SMOOTHNESS-ADAPTIVE SHARPNESS-AWARE MINIMIZATION FOR FINDING FLATTER MINIMA

Hiroki Naganuma*
Mila, Université de Montreal
naganuma.hiroki@mila.quebec

Junhyung Lyle Kim*
Rice University
jlylekim@rice.edu

Anastasios Kyrillidis
Rice University
anastasios@rice.edu

Ioannis Mitliagkas
Mila, Université de Montreal
ioannis@mila.quebec

ABSTRACT

The sharpness-aware minimization (SAM) procedure recently gained increasing attention due to its favorable generalization ability to unseen data. SAM aims to find flatter (local) minima, utilizing a minimax objective. An immediate challenge in the application of SAM is the adjustment of two pivotal step sizes, which significantly influence its effectiveness. We introduce a novel, straightforward approach for adjusting step sizes that adapts to the smoothness of the objective function, thereby reducing the necessity for manual tuning. This method, termed Smoothness-Adaptive SAM (SA-SAM), not only simplifies the optimization process but also promotes the method's inherent tendency to converge towards flatter minima, enhancing performance in specific models.

1 INTRODUCTION

In recent years, there has been a growing expectation for the practical application of machine learning, as evidenced by the exponential increase in publicly available papers categorized under AI and Machine Learning on Arxiv (Krenn et al., 2022). Recent breakthroughs in large language models and generative models require increasingly complex and large models that are pre-trained on large unlabeled datasets. Without powerful computing resources, adapting such large models, set aside training from scratch, can be challenging. Such limitation would also hinder the practitioners from performing a sweep of different hyperparameters (such as the step size or the weight decay parameter), or even to fine-tune a pre-trained model for downstream tasks. This disparity in resources obstructs the democratic development of machine learning methods and infrastructure.

As a result, *what point a training algorithm converges to* in a single training configuration is an important criterion to consider. To that end, there have been many previous studies that investigate the relation between generalization and various metrics. Among them, the sharpness of the local minima has been reported to have an important correlation with generalization (Jiang et al., 2019; Dziugaite et al., 2020). Confirming these empirical studies, recently, Zou et al. (2024) established the theoretical connection between (relative) sharpness and out-of-distribution (OOD) generalization.

The correlation between the sharpness/flatness of the local minima and the ability to generalize has long been observed in several works (Keskar et al., 2016; Foret et al., 2021; Adilova et al., 2023; Cha et al., 2021). The intuition is rather simple: if the loss surface where an algorithm converged to have low sharpness, the flat minima of the deep network should still have low loss when it is slightly perturbed due to train vs. test, or OOD discrepancy. Indeed, recently, Liu et al. (2023) empirically observed a strong correlation between flatness, measured by the trace of Hessian, $\mathrm{Tr}(H)$, and downstream performance among models with the same pre-training loss.

Motivated by such reasoning, the sharpness-aware minimization (SAM) (Foret et al., 2021) recently gained attention due to its ability to generalize well across many different domains (Naganuma &

*equal contributions

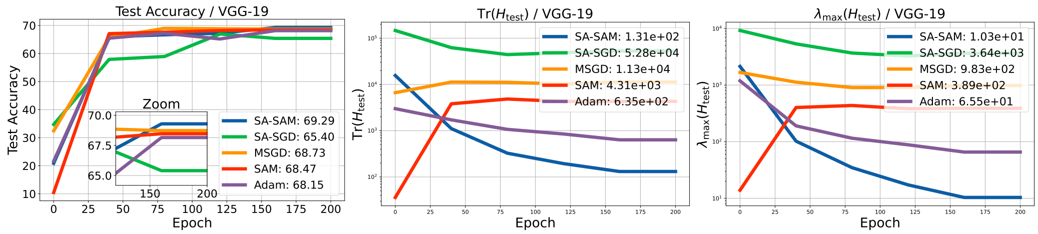

Figure 1: *Test accuracy (OOD generalization) and the curvature information, measured by the trace* $(\mathrm{Tr}(H))$ *and the leading eigenvalue* $(\lambda_{max}(H))$ *of the Hessian, on CIFAR10-C dataset trained with* `VGG-19` *for the considered optimizers.* Our proposed method, SA-SAM, not only achieves the best test accuracy but also converges to flatter minima.

