Abstract

Energy-efficient deep neural network (DNN) accelerators are prone to non-idealities that degrade DNN performance at inference time. To mitigate such degradation, existing methods typically add perturbations to the DNN weights during training to simulate inference on noisy hardware. However, this often requires knowledge about the target hardware and leads to a trade-off between DNN performance and robustness, decreasing the former to increase the latter. In this work, we show that applying sharpness-aware training by optimizing for both the loss value and loss sharpness significantly improves robustness to noisy hardware at inference time while also increasing DNN performance. We further motivate our results by showing a high correlation between loss sharpness and model robustness. We show superior performance compared to injecting noise during training and aggressive weight clipping on multiple architectures, optimizers, datasets, and training regimes without relying on any assumptions about the target hardware. This is observed on a generic noise model as well as on accurate noise simulations from real hardware.

1. Introduction

The success of deep neural networks (DNNs) has also been accompanied by an increase in training complexity and computational demands, prompting efficient DNN designs [35–37]. However, with the slowing down of Moore’s law and the ending of Dennard scaling, power consumption is now the key design constraint for DNN accelerators, which calls for new hardware and algorithms. Emerging approaches, such as in-memory computing, are promising directions to improve the energy consumption and throughput of existing DNNs [54]. This is particularly important in computer vision applications that have low-energy or high-throughput requirements.

Despite their appeal, highly efficient hardware implementations are often prone to variabilities [57] which perturb the DNN weights and lead to a degradation of DNN performance [19, 20, 50, 55]. The main approach for improving the robustness of DNNs has been to apply weight perturbations during training [4, 12, 15, 19]. However, such approaches typically rely on noise simulations from the target hardware to which the DNN will be deployed. Moreover, existing robustness methods provide a trade-off between DNN performance and DNN robustness, decreasing the former to increase the latter.

The goal of this work is to increase both model robustness and performance without relying on any noise simulations from the target hardware. By doing so, we do not compromise the applicability of our approach, neither by reducing the original DNN performance nor by tailoring it to a specific hardware design. To achieve this, we propose to apply sharpness-aware minimization methods during training to promote accurate DNN inference after deployment on noisy, yet energy-efficient, hardware.

The benefit of converging to a smoother loss landscape has been primarily tied to improving generalization performance [5, 10, 16, 17, 21, 42]. With this goal in mind, Foret et al. [11] recently proposed sharpness-aware minimization (SAM) by minimizing both the loss value and loss sharpness within a maximization region around each parameter during training. By showing a high correlation between loss sharpness and test performance, SAM has ignited several follow-up works since its proposal. Particularly, adaptive sharpness-aware minimization (ASAM) [25], which reformulated sharpness to be invariant to weight scaling and to enable convergence over larger neighborhood regions.

In this work, we empirically show that sharpness-aware training directly contributes to a more accurate inference on noisy hardware, increasing both model robustness and performance due to convergence to flatter minima. Our in-depth studies also show a high correlation between loss sharpness and model robustness. Moreover, we show that sharpness-aware training may be used in combination with existing robustness methods to improve their DNN performance and robustness trade-off. This is showcased by
our extensive experiments on several DNN architectures (ResNet-18, ResNet-32, VGG-13, and MobileNetV2), optimizers (SGD and Adam), datasets (CIFAR-10, CIFAR-100, and ImageNet), and learning regimes (training from scratch and finetuning). Finally, on top of a generic noise model, we test the different approaches on accurate noise simulations from real hardware, showing the benefits of sharpness-aware training in practical settings.

2. Related work

The deployment of pre-trained models on noisy hardware for highly efficient inference is known to introduce non-idealities. This is caused by noise inherent to the device [56] such as programming noise after weight transfer to the target hardware and read noise every time the programmed weights are accessed. Without robustness measures, such hardware noise significantly hinders the performance of neural networks. To promote robustness after deployment in noisy hardware at inference time, existing methods typically inject noise or faults to DNN weights during training [1,2,27,31,50]. In particular, adding weight noise [19] and promoting redundancy by performing aggressive weight clipping [51] have been shown to be effective methods for increasing DNN robustness. However, existing robustness methods often lead to a decrease of DNN performance for promoting robustness. Moreover, they typically rely on noise measurements from the target hardware to improve the performance and robustness trade-off.

Sharpness-aware training has recently gathered increased interest [6, 11, 18, 53]. Particularly, SAM has sparked a lot of new follow-up works due to the significant increase in generalization performance presented in the original paper. Variants mainly focus on increasing the efficiency [8, 9, 30, 58, 59], performance [22, 25, 60], or understanding [3] of sharpness-aware training. Efforts have also been made to extend SAM to specific use-cases such as quantization-aware training [29] or data imbalance settings [28]. Several works [7, 25] have also highlighted the importance of scale-invariant sharpness measures, including in the context of model robustness against adversarial examples [52].

