12360–12370.

Liu, Y.; Mai, S.; Cheng, M.; Chen, X.; Hsieh, C.-J.; and You, Y. 2022b. Random Sharpness-Aware Minimization. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.

Liu, Y.; Yu, S.; and Lin, T. 2023. Hessian regularization of deep neural networks: A novel approach based on stochastic estimators of Hessian trace. *Neurocomputing*, 536: 13–20.

Loshchilov, I.; and Hutter, F. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

Maddox, W. J.; Benton, G.; and Wilson, A. G. 2020. Rethinking parameter counting in deep models: Effective dimensionality revisited. *arXiv preprint arXiv:2003.02139*.

Mi, P.; Shen, L.; Ren, T.; Zhou, Y.; Sun, X.; Ji, R.; and Tao, D. 2022. Make Sharpness-Aware Minimization Stronger: A Sparsified Perturbation Approach. In Oh, A. H.; Agarwal, A.; Belgrave, D.; and Cho, K., eds., *Advances in Neural Information Processing Systems*.

Moosavi-Dezfooli, S.-M.; Fawzi, A.; Uesato, J.; and Frossard, P. 2019. Robustness via curvature regularization, and vice versa. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 9078–9086.

Neyshabur, B.; Bhojanapalli, S.; McAllester, D.; and Srebro, N. 2017. Exploring generalization in deep learning. *Advances in neural information processing systems*, 30.

Oord, A.; Li, Y.; Babuschkin, I.; Simonyan, K.; Vinyals, O.; Kavukcuoglu, K.; Driessche, G.; Lockhart, E.; Cobo, L.; Stimberg, F.; et al. 2018. Parallel wavenet: Fast high-fidelity speech synthesis. In *International conference on machine learning*, 3918–3926. PMLR.

Pereyra, G.; Tucker, G.; Chorowski, J.; Kaiser, L.; and Hinton, G. 2017. Regularizing Neural Networks by Penalizing Confident Output Distributions.

Sokolić, J.; Giryes, R.; Sapiro, G.; and Rodrigues, M. R. 2017. Robust large margin deep neural networks. *IEEE Transactions on Signal Processing*, 65(16): 4265–4280.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research*, 15(1): 1929–1958.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Wilson, A. G.; and Izmailov, P. 2020. Bayesian Deep Learning and a Probabilistic Perspective of Generalization. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *Advances in Neural Information Processing Systems*, volume 33, 4697–4708. Curran Associates, Inc.

Wu, L.; Wang, M.; and Su, W. 2022a. The alignment property of SGD noise and how it helps select flat minima: A stability analysis. *Advances in Neural Information Processing Systems*, 35: 4680–4693.

Wu, L.; Wang, M.; and Su, W. 2022b. When does sgd favor flat minima? a quantitative characterization via linear stability. *arXiv preprint arXiv:2207.02628*.

Yao, Z.; Gholami, A.; Keutzer, K.; and Mahoney, M. W. 2020. Pyhessian: Neural networks through the lens of the hessian. In *2020 IEEE international conference on big data (Big data)*, 581–590. IEEE.

Zagoruyko, S.; and Komodakis, N. 2016. Wide residual networks. *arXiv preprint arXiv:1605.07146*.

Zhang, H.; Cisse, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2017. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*.

Zhao, Y.; Zhang, H.; and Hu, X. 2022. Penalizing gradient norm for efficiently improving generalization in deep learning. In *International Conference on Machine Learning*, 26982–26992. PMLR.

Zhuang, J.; Gong, B.; Yuan, L.; Cui, Y.; Adam, H.; Dvornek, N.; Tatikonda, S.; Duncan, J.; and Liu, T. 2022. Surrogate gap minimization improves sharpness-aware training. *arXiv preprint arXiv:2203.08065*.

# SUPPLEMENTARY MATERIALS

## Proofs

In this section we provide proofs for Theorem 1 and Theorem 2 in the main text.