Kimura, 2023; Bahri et al., 2021; Na et al., 2022; Naganuma et al., 2022; Paranjape et al., 2023). In particular, in lieu of simple minimization, SAM formulates the objective as a minimax problem:

$$\min_w L^{\mathrm{SAM}}(w) + \lambda\|w\|_2^2 \quad \text{where} \quad L^{\mathrm{SAM}}(w) := \max_{\|\epsilon\|_p \leqslant \rho} L(w+\epsilon). \tag{1}$$

Here, $\rho \geqslant 0$, which we call the perturbation radius, is a hyperparameter that needs to be tuned. $p \in [1, \infty]$ can be changed, but $p = 2$ is typically used (Foret et al., 2021). Further, in practice, the maximization step is approximated with a single (stochastic) gradient ascent step:

$$\hat{\epsilon}(w) = \rho \frac{\nabla_w L(w)}{\|\nabla_w L(w)\|} \approx \arg \max_{\|\epsilon\|_p \leqslant \rho} L(w+\epsilon),$$

where the gradient can be computed efficiently via: $\nabla_w L^{\mathrm{SAM}} \approx \nabla_w L(w)|_{w+\hat{\epsilon}}$. Finally, the above approximation results in the following SAM update:

$$w_{t+1} = w_t - \eta_t \nabla L\Big(w_t + \rho_t \frac{\nabla L(w_t)}{\|\nabla L(w_t)\|_2}\Big). \tag{2}$$

While the empirical benefit of (2) has been repeatedly observed, there is an immediate challenge to be addressed: how to tune η_t and ρ_t properly? How do they affect the point to which the algorithm converges? What happens when these hyperparameters are not properly tuned?

Addressing the above, we propose the smoothness-adaptive sharpness-aware minimization (SA-SAM). The core of SA-SAM is a simple step size rule in (5) combined with the SAM procedure in (2); see also Algorithm 1. Surprisingly, the smoothness-adaptive step size not only alleviates tuning effort aligning well with the resource-constraint setting but also reinforces the SAM's implicit regularization to flatter minima, as can be seen in Figure 1. Our contributions can be summarized as follows.

- We propose smoothness-adaptive sharpness-aware minimization (SA-SAM), where the step size of SAM in (2) is approximated with the inverse of local smoothness, using (5) as shown in Algorithm 1 and validate its effectiveness shown in Figure 1.
- SA-SAM promotes the SAM's inherent tendency to converge towards flatter minima, shown in Figure 2 and more extensively in Section 3. Moreover, SA-SAM's adaptivity reduces the necessity for manual tuning, as demonstrated in Figure 3.

2 Smoothness-Adaptive SAM and Related Work

In this section, we present SA-SAM, and make connections to related work. We first summarize SA-SAM in Algorithm 1. We denote with $\nabla L(w;\xi)$ the stochastic gradients at w with mini-batch ξ.

Remark 1. *In line 1, the initial choice of η_0 should be provided. We recommend using a small η_0 to prevent divergence in the first iteration in line 2. In the subsequent iterations, the step size η_t is automatically adjusted according to line 5. In line 6, we tune the perturbation radius ρ_t based on Andriushchenko & Flammarion (2022, Theorem 2), which may be improvable. Finally, we use the same mini-batch ξ_t in all stochastic gradient evaluations—using different mini-batches may lead to different dynamics of Algorithm 1.*