In a similar vein to our work, Sun et al. [53] recently related the sharpness of the loss landscape with robustness to adversarial noise perturbations. This was further observed by Kim et al. [22]. We follow this under-explored research direction and provide an in-depth study on the effect of loss sharpness in robustness against noisy hardware. Stutz et al. [52] also recently studied the flatness of the (robust) loss landscape on the basis of adversarial training with perturbed examples [32]. In particular, they tackle the problem of robust overfitting [14], i.e. having high robustness to adversarial examples seen during training but generalizing poorly to new adversarial examples at test time, through the lens of flat minima. Here, we focus on the problem of improving robustness against noisy weights (rather than adversarial examples) at inference time.

3. Sharpness-aware minimization (SAM)

The goal of sharpness-aware minimization or SAM is to promote a smoother loss landscape by optimizing for both the loss value and loss sharpness during training. Generally speaking, given a parameter $w$, the goal is to find a region in the loss landscape where not only does $w$ have a low training loss $L$ but also do its neighbor points. Considering the $L_2$ norm and discarding the regularization term in the original algorithm for simplicity, SAM uses the following objective:

$$\min_w \max_{\|\epsilon\|_2 \leq \rho} L(w + \epsilon),$$

where the size of the neighborhood region is defined by a sphere with radius $\rho$ and the optimal $\epsilon$ may be efficiently estimated via a first-order approximation.

By building on the strong correlation between sharpness and generalization performance, SAM is generally used in practice to achieve better test performance. However, there are two main drawbacks. The first is that, despite its efficiency in estimating the worst-case weight perturbations, SAM’s objective requires two forward passes for every backward pass. To mitigate this additional time complexity, the authors propose to leverage distributed training. Another drawback of SAM is that the sharpness calculation is not independent from weight scaling. This allows the manipulation of sharpness values by applying scaling operators to the weights such that weight values change without altering the model’s final prediction [7, 52].

3.1. Adaptive sharpness-aware minimization (ASAM)

To tackle the scale variance issue, adaptive sharpness-aware minimization or ASAM was proposed by Kwon et al. [25]. By taking into account scaling operators that do not change the model’s loss, ASAM creates a new notion of adaptive sharpness that is invariant to parameter scaling, contrarily to SAM. This is reflected in ASAM’s objective:

$$\min_w \max_{\|T_w^{-1}\epsilon\|_2 \leq \rho} L(w + \epsilon),$$

where $T_w^{-1}$ is a normalization operator of $w$ representing a family of linear operators that do not change $L$. This also leads to a change in the neighborhood shape, which is now ellipsoidal instead of spherical.

In practice, the adaptive sharpness introduced by ASAM shows a higher correlation with generalization performance and improved convergence using larger neighborhood sizes, i.e. larger maximization regions. This is particularly important in the scope of this work since our experimental results
showcase that a higher $\rho$ generally leads to better performance retention and has a higher correlation to performance loss in noisy regimes than with smaller neighborhood sizes.

4. Model robustness on a generic noise model

As our use-case, we consider memristor-based DNN implementations [19, 20], where the weights of all fully-connected or convolutional layers of a pre-trained DNN are linearly mapped to the range of possible conductance values from 0 to $G_{\text{max}}$. More concretely, the ideal conductance values $G_{T,ij}^l$ for the weights $W_{ij}^l$ of layer $l$ is

$$G_{T,ij}^l = W_{ij}^l \times \frac{G_{\text{max}}}{W_{\text{max}}}$$

(3)

where $W_{\text{max}}$ is layer $l$’s maximum absolute weight. However, as pointed out previously, $G_{T,ij}^l$ is not achievable in practice since conductance errors $\delta_{ij}$ are originated from programming and read noise [56] as well as conductance drift over time [1]. Hence, in the general case, the non-ideal conductance values $G_{ij}^l$ may be defined as

$$G_{ij}^l = G_{T,ij}^l \times \delta_{ij}$$

(4)

with $\delta_{ij} \sim N(1, \sigma_c^2)$. Following Joshi et al. [19], $\sigma_c$ represents the conductance variation of $G_{ij}^l$ relative to $G_{T,ij}^l$.

4.1. Experimental details

We use the generic noise model presented in Eq. (4), which may be used to accurately estimate inference accuracy in noise models derived from measurements of existing noisy hardware implementations. We note that $\sigma_c = 0.0$ refers to the special case where no noise is applied to the DNN weights. We first performed experiments using ResNet-32 [13], MobileNetV2 [47], and VGG-13 [49] trained on CIFAR-10 and CIFAR-100 [24]. For a direct method comparison, we matched the ResNet-32 architecture used by Joshi et al. [19] with a reduced number of channels. We also adopted their training procedure, training for 200 epochs with a batch size of 128 and starting with a learning rate of 0.1, divided by 10 every 50 epochs.