### Proof of Theorem 1.

**Theorem 2.** *If $L_{\mathcal{S}}(\boldsymbol{w})$ is 2-times-differentiable at $\boldsymbol{w}$, with $\boldsymbol{v} \sim N(0, I)$ , by finite difference we have*

$$\begin{cases} \boldsymbol{v}^\top \nabla L_{\mathcal{S}}(\boldsymbol{w}) = \frac{1}{2\rho}(L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v}) - L_{\mathcal{S}}(\boldsymbol{w}-\rho\boldsymbol{v})) + o\left(\epsilon^2\right); \\ \boldsymbol{v}^\top \nabla^2 L_{\mathcal{S}}(\boldsymbol{w})\boldsymbol{v} = \dfrac{1}{\rho^2}(L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v}) + L_{\mathcal{S}}(\boldsymbol{w}-\rho\boldsymbol{v}) \\ \qquad\qquad - 2L_{\mathcal{S}}(\boldsymbol{w})) + o\left(\epsilon^3\right). \end{cases}$$

**Proof**: Using Taylor polynomial expansion of $L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v})$ and $L_{\mathcal{S}}(\boldsymbol{w}-\rho\boldsymbol{v})$ centered at $\boldsymbol{w}$. We have

$$\begin{cases} L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v}) = L_{\mathcal{S}}(\boldsymbol{w}) + \rho\boldsymbol{v}\nabla L_{\mathcal{S}}(\boldsymbol{w}) + \mathcal{O}\left(\rho^2\right) \\ L_{\mathcal{S}}(\boldsymbol{w}-\rho\boldsymbol{v}) = L_{\mathcal{S}}(\boldsymbol{w}) - \rho\boldsymbol{v}\nabla L_{\mathcal{S}}(\boldsymbol{w}) + \mathcal{O}\left(\rho^2\right) \end{cases} \tag{8}$$

By rearranging the above two equations, we obtain $\boldsymbol{v}^\top \nabla L_{\mathcal{S}}(\boldsymbol{w}) = \frac{1}{2\rho}(L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v}) - L_{\mathcal{S}}(\boldsymbol{w}-\rho\boldsymbol{v})) + O\left(\rho^2\right)$.

We rewrite $\boldsymbol{v}^\top \nabla^2 L_{\mathcal{S}}(\boldsymbol{w})\boldsymbol{v}$ as directional derivatives as $\nabla^2_{\boldsymbol{v}} L_{\mathcal{S}}(\boldsymbol{w})$. Reapplying the above formulation gives

$$\begin{aligned} \nabla^2_{\boldsymbol{v}} L_{\mathcal{S}}(\boldsymbol{w}) &= \frac{1}{\rho}(\nabla_{\boldsymbol{v}} L_{\mathcal{S}}(\boldsymbol{w}+0.5\rho\boldsymbol{v}) - \nabla_{\boldsymbol{v}} L_{\mathcal{S}}(\boldsymbol{w}-0.5\rho\boldsymbol{v})) \\ &\quad + O\left(\rho^2\right) \\ &= \frac{1}{\rho^2}(L_{\mathcal{S}}(\boldsymbol{w}+0.5\rho\boldsymbol{v}+0.5\rho\boldsymbol{v}) - L_{\mathcal{S}}(\boldsymbol{w}+0.5\rho\boldsymbol{v}-0.5\rho\boldsymbol{v}) \\ &\quad - L_{\mathcal{S}}(\boldsymbol{w}-0.5\rho\boldsymbol{v}+0.5\rho\boldsymbol{v}) + L_{\mathcal{S}}(\boldsymbol{w}-0.5\rho\boldsymbol{v}-0.5\rho\boldsymbol{v})) \\ &\quad + O\left(\rho^2\right) \\ &= \frac{1}{\rho^2}(L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v}) + L_{\mathcal{S}}(\boldsymbol{w}-\rho\boldsymbol{v}) - 2L_{\mathcal{S}}(\boldsymbol{w})) + O\left(\rho^2\right) \end{aligned}$$

□

### Proof of PAC-Bayesian generalization error bounds.

**Theorem 3.** *(Stated informally) For any $\delta \in (0,1)$, with probability at least $1-\delta$ over a draw of the training set $\mathcal{S}$ and a solution $\boldsymbol{w}^*$ found by a gradient-based optimizer, by picking $\boldsymbol{v}$ follows standard Gaussian, the following inequality holds:*

$$\mathbb{E}_{\boldsymbol{v}\sim N(0,I)} L_{\mathcal{D}}(\boldsymbol{w}^*+\rho\boldsymbol{v}) \le L_{\mathcal{S}}(\boldsymbol{w}^*) + \frac{\rho^2}{2}\operatorname{Tr}\left(\nabla^2 L\left(\boldsymbol{w}^*\right)\right) + \gamma\|\nabla L_{\mathcal{S}}(\boldsymbol{w}^*)\|_2 + h\left(\|\boldsymbol{w}^*\|_2^2/\rho^2\right)$$