Algorithm 1 Smoothness-Adaptive Sharpness-Aware Minimization (SA-SAM)

input: $w_0 \in \mathbb{R}^d$, $\eta_0 > 0, \theta_0 = +\infty$, and ξ_0.
$w_1 = w_0 - \eta_0 \nabla L\big(w_0 + \rho_0 \frac{\nabla L(w_0;\xi_0)}{\|\nabla L(w_0;\xi_0)\|_2}; \xi_0\big)$
for each round $t = 1, \dots$ **do**
  Sample mini-batch ξ_t
  $\eta_t = \min\left\{\frac{\|w_t - w_{t-1}\|}{2\|\nabla L(w_t;\xi_t) - \nabla L(w_{t-1};\xi_t)\|}, \sqrt{1+\theta_t}\eta_{t-1}\right\}$
  $\rho_t = \sqrt{\eta_t}$
  $w_{t+1} = w_t - \eta_t \nabla L\Big(w_t + \rho_t \frac{\nabla L(w_t;\xi_t)}{\|\nabla L(w_t;\xi_t)\|_2}; \xi_t\Big)$
  $\theta_t = \eta_t / \eta_{t-1}$
end for

Local smoothness-adaptive step size. The gradient descent (GD) method is arguably the most fundamental optimization algorithm for minimizing a function $L(w) : \mathbb{R}^d \to \mathbb{R}$. GD iterates with the step size η_t as: $w_{t+1} = w_t - \eta_t \nabla L(w_t)$. Under the assumption that the objective function is *globally* β-smooth, that is:

$$\|\nabla L(x) - \nabla L(y)\| \leqslant \beta \cdot \|x - y\| \quad \forall x, y \in \mathbb{R}^d, \tag{3}$$

the step size $\eta_t = \frac{1}{\beta}$ is the "optimal" step size for GD, which converges for convex L at the rate:

$$L(w_{t+1}) - L(w^\star) \leqslant \frac{\beta\|w_0 - w^\star\|^2}{2(2t+1)}, \tag{4}$$

while $\eta_t \in (0, \frac{2}{\beta})$ defines the "admissible" step size range required to ensure that GD converges for this function class (Polyak, 1963). However, in practice, estimating the (global) smoothness constant β is incredibly challenging.

In Malitsky & Mishchenko (2020), a novel step size rule for GD was introduced, which adapts to the local curvature of the objective function. The proposed step size is simple:

$$\eta_t = \min\left\{\frac{\|w_t - w_{t-1}\|}{2\|\nabla L(w_t) - \nabla L(w_{t-1})\|}, \sqrt{1+\theta_{t-1}}\eta_{t-1}\right\}, \quad \theta_{t-1} = \eta_{t-1}/\eta_{t-2}. \tag{5}$$

The step size in (5) adapts to the *local smoothness* of the iterates, that is:

$$\|\nabla L(w_t) - \nabla L(w_{t-1})\| \leqslant \beta_t \cdot \|w_t - w_{t-1}\|, \quad \forall t = 1, 2, \dots \tag{6}$$

and can *increase during iterations*, while the second condition in (5) ensures η_t does not increase arbitrarily. This step size was extended to the heterogeneous federated learning in Kim et al. (2024).

Connection to the edge of stability and implicit curvature regularization.

As can be seen in (5), the local smoothness constant β_t is at most the global smoothness constant β in (3). As a result, the local smoothness-adaptive step size in (5) is lower bounded by $1/\beta$. Connecting to the concept of the *edge of stability* coined in Cohen et al. (2021), we conjecture the step size in (5) implicitly regularize the sharpness in a non-trivial way.

Moreover, as mentioned in Section 1, SAM in (2) is designed to promote converging to flatter minima. Indeed, in Agarwala & Dauphin (2023), it was demonstrated both empirically and theoretically that SAM enjoys a modified *edge of stability* behavior, which leads to the stabilization of the largest eigenvalues at a lower magnitude.