To further expand our exploration of different models and datasets, we finetuned a ResNet-18 model pre-trained on ImageNet [46] provided by PyTorch for the different methods. More specifically, we followed Joshi et al. [19]’s finetuning procedure and first initialized our models with the weights of the aforementioned ResNet-18 model. Then, we finetuned for 10 epochs with a batch size of 400, a learning rate of 0.001, and a weight decay of 0.0001.

For the models trained with ASAM, we used neighborhood sizes of $\rho \in \{0.5, 1.0, 1.5, 2.0\}$, whereas the SAM models were trained with smaller neighborhood sizes ($\rho \in \{0.05, 0.1, 0.2, 0.5\}$) to prevent performance loss. To promote a cleaner visualization, we only report the neighborhood size with the best performance; we found that $\rho = 0.5$ and $\rho = 1.0$ often work the best for ASAM or $\rho = 0.05$ and $\rho = 0.1$ for SAM. Please refer to the appendix for additional details.

4.1.1 Adaptive batch normalization statistics (AdaBS)

On top of a simple baseline trained with vanilla SGD, we experimented with two methods: the additive noise approach proposed by Joshi et al. [19] and aggressive weight clipping [51]. More specifically, the first method applies additive Gaussian noise to DNN weights, whereas the second method clips the DNN weights into a small range of possible values. The models are trained from scratch and use the training settings previously described.

The additive random noise proposed by Joshi et al. [19] is sampled from a Gaussian distribution $\mathcal{N}(0, \sigma_n^2)$, where

$$\sigma_n = \frac{W_{\text{max}}^l \times \sigma_{G}^l}{G_{\text{max}}}$$

(5)

and $\sigma_G$ represents the standard deviation of hardware non-idealities observed in practice. Both $\sigma_G$ and $G_{\text{max}}$ are device characteristics that are set to 0.94 and 25, respectively, following the empirical measurements on 1 million of phase-change memory devices [19].

Since the amount of added noise is proportional to the maximum absolute weight value of a given layer, we perform weight clipping after each weight update; we used the range $[-\alpha \times \sigma_{W_l}, \alpha \times \sigma_{W_l}]$, where $\sigma_{W_l}$ is the standard deviation of the weights of layer $l$ and $\alpha$ is a predefined hyper-parameter defaulted to 2.0. We tried a different range of $\alpha \in \{1.5, 2.0, 2.5\}$, but the best performance for all CIFAR10/100 models was achieved with the default $\alpha$ value of 2.0. For finetuning on ImageNet, we used $\alpha = 2.5$, as the original authors suggested [19].

For aggressive weight clipping, we tried the values for the clipping range $c$, as performed by the original authors.
Figure 1. Performance of the different methods under a range of random conductance variations. We plot the mean and standard deviation over 10 inference runs.

We also observe that combining aggressive weight clipping with ASAM compared to the rest of the methods, particularly on the architectures that achieve higher accuracy, i.e., MobileNetV2 and VGG-13. This is specifically important since these architectures are the ones that will likely be deployed in practice.

We also show the performance retention of the different methods when trained with different optimizers. In particular, we used the Adam optimizer [23] to train a ResNet-32 model on CIFAR-10 for the different methods using the same training setup as SGD but using the default hyper-parameter configuration for Adam provided by PyTorch [44]. Results are shown in Fig. 2. We notice a similar trend and clearly observe an increase in model robustness with ASAM compared to the rest of the methods, particu-
this is due to the intricacies of finetuning a large model on ImageNet. Nevertheless, we see that there is always either a SAM or ASAM variant that achieves the best robustness at every tested noise level. Moreover, the models finetuned with only SAM or ASAM always reach a better performance than the models trained with only SGD or SGD and weight clipping across all tested noise levels, including the non-noisy setting. We note that despite the good robustness performance when training with additive noise, in particular with SGD, all models show a drop in DNN performance at $\sigma_c = 0.0$. Moreover, we see that all SAM and ASAM variants reach similar or higher robustness than the model finetuned with SGD and additive noise at the highest conductance tested ($\sigma_c = 0.4$).