**Proof sketch**: We follow the most basic PAC-Bayesian generalization error bounds as (Alquier, Ridgway, and Chopin 2016; Gat et al. 2022): For any $\lambda > 0$, for any $\delta \in (0,1)$ and for any prior distribution $p$, with probability at least $1-\delta$ over the draw of the training set $S$, the following holds simultaneously for any posterior distribution $q$:

$$\mathbb{E}_{w\sim q}\left[L_D(w)\right] \le \mathbb{E}_{w\sim q}\left[L_{\mathcal{S}}(w)\right] + \frac{1}{\lambda}[C(\lambda,p) + KL(q\|p) + \log(1/\delta)] \tag{9}$$

where $C(\lambda,p) \triangleq \log\left(\mathbb{E}_{w\sim p}\left[e^{\lambda(L_D(w)-L_{\mathcal{S}}(w))}\right]\right)$.

Rearrange the first term as

$$\begin{aligned} \mathbb{E}_{w\sim q}\left[L_{\mathcal{S}}(w)\right] &= L_{\mathcal{S}}(\boldsymbol{w}) + \mathbb{E}_{w\sim q}\left[L_{\mathcal{S}}(w)\right] - L_{\mathcal{S}}(\boldsymbol{w}) \\ &= L_{\mathcal{S}}(\boldsymbol{w}) + \mathop{\mathbb{E}}_{\boldsymbol{v}\sim\mathcal{N}(\mathbf{0},I)}\left[L_{\mathcal{S}}(\boldsymbol{w}+\rho\boldsymbol{v})\right] - L_{\mathcal{S}}(\boldsymbol{w}) \\ &= L_{\mathcal{S}}(\boldsymbol{w}) + \frac{\rho^2}{2}\operatorname{Tr}\left(\nabla^2 L\left(\boldsymbol{w}\right)\right) \end{aligned}$$

From (Gat et al. 2022), we have $C(\lambda,p) \le \gamma\|\nabla L_{\mathcal{S}}(\boldsymbol{w})\|_2$ and $KL(q\|p)$ is a function of $\|\boldsymbol{w}\|_2^2$ when $\boldsymbol{v} \sim N(0,I)$.

□

## Faster and Smoother Convergence

The convergence in a single run of SAM vs. CR-SAM is presented in Fig. 5. See caption for details.

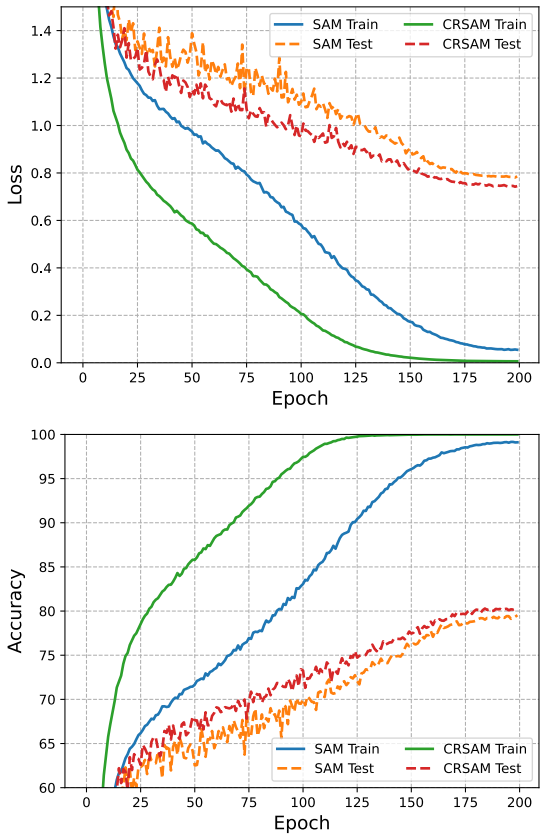



Figure 5: Evolution of training and testing loss/accuracy on CIFAR100 trained with ResNet18 by SAM and our proposed CR-SAM. CR-SAM achieves much faster and stabler convergence, which can be explained by the fact that CR-SAM discourages excessive curvature and thus reduces optimization complexity, making the local minimum easier to reach.

## Details of Experimental Setup

Our experimental setup for training CIFAR10/100 and ImageNet from scratch is detailed in Table 4 and Table 5.