Combining both insights, SA-SAM in Algorithm 1 seems to enhance the implicit regularization that favors flatter minima. In Section 3, we demonstrate this point empirically.

Related works on SAM. Foret et al. (2021) developed SAM to improve model generalization by concurrently minimizing loss and sensitivity to input shifts. Due to the challenge of explicit calculation of curvature in modern neural networks, SAM approximates this by optimizing (1).

Among works that aim to understand why SAM has high generalization performance, some claim that SAM leads to low-rank feature acquisition (Andriushchenko et al., 2023), while others show that SAM can achieve high generalization even with little correlation between flatness and generalization (Wen et al., 2023). SAM mechanism was also studied for different batch sizes: Wen et al. (2022) subdivided the problem into minimizing $\mathrm{Tr}(H)$ for single batch and $\lambda_{\max}(H)$ for full batch.

In terms of efficiency, Mueller et al. (2023) proposed a method to reduce computational cost while maintaining generalization performance by adapting SAM only to the batch normalization layer. Liu et al. (2022) proposed to compute only the steps of the internal gradient ascent periodically. In training pre-trained models for the downstream task of NLP, Liu et al. (2023) reported that $\mathrm{Tr}(H)$ correlates with the performance of the downstream task. Our proposal may enable the training of pre-trained models that can further improve the performance of the downstream task.

3 EXPERIMENTS

In this section, we perform empirical evaluations of the proposed method. We focus on the generalization performance, both in-distribution, denoted by the *validation* accuracy, and out-of-distribution, denoted by the *test* accuracy. We first recall the experimental details below.

Datasets. We use the CIFAR10-C dataset (Hendrycks & Dietterich, 2019) as a primary dataset for examining OOD generalization. This dataset, featuring CIFAR images corrupted by 19 types of noise, helps assess how well models perform under shifts in data distribution—a key challenge in real-world applications. The noise in CIFAR10-C mimics common disturbances in practice, emphasizing the need for models that maintain accuracy despite these corruptions.

Model architectures. We consider two distinct model architectures, `ViT-Small` and `VGG-19`, applied in many imaging applications.

Optimizers. We consider five different optimizers, including SA-SAM in Algorithm 1, vanilla SAM (Foret et al., 2021), smoothness-adaptive SGD (SA-SGD) (Malitsky & Mishchenko, 2020), momentum SGD (MSGD), and ADAM (Kingma & Ba, 2014). For learning rates (LR), we use a simple grid: $\{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}\}$. Similarly, for the weight decay parameter, we use $\{0, 10^{-8}, 10^{-6}, 10^{-4}\}$. For the other hyperparameters, such as the two momentum values for Adam, we use the default values from Pytorch. For SA-SAM and SAM, we need a way to tune the perturbation radius, ρ_t. Following Andriushchenko & Flammarion (2022, Theorem 2), we set it to be $\rho_t \leftarrow \sqrt{\eta_t}$. For SA-SAM and SA-SGD, we append δ in front of the second condition: $\sqrt{1+\delta\theta_t}\eta_t$ following Malitsky & Mishchenko (2020), and use $\delta = 0.02$ for all experiments.

Curvature measurements. We focus on the eigenvalues of the Hessian matrix of the loss function for the curvature information. Due to the computational complexity, we utilize approximation methods based on PyHessian (Yao et al., 2020). Further details can be found in Appendix A.

3.1 SA-SAM CONVERGES TO FLATTER MINIMA

Figure 1 shows the results of the grid search with LR and weight decay on the OOD dataset (CIFAR10-C) for the hyperparameter configuration with the highest validation accuracy for CIFAR10, trained with `VGG-19`. In this setting, SA-SAM (blue line) not only achieves the best test accuracy (OOD generalization), but also enjoys considerably lower curvature, measured by $\mathrm{Tr}(H)$ and $\lambda_{\max}(H)$.