### 4.4. Correlation between sharpness and robustness

For measuring sharpness, we use the $m$-sharpness metric proposed by Foret et al. [11], which stems from the original SAM formulation (Eq. (1)), and further extend it to ASAM’s objective (Eq. (2)). Considering a training set ($S_{\text{train}}$) composed of $n$ minibatches $S$ of size $m$, we compute the difference of the loss $l_s$ of a given sample $s$ with and without a worst-case perturbation $\epsilon$ on $w$. For SAM, $m$-sharpness is calculated as

$$\frac{1}{n} \sum_{S \in S_{\text{train}}} \max_{\|\epsilon\| \leq \rho} \frac{1}{m} \sum_{s \in S} l_s(w + \epsilon) - l_s(w),$$

whereas for ASAM, $m$-sharpness is obtained by

$$\frac{1}{n} \sum_{S \in S_{\text{train}}} \max_{\|T_w^{-1}\epsilon\| \leq \rho} \frac{1}{m} \sum_{s \in S} l_s(w + \epsilon) - l_s(w).$$
In our experiments, we used $m = 400$ and $m = 128$ for measuring the sharpness of models finetuned on ImageNet and trained on CIFAR-10/100, respectively. We additionally computed the metric proposed by Keskar et al. [21] which is computed based on the largest value of a loss function around a cuboid neighborhood of a solution. We used a neighborhood size of $1e^{-3}$, as the authors suggested, using the full parameter space without random projections.

To relate sharpness and robustness, we treat robustness as the performance gap measured by the difference in test accuracy between the noiseless models, i.e., with no conductance variation applied to the weights ($\sigma_c = 0.0$), and different noisy model configurations, i.e., with different conductance variations ($\sigma_c \in 0.1, 0.2, 0.3, 0.4$). We measured the Pearson correlation between sharpness and the aforementioned performance gap over different neighborhood sizes ($\rho$) for both SAM and ASAM. Results are shown in Fig. 4.

We observe that ASAM’s $m$-sharpness shows the highest Pearson correlation, particularly at large neighborhood sizes, followed by SAM. We also see that higher conductance variations lead to a slight increase in ASAM’s best neighborhood size, i.e., the $\rho$ with the best-observed correlation. Interestingly, both ASAM and SAM reach their best correlation with the performance gap between the noiseless model ($\sigma_c = 0.0$) and the noisy model realization obtained at $\sigma_c = 0.3$. In contrast, Keskar et al. [21]'s sharpness reaches the highest correlation at the lowest noisy settings ($\sigma_c = 0.1$). The high correlation values of SAM and ASAM on the noisiest regimes suggest that the maximization regions used by sharpness-aware training to minimize loss sharpness are a good proxy for increasing robustness in noisy hardware settings.

We note that ASAM’s $m$-sharpness reaches a higher correlation than SAM’s $m$-sharpness across all neighborhood sizes on each tested conductance variation level. We provide a visualization of the best correlation between sharpness and performance gap on $\sigma_c = 0.0$ and $\sigma_c = 0.4$ of each method in the appendix. In practice, both SAM and ASAM are constrained by the size of the neighborhood region before performance loss since there is a trade-off be-
Figure 6. Correlation between ASAM’s $m$-sharpness (Eq. (7), $\rho = 0.5$) and the performance gap of the different methods on ResNet-18 finetuned on ImageNet. We plot the mean and standard deviation over 3 inference runs.

Figure 7. Correlation between ASAM’s $m$-sharpness (Eq. (7), $\rho = 0.5$) and the performance gap of the different methods on ResNet-32 trained on CIFAR-10 using the Adam optimizer. We plot the mean and standard deviation over 10 inference runs.

between minimizing loss sharpness over loss value. However, ASAM shows a higher correlation in the practical neighborhood size ranges, as observed in the smaller neighborhood sizes in Fig. 4, especially considering its ability to converge over larger maximization regions when compared to SAM. We believe that this may be one of the reasons why we observe better robustness in the ASAM variants as compared to the SAM variants in previous experiments.

We present the relation between sharpness and robustness of all the tested models using ASAM’s $m$-sharpness with the default neighborhood size ($\rho = 0.5$) in Figs. 5, 6, and 7. We observe a strong correlation across all architectures, optimizers, and datasets showcasing the ability of ASAM’s $m$-sharpness in acting as a generic robustness metric. As expected, we see that the ASAM variants have lower sharpness (and higher robustness) followed by the SAM and SGD variants. We also note that aggressive weight clipping promotes flatter minima, which was also previously noticed [52] and might explain the success of the method in increasing model robustness. Finally, we observe that the VGG-13 architecture in Fig. 5 achieves the highest robustness and also has the lowest sharpness compared to the rest of the architectures. We provide visualizations for the other sharpness metrics in the appendix.