Table 4: Hyperparameters for training from scratch on CIFAR10 and CIFAR100

| | CIFAR-10 | | | CIFAR-100 | | |
|---|---|---|---|---|---|---|
| **ResNet-18** | SGD | SAM | CR-SAM | SGD | SAM | CR-SAM |
| Epoch | | 200 | | | 200 | |
| Batch size | | 128 | | | 128 | |
| Data augmentation | | Basic | | | Basic | |
| Peak learning rate | | 0.05 | | | 0.05 | |
| Learning rate decay | | Cosine | | | Cosine | |
| Weight decay | | $5 \times 10^{-3}$ | | | $5 \times 10^{-3}$ | |
| $\rho$ | - | 0.05 | 0.10 | - | 0.10 | 0.15 |
| $\alpha$ | - | - | 0.1 | - | - | 0.5 |
| $\beta$ | - | - | 0.01 | - | - | 0.01 |
| **ResNet-101** | SGD | SAM | CR-SAM | SGD | SAM | CR-SAM |
| Epoch | | 200 | | | 200 | |
| Batch size | | 128 | | | 128 | |
| Data augmentation | | Basic | | | Basic | |
| Peak learning rate | | 0.05 | | | 0.05 | |
| Learning rate decay | | Cosine | | | Cosine | |
| Weight decay | | $5 \times 10^{-3}$ | | | $5 \times 10^{-3}$ | |
| $\rho$ | - | 0.05 | 0.10 | - | 0.10 | 0.15 |
| $\alpha$ | - | - | 0.2 | - | - | 0.5 |
| $\beta$ | - | - | 0.05 | - | - | 0.05 |
| **Wide-28-10** | SGD | SAM | CR-SAM | SGD | SAM | CR-SAM |
| Epoch | | 200 | | | 200 | |
| Batch size | | 128 | | | 128 | |
| Data augmentation | | Basic | | | Basic | |
| Peak learning rate | | 0.05 | | | 0.05 | |
| Learning rate decay | | Cosine | | | Cosine | |
| Weight decay | | $1 \times 10^{-3}$ | | | $1 \times 10^{-3}$ | |
| $\rho$ | - | 0.10 | 0.10 | - | 0.10 | 0.15 |
| $\alpha$ | - | - | 0.5 | - | - | 0.5 |
| $\beta$ | - | - | 0.1 | - | - | 0.1 |
| **PyramidNet-110** | SGD | SAM | CR-SAM | SGD | SAM | CR-SAM |
| Epoch | | 200 | | | 200 | |
| Batch size | | 128 | | | 128 | |
| Data augmentation | | Basic | | | Basic | |
| Peak learning rate | | 0.05 | | | 0.05 | |
| Learning rate decay | | Cosine | | | Cosine | |
| Weight decay | | $5 \times 10^{-3}$ | | | $5 \times 10^{-3}$ | |
| $\rho$ | - | 0.15 | 0.20 | - | 0.15 | 0.20 |
| $\alpha$ | - | - | 0.5 | - | - | 0.5 |
| $\beta$ | - | - | 0.1 | - | - | 0.1 |

Table 5: Hyperparameters for training on ImageNet from scratch.

| | ResNet-50 | | | ResNet-101 | | |
|---|---|---|---|---|---|---|
| **ImageNet** | SGD | SAM | CR-SAM | SGD | SAM | CR-SAM |
| Epoch | | 90 | | | 90 | |
| Batch size | | 512 | | | 512 | |
| Data augmentation | | Inception-style | | | Inception-style | |
| Peak learning rate | | 1.3 | | | 1.3 | |
| Learning rate decay | | Cosine | | | Cosine | |
| Weight decay | | $3 \times 10^{-5}$ | | | $3 \times 10^{-5}$ | |
| $\rho$ | - | 0.10 | 0.15 | - | 0.10 | 0.15 |
| $\alpha$ | - | - | 0.1 | - | - | 0.2 |
| $\beta$ | - | - | 0.01 | - | - | 0.01 |
| | **ViT-S/32** | | | **ViT-B/32** | | |
| **ImageNet** | SGD | SAM | CR-SAM | SGD | SAM | CR-SAM |
| Epoch | | 300 | | | 300 | |
| Batch size | | 512 | | | 512 | |
| Data augmentation | | Inception-style | | | Inception-style | |
| Peak learning rate | | $3 \times 10^{-3}$ | | | $3 \times 10^{-3}$ | |
| Learning rate decay | | Cosine | | | Cosine | |
| Weight decay | | 0.3 | | | 0.3 | |
| $\rho$ | - | 0.05 | 0.10 | - | 0.05 | 0.10 |
| $\alpha$ | - | - | 0.05 | - | - | 0.05 |
| $\beta$ | - | - | 0.01 | - | - | 0.01 |