To investigate further, in Figure 2, we plot the test $\mathrm{Tr}(H)$ for various learning rates[1] and weight decay parameters, for both `VGG-19` and `ViT-Small`. As can be seen, SA-SAM consistently achieves lower curvature for both models, regardless of the hyperparameters. In the extreme case, $\mathrm{Tr}(H)$ of SA-SAM is smaller than that of Adam by 10^{24}. Note that we also provide the $\lambda_{\max}(H)$, as well as the curvature information in validation datasets (in-distribution generalization) in Appendix B.

3.2 EASE OF TUNING

The step size of stochastic gradient-based methods is notoriously hard to tune (Moulines & Bach, 2011; Toulis & Airoldi, 2017; Kim et al., 2022). On top of that, SAM requires the perturbation radius, ρ_t, to be tuned as well. In this regard, a major advantage of SA-SAM is that it significantly alleviates the tuning effort required, as the main step size η_t is computed automatically according to (5). As mentioned before, for the perturbation radius ρ_t, we simply use $\rho_t \leftarrow \sqrt{\eta_t}$ following Andriushchenko & Flammarion (2022, Theorem 2); other choices might improve SA-SAM, but we leave the investigation of the optimal choice of ρ_t for future work.

[1] For SA-SAM and SA-SGD, learning rate (LR) indicates the *initial* step size, η_0.

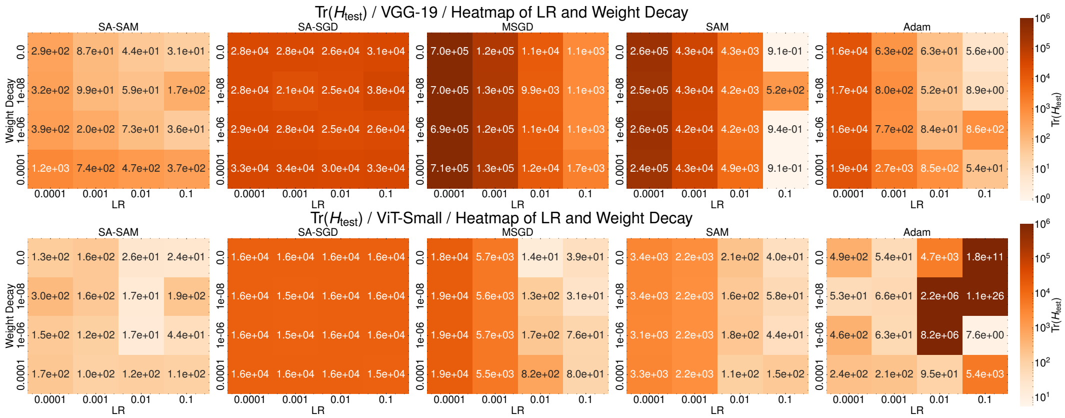

Figure 2: *Heatmap of the test (OOD generalization)* $\mathrm{Tr}(H)$ *for considered optimizers with various learning rates and weight decay parameters, for CIFAR10-C dataset trained with* `VGG-19` *(top) and* `ViT-Small` *(bottom).*

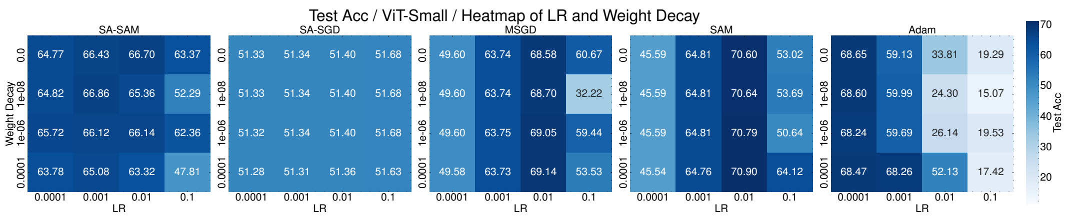

Figure 3: *Heatmap of test accuracy (OOD generalization) for considered optimizers with various learning rates and weight decay parameters, for CIFAR10-C dataset trained with* `ViT-Small`*.*

To demonstrate the ease of tuning, in Figure 3, we plot the test accuracy (OOD generalization) of the considered optimizers for various learning rates and weight decay parameters, for `ViT-Small`. SA-SAM generally exhibits better performance compared to other optimizers. As expected, the performance of MSGD, SAM, and Adam varies significantly with LR, in contrast to SA-SAM and SA-SGD. A caveat for SA-SGD is that the initial learning rate should not be too large; see Remark 1.