5. Model robustness on noise simulations from real hardware

Since the generic noise model used in Section 4 may not exactly match existing hardware implementations, we also performed experiments using an inference simulator on real hardware provided by IBM’s analog hardware acceleration kit [45]. This simulator uses the empirical measurements from 1 million phase-change memory devices [41] to accurately simulate how hardware noise affects the DNN weights [19]. Particularly, by taking into account the programming and read noise, we can simulate the DNN weights of a given model after deployment, i.e., after weight transfer to the target hardware. We refer to the library’s documentation\footnote{\url{https://aihwkit.readthedocs.io/en/latest/pcm_inference.html}} for additional details.

5.1. Performance after weight transfer

We first tested the performance of the different methods on ResNet-32 and VGG-13 after training on CIFAR-10 and after weight transfer to the target hardware. We note that we could not run simulations for MobileNetV2 since grouped convolutions are not supported by the inference simulator library at the time of this writing. Results are shown in Fig. 8.

We observe that ASAM achieves both the best performance after training and after weight transfer on both architectures. On the other hand, training with noise or aggressive weight clipping provides a trade-off between performance after training and after deployment. Specifically, on ResNet-32, the performance of both methods after training decreases when compared to SGD but robustness increases. However, this is not observed on VGG-13, where aggressive weight clipping fails to promote either better performance or better robustness relative to standard SGD. Lastly, SAM promotes higher robustness than weight clipping on
## 6. Conclusion

In this work, we proposed to leverage adaptive sharpness-aware training for improving the inference performance of DNNs in noisy hardware implementations. Our results on different architectures, optimizers, datasets, and training regimes showcase the benefits of training for lower sharpness to increase both DNN performance and DNN robustness. In contrast, existing robustness methods typically achieve a trade-off between the two, decreasing performance to increase robustness. Our studies on the correlation between SAM and ASAM’s $m$-sharpness and model robustness pave the way to further improving sharpness-aware training algorithms to promote higher model robustness. We highlight that establishing a better trade-off between loss value and loss sharpness minimization of existing algorithms, particularly ASAM, is critical for improving DNNs robustness on noisy hardware.

Studying how additional SAM variants [22] help retain inference performance on noisy hardware and establishing strong theoretical foundations are important future directions. The insights of our work may be used to develop new sharpness-aware algorithms, e.g. by focusing on increasing the neighborhood region without sacrificing performance. Extending our studies to image [38,39] and text [48] generation by measuring the impact of hardware faults on sample quality and diversity [33,34,40,43] is also worth exploring.

### 8. Conclusion

In this work, we proposed to leverage adaptive sharpness-aware training for improving the inference performance of DNNs in noisy hardware implementations. Our results on different architectures, optimizers, datasets, and training regimes showcase the benefits of training for lower sharpness to increase both DNN performance and DNN robustness. In contrast, existing robustness methods typically achieve a trade-off between the two, decreasing performance to increase robustness. Our studies on the correlation between SAM and ASAM’s $m$-sharpness and model robustness pave the way to further improving sharpness-aware training algorithms to promote higher model robustness. We highlight that establishing a better trade-off between loss value and loss sharpness minimization of existing algorithms, particularly ASAM, is critical for improving DNNs robustness on noisy hardware.

Studying how additional SAM variants [22] help retain inference performance on noisy hardware and establishing strong theoretical foundations are important future directions. The insights of our work may be used to develop new sharpness-aware algorithms, e.g. by focusing on increasing the neighborhood region without sacrificing performance. Extending our studies to image [38,39] and text [48] generation by measuring the impact of hardware faults on sample quality and diversity [33,34,40,43] is also worth exploring.
Acknowledgments

This work was supported by an IVADO grant [PRF-2019-4784991664] and by the Natural Sciences and Engineering Research Council of Canada (NSERC) [RGPIN-2021-04242]. Sarath Chandar is supported by a Canada CIFAR AI Chair and an NSERC Discovery Grant. The authors acknowledge the material support of the Digital Research Alliance of Canada and NVIDIA in the form of computational resources.

References

[1] S. Ambrogio, M. Gallot, K. Spoon, H. Tsai, C. Mackin, M. Wesson, S. Kariyappa, P. Narayanan, C.-C. Liu, A. Kumar, A. Chen, and G. W. Burr. Reducing the impact of phase-change memory conductance drift on the inference of large-scale hardware neural networks. In International Electron Devices Meeting, 2019. 2, 3

[2] Stefano Ambrogio, Pritish Narayanan, Hsinyu Tsai, Robert M Shelby, Irem Boybat, Carmelo Di Nolfo, Severin Sidler, Massimo Giordano, Martina Bodini, Nathan CP Farinha, et al. Equivalent-accuracy accelerated neural-network training using analogue memory. Nature, 2018. 2

[3] Maksym Andriushchenko and Nicolas Flammarion. Towards understanding sharpness-aware minimization. In International Conference on Machine Learning, 2022. 2