Additional results for the evaluation on the validation dataset (in-distribution generalization) as well as results using `VGG-19` can be found in Figure 4 in Appendix B.

4 Conclusion and Future Work

In this work, we proposed SA-SAM, an automatic step size tuner for sharpness-aware minimization. Our proposed method is adaptive to the local smoothness of the objective function. Equipped with the SAM motion, SA-SAM not only alleviates the tuning effort but also reinforces the implicit regularization towards flatter minima. We empirically demonstrated our findings by training `ViT-Small` and `VGG-19` with various optimizers, and investigated the validation and test accuracies, as well as the corresponding curvature information measured by the trace and leading eigenvalue of the Hessian.

While the implicit regularization of SA-SAM towards flatter minima is interesting on its own, further investigation is required on the utilization of the property. To that end, recently, Liu et al. (2023) empirically observed a strong correlation between flatness measured by the trace of Hessian and the downstream task performance, spearheading a concrete case where converging to flatter minima can indeed be helpful. Investigating the performance of SA-SAM in such scenario constitutes important future work of SA-SAM.

SA-SAM also has much potential for improvements. In particular, as mentioned in Remark 1, our choice of the perturbation radius ρ_t might be suboptimal; similarly, different choices of mini-batch sampling scheme may lead to better OOD generalization performance of SA-SAM. Finally, theoretical understandings of SA-SAM in the flavor of Zou et al. (2024) and its connection to the edge of stability and progressive sharpening can be investigated.

REFERENCES

Linara Adilova, Amr Abourayya, Jianning Li, Amin Dada, Henning Petzka, Jan Egger, Jens Kleesiek, and Michael Kamp. Fam: Relative flatness aware minimization. *arXiv preprint arXiv:2307.02337*, 2023.

Atish Agarwala and Yann Dauphin. Sam operates far from home: eigenvalue regularization as a dynamical phenomenon. In *International Conference on Machine Learning*, pp. 152–168. PMLR, 2023.

Maksym Andriushchenko and Nicolas Flammarion. Towards understanding sharpness-aware minimization. In *International Conference on Machine Learning*, pp. 639–668. PMLR, 2022.

Maksym Andriushchenko, Dara Bahri, Hossein Mobahi, and Nicolas Flammarion. Sharpness-aware minimization leads to low-rank features. *arXiv preprint arXiv:2305.16292*, 2023.

Haim Avron and Sivan Toledo. Randomized algorithms for estimating the trace of an implicit symmetric positive semi-definite matrix. *J. ACM*, 58(2), 2011. ISSN 0004-5411. doi: 10.1145/1944345.1944349. URL https://doi.org/10.1145/1944345.1944349.

Dara Bahri, Hossein Mobahi, and Yi Tay. Sharpness-aware minimization improves language model generalization. *arXiv preprint arXiv:2110.08529*, 2021.

Junbum Cha, Sanghyuk Chun, Kyungjae Lee, Han-Cheol Cho, Seunghyun Park, Yunsung Lee, and Sungrae Park. Swad: Domain generalization by seeking flat minima. *Advances in Neural Information Processing Systems*, 34:22405–22418, 2021.

Jeremy M Cohen, Simran Kaur, Yuanzhi Li, J Zico Kolter, and Ameet Talwalkar. Gradient descent on neural networks typically occurs at the edge of stability. *arXiv preprint arXiv:2103.00065*, 2021.