[4] H.-Y. Chang, P. Narayanan, S. C. Lewis, N. C. P. Farinha, K. Hosokawa, C. Mackin, H. Tsai, S. Ambrogio, A. Chen, and G. W. Burr. AI hardware acceleration with analog memory: Microarchitectures for low energy at high speed. IBM Journal of Research and Development, 2019. 1

[5] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-SGD: Biscing gradient descent into wide valleys. In International Conference on Learning Representations, 2017. 1

[6] Xiangning Chen, Cho-Jui Hsieh, and Boqing Gong. When vision Transformers outperform ResNets without pre-training or strong data augmentations. In International Conference on Learning Representations, 2022. 2

[7] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In International Conference on Machine Learning, 2017. 2

[8] 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, 2022. 2

[9] Jiawei Du, Daquan Zhou, Jiashi Feng, Vincent YF Tan, and Joey Tianyi Zhou. Sharpness-aware training for free. arXiv preprint arXiv:2205.14083, 2022. 2

[10] Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In Conference on Uncertainty in Artificial Intelligence, 2017. 1

[11] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In International Conference on Learning Representations, 2021. 1, 2, 5

[12] Tayfun Gokmen, Malte J. Rasch, and Wilfried Haensch. The marriage of training and inference for scaled deep learning analog hardware. In International Electron Devices Meeting, 2019. 1

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Conference on Computer Vision and Pattern Recognition, 2016. 3

[14] Warren He, James Wei, Xinyun Chen, Nicholas Carlini, and Dawn Song. Adversarial example defense: Ensembles of weak defenses are not strong. In USENIX Workshop on Offensive Technologies, 2017. 2

[15] Sébastien Henwood, François Leduc-Primeau, and Yvon Savaria. Layerwise noise maximisation to train low-energy deep neural networks. In International Conference on Artificial Intelligence Circuits and Systems, 2020. 1

[16] Sepp Hochreiter and Jürgen Schmidhuber. Simplifying neural nets by discovering flat minima. Advances in Neural Information Processing Systems, 1994. 1

[17] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In Conference on Uncertainty in Artificial Intelligence, 2018. 1

[18] 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. 2

[19] Vinay Joshi, Manuel Le Gallo, Simon Haefeli, Irem Boybat, Sasidharan Rajalekshmi Nandakumar, Christophe Piveteau, Martino Dazzi, Bipin Rajendran, Abu Sebastian, and Evangelos Eleftheriou. Accurate deep neural network inference using computational phase-change memory. Nature Communications, 2020. 1, 2, 3, 7

[20] Jonathan Kern, Sébastien Henwood, Gonçalo Mordido, Elsa Dupraz, Abdeldjalil Aïssa-El-Bey, Yvon Savaria, and François Leduc-Primeau. MemSE: Fast MSE prediction for noisy memristor-based DNN accelerators. In International Conference on Artificial Intelligence Circuits and Systems, 2022. 1, 3

[21] 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, 2016. 1, 6, 12, 13, 15, 16

[22] 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. 2, 8

[23] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014. 4

[24] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images., 2009. 3

[25] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In International Conference on Machine Learning, 2017. 2

[26] 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, 2022. 2

[27] Gintare Karolina Dziugaite and Daniel M. Roy. Computing nonvacuous generalization bounds for deep (stochastic) neural networks with many more parameters than training data. In Conference on Uncertainty in Artificial Intelligence, 2017. 1

[28] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. In International Conference on Learning Representations, 2021. 1, 2, 5

[29] Tayfun Gokmen, Malte J. Rasch, and Wilfried Haensch. The marriage of training and inference for scaled deep learning analog hardware. In International Electron Devices Meeting, 2019. 1

[30] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In Conference on Computer Vision and Pattern Recognition, 2016. 3

[31] Warren He, James Wei, Xinyun Chen, Nicholas Carlini, and Dawn Song. Adversarial example defense: Ensembles of weak defenses are not strong. In USENIX Workshop on Offensive Technologies, 2017. 2

[32] Sébastien Henwood, François Leduc-Primeau, and Yvon Savaria. Layerwise noise maximisation to train low-energy deep neural networks. In International Conference on Artificial Intelligence Circuits and Systems, 2020. 1

[33] Sepp Hochreiter and Jürgen Schmidhuber. Simplifying neural nets by discovering flat minima. Advances in Neural Information Processing Systems, 1994. 1

[34] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In Conference on Uncertainty in Artificial Intelligence, 2018. 1

[35] 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. 2

[36] Vinay Joshi, Manuel Le Gallo, Simon Haefeli, Irem Boybat, Sasidharan Rajalekshmi Nandakumar, Christophe Piveteau, Martino Dazzi, Bipin Rajendran, Abu Sebastian, and Evangelos Eleftheriou. Accurate deep neural network inference using computational phase-change memory. Nature Communications, 2020. 1, 2, 3, 7

[37] Jonathan Kern, Sébastien Henwood, Gonçalo Mordido, Elsa Dupraz, Abdeldjalil Aïssa-El-Bey, Yvon Savaria, and François Leduc-Primeau. MemSE: Fast MSE prediction for noisy memristor-based DNN accelerators. In International Conference on Artificial Intelligence Circuits and Systems, 2022. 1, 3

[38] 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, 2016. 1, 6, 12, 13, 15, 16

[39] 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. 2, 8

[40] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014. 4

[41] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images., 2009. 3
[25] 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. 1, 2

[26] Manuel Le Gallo, Abu Sebastian, Giovanni Cherubini, Heiner Giefer, and Evangelos Eleftheriou. Compressed sensing with approximate message passing using in-memory computing. Transactions on Electron Devices, 2018. 3

[27] Can Li, Zhongrui Wang, Mingyi Rao, Daniel Belkin, Wennhao Song, Hao Jiang, Peng Yan, Yunning Li, Peng Lin, Miao Hu, et al. Long short-term memory networks in memristor crossbar arrays. Nature Machine Intelligence, 2019. 2

[28] Hong Liu, Jeff Z. HaoChen, Adrien Gaidon, and Tengyu Ma. Self-supervised learning is more robust to dataset imbalance. In NeurIPS 2021 Workshop on Distribution Shifts: Connecting Methods and Applications, 2021. 2

[29] Jing Liu, Jianfei Cai, and Bohan Zhuang. Sharpness-aware quantization for deep neural networks. arXiv preprint arXiv:2111.12273, 2021. 2

[30] Yong Liu, Siqi Mai, Xiangning Chen, Cho-Jui Hsieh, and Yang You. Towards efficient and scalable sharpness-aware minimization. arXiv preprint arXiv:2203.02714, 2022. 2

[31] Charles Mackin, Hsinyu Tsai, Stefano Ambrogio, Pritish Narayanan, An Chen, and Geoffrey W. Burr. Weight programming in DNN analog hardware accelerators in the presence of NVM variability. Advanced Electronic Materials, 2019. 2

[32] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In International Conference on Learning Representations, 2018. 2

[33] Gonçalo Mordido and Christoph Meinel. Mark-evaluate: Assessing language generation using population estimation methods. In International Conference on Computational Linguistics, 2020. 8

[34] Gonçalo Mordido, Julian Niedermeier, and Christoph Meinel. Assessing image and text generation with topological analysis and fuzzy logic. In Winter Conference on Applications of Computer Vision, 2021. 8

[35] Gonçalo Mordido, Matthijs Van Keirsbilck, and Alexander Keller. Instant quantization of neural networks using monte carlo methods. In Workshop on Energy Efficient Machine Learning and Cognitive Computing - NeurIPS Edition, 2019. 1

[36] Gonçalo Mordido, Matthijs Van Keirsbilck, and Alexander Keller. Monte carlo gradient quantization. In Conference on Computer Vision and Pattern Recognition Workshops, 2020. 1

[37] Gonçalo Mordido, Matthijs Van keirsbilck, and Alexander Keller. Compressing 1D time-channel separable convolutions using sparse random ternary matrices. In Interspeech, 2021. 1

[38] Gonçalo Mordido, Haojin Yang, and Christoph Meinel. Dropout-GAN: Learning from a dynamic ensemble of discriminators. arXiv preprint arXiv:1807.11346, 2018. 8

[39] Gonçalo Mordido, Haojin Yang, and Christoph Meinel. microbatchGAN: Stimulating diversity with multi-adversarial discrimination. In Winter Conference on Applications of Computer Vision, 2020. 8

[40] Gonçalo Mordido, Haojin Yang, and Christoph Meinel. Evaluating post-training compression in GANs using locality-sensitive hashing. arXiv preprint arXiv:2103.11912, 2021. 8

[41] SR Nandakumar, Irem Boybat, Vinay Joshi, Christophe Piveteau, Manuel Le Gallo, Bipin Rajendran, Abu Sebastian, and Evangelos Eleftheriou. Phase-change memory models for deep learning training and inference. In International Conference on Electronics, Circuits and Systems, 2019. 7

[42] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. Advances in Neural Information Processing Systems, 2017. 1

[43] Julian Niedermeier, Gonçalo Mordido, and Christoph Meinel. Improving the evaluation of generative models with fuzzy logic. arXiv preprint arXiv:2002.03772, 2020. 8

[44] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasa Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintal. PyTorch: An imperative style, high-performance deep learning library. In Advances in Neural Information Processing Systems, 2019. 4