Gintare Karolina Dziugaite, Alexandre Drouin, Brady Neal, Nitarshan Rajkumar, Ethan Caballero, Linbo Wang, Ioannis Mitliagkas, and Daniel M Roy. In search of robust measures of generalization. *Advances in Neural Information Processing Systems*, 33:11723–11733, 2020.

Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In *International Conference on Learning Representations*, 2021.

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019.

Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. *arXiv preprint arXiv:1912.02178*, 2019.

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. *arXiv preprint arXiv:1609.04836*, 2016.

Junhyung Lyle Kim, Panos Toulis, and Anastasios Kyrillidis. Convergence and stability of the stochastic proximal point algorithm with momentum. In *Learning for Dynamics and Control Conference*, pp. 1034–1047. PMLR, 2022.

Junhyung Lyle Kim, Taha Toghani, Cesar A Uribe, and Anastasios Kyrillidis. Adaptive federated learning with auto-tuned clients. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=g0mlwqs8pi.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

Mario Krenn, Lorenzo Buffoni, Bruno Coutinho, Sagi Eppel, Jacob Gates Foster, Andrew Gritsevskiy, Harlin Lee, Yichao Lu, Joao P Moutinho, Nima Sanjabi, et al. Predicting the future of ai with ai: High-quality link prediction in an exponentially growing knowledge network. *arXiv preprint arXiv:2210.00881*, 2022.

Hong Liu, Sang Michael Xie, Zhiyuan Li, and Tengyu Ma. Same pre-training loss, better downstream: Implicit bias matters for language models. In *International Conference on Machine Learning*, pp. 22188–22214. PMLR, 2023.

Yong Liu, Siqi Mai, Xiangning Chen, Cho-Jui Hsieh, and Yang You. Towards efficient and scalable sharpness-aware minimization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12360–12370, 2022.

Yura Malitsky and Konstantin Mishchenko. Adaptive gradient descent without descent. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 6702–6712. PMLR, 2020.

Eric Moulines and Francis Bach. Non-asymptotic analysis of stochastic approximation algorithms for machine learning. *Advances in neural information processing systems*, 24, 2011.

Maximilian Mueller, Tiffany Vlaar, David Rolnick, and Matthias Hein. Normalization layers are all that sharpness-aware minimization needs. *arXiv preprint arXiv:2306.04226*, 2023.

Clara Na, Sanket Vaibhav Mehta, and Emma Strubell. Train flat, then compress: Sharpness-aware minimization learns more compressible models. *arXiv preprint arXiv:2205.12694*, 2022.

Hiroki Naganuma and Masanari Kimura. Necessary and sufficient hypothesis of curvature: Understanding connection between out-of-distribution generalization and calibration. *ICLR2023 workshop on Domain Generalization*, 2023.

Hiroki Naganuma, Kartik Ahuja, Shiro Takagi, Tetsuya Motokawa, Rio Yokota, Kohta Ishikawa, Ikuro Sato, and Ioannis Mitliagkas. Empirical study on optimizer selection for out-of-distribution generalization. *arXiv preprint arXiv:2211.08583*, 2022.

Jay N Paranjape, Nithin Gopalakrishnan Nair, Shameema Sikder, S Swaroop Vedula, and Vishal M Patel. Adaptivesam: Towards efficient tuning of sam for surgical scene segmentation. *arXiv preprint arXiv:2308.03726*, 2023.

Boris Teodorovich Polyak. Gradient methods for the minimisation of functionals. *USSR Computational Mathematics and Mathematical Physics*, 3(4):864–878, 1963.

Panos Toulis and Edoardo M. Airoldi. Asymptotic and finite-sample properties of estimators based on stochastic gradients. *The Annals of Statistics*, 45(4):1694 – 1727, 2017. doi: 10.1214/16-AOS1506.

Kaiyue Wen, Tengyu Ma, and Zhiyuan Li. How does sharpness-aware minimization minimize sharpness? *arXiv preprint arXiv:2211.05729*, 2022.

Kaiyue Wen, Tengyu Ma, and Zhiyuan Li. Sharpness minimization algorithms do not only minimize sharpness to achieve better generalization. *arXiv preprint arXiv:2307.11007*, 2023.

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

Yingtian Zou, Kenji Kawaguchi, Yingnan Liu, Jiashuo Liu, Mong-Li Lee, and Wynne Hsu. Towards robust out-of-distribution generalization bounds via sharpness. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=tPEwSYPtAC.

APPENDIX

A DETAILS OF THE EXPERIMENTAL SETUP

Hardware Configurations: In our experiments, we used an anonymous cloud (to appear in the camera-ready version). The detailed machine specifications are as follows:

- Memory: 64 GB
- CPU: `AMD EPYC 7502` x 40
- GPU: `Quadro RTX 8000` x 1

Software and Library Configurations: The following is the software and library environment we used.

- Python: 3.7.6
- PyTorch: 1.8.1
- CUDA: 11.1
- CUDNN: 8005

Hessian Calculation: In our experimental setup, explicitly calculating the Hessian presents a challenge due to memory size limitations, particularly in practical problem settings. Consequently, we opted to use approximation calculations. More specifically, we utilized the calculation methods for $\mathrm{Tr}(H)$ and $\lambda_{\text{max}}(H)$ provided in PyHessian[2] (Yao et al., 2020).

The Hutchinson method, a widely-used algorithm known for approximating the trace of the Hessian matrix, was employed (Avron & Toledo, 2011). The Hutchinson method approximates the expected value of the quadratic form of the Hessian matrix and a Rademacher random vector—where each element takes the value of either 1 or -1 with equal probability of $\frac{1}{2}$.

$$\begin{aligned}\mathrm{Tr}(H) &= \mathrm{Tr}(HI) = \mathrm{Tr}(H\mathbb{E}[vv^T]) \\ &= \mathbb{E}[\mathrm{Tr}(Hvv^T)] = \mathbb{E}[v^T Hv]\end{aligned} \tag{7}$$

The computation of $\lambda_{\text{max}}(H)$ was computed using the power iteration method. This computational approach ensures a balance between resource efficiency and computation accuracy, enabling the practical estimation of critical metrics within our experimental context.

B ADDITIONAL EXPERIMENTAL RESULTS

Figure 4 shows the validation (i.e., in-distribution generalization) performance and curvature information trained on `VGG-19`.

The overall results of part of the results presented in Section 3 are shown in Figure 5 .

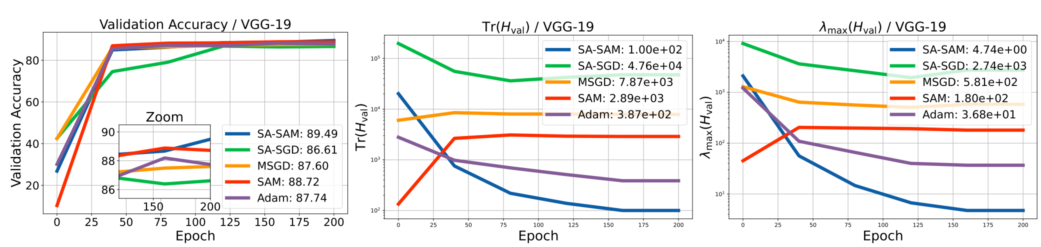

Figure 4: Validation Accuracy, $\mathrm{Tr}(H)$ and $\lambda_{\text{max}}(H)$ / CIFAR10-C on `VGG-19`.

[2] https://github.com/amirgholami/PyHessian

Figure 5: Accuracy, $\mathrm{Tr}(H)$ and $\lambda_{\max(H)}$ / CIFAR10-C on ViT-Small for both validation (in-distribution) and test (OOD generalization).