[45] Malte J Rasch, Diego Moreda, Tayfun Gokmen, Manuel Le Gallo, Fabio Carta, Cindy Goldberg, Kaoutar El Maghraoui, Abu Sebastian, and Vijay Narayanan. A flexible and fast PyTorch toolkit for simulating training and inference on analog crossbar arrays. In International Conference on Artificial Intelligence Circuits and Systems, 2021. 7

[46] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. ImageNet large scale visual recognition challenge. International Journal of Computer Vision, 2015. 3

[47] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In Conference on Computer Vision and Pattern Recognition, 2018. 3

[48] Jonathan Sauder, Ting Hu, Xiaoyin Che, Goncalo Mordido, Haojin Yang, and Christoph Meinel. Best student forcing: A simple training mechanism in adversarial language generation. In Language Resources and Evaluation Conference, 2020. 8

[49] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556, 2014. 3

[50] Katie Spoon, Hsinyu Tsai, An Chen, Malte J. Rasch, Stefano Ambrogio, Charles Mackin, Andrea Facoli, Alexander M. Friz, Pritish Narayanan, Milos Stanisavljevic, and Geoffrey W. Burr. Toward software-equivalent accuracy on Transformer-based deep neural networks with analog memory devices. Frontiers in Computational Neuroscience, 2021. 1, 2
[51] David Stutz, Nandhini Chandramoorthy, Matthias Hein, and Bernt Schiele. Bit error robustness for energy-efficient DNN accelerators. *Machine Learning and Systems*, 2021. 2, 3, 4, 5

[52] David Stutz, Matthias Hein, and Bernt Schiele. Relating adversarially robust generalization to flat minima. In *International Conference on Computer Vision*, 2021. 2, 7

[53] Xu Sun, Zhiyuan Zhang, Xuancheng Ren, Ruixuan Luo, and Liangyou Li. Exploring the vulnerability of deep neural networks: A study of parameter corruption. *AAAI Conference on Artificial Intelligence*, 2021. 2

[54] Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel S Emer. Efficient processing of deep neural networks. *Synthesis Lectures on Computer Architecture*, 2020. 1

[55] Thierry Tambe, Coleman Hooper, Lillian Pentecost, Tianyu Jia, En-Yu Yang, Marco Donato, Victor Sanh, Paul Whatmough, Alexander M Rush, David Brooks, et al. EdgeBERT: Sentence-level energy optimizations for latency-aware multi-task NLP inference. *International Symposium on Microarchitecture*, 2021. 1

[56] H. Tsai, S. Ambrogio, C. Mackin, P. Narayanan, R. M. Shelby, K. Rocki, A. Chen, and G. W. Burr. Inference of long-short term memory networks at software-equivalent accuracy using 2.5M analog phase change memory devices. In *Symposium on VLSI Technology*, 2019. 2, 3

[57] Cong Xu, Dimin Niu, Naveen Muralimanohar, Norman P. Jouppi, and Yuan Xie. Understanding the trade-offs in multi-level cell ReRAM memory design. In *Annual Design Automation Conference*, 2013. 1

[58] Yang Zhao, Hao Zhang, and Xiuyuan Hu. SS-SAM: Stochastic scheduled sharpness-aware minimization for efficiently training deep neural networks. *arXiv preprint arXiv:2203.09962*, 2022. 2

[59] Wenxuan Zhou and Muhao Chen. $\delta$-SAM: Sharpness-aware minimization with dynamic reweighting. *arXiv preprint arXiv:2112.08772*, 2021. 2

[60] Juntang Zhuang, Boqing Gong, Liangzhe Yuan, Yin Cui, Hartwig Adam, Nicha C Dvornek, sekhar tatikonda, James s Duncan, and Ting Liu. Surrogate gap minimization improves sharpness-aware training. In *International Conference on Learning Representations*, 2022. 2
A. Hyper-parameter tuning

The considered ranges for the different hyper-parameters are presented in Tab. 1. The best configurations based on performance after training for CIFAR-10, CIFAR-100, and ImageNet are presented in Tabs. 2, 3, and 4, respectively. These configurations were the ones used to report the results in the main paper. We note that for the models trained with Adam, we used the ResNet-32 configurations reported in Tab. 2.

Table 1. Hyper-parameter choices for the different methods.

| HYPER-PARAMETER | CHOICES |
|-----------------|---------|
| ASAM’s $\rho$   | $\{0.5, 1.0, 1.5, 2.0\}$ |
| SAM’s $\rho$    | $\{0.05, 0.1, 0.2, 0.5\}$ |
| $\alpha$        | $\{1.5, 2.0, 2.5\}$ |
| $c$             | $\{\pm 0.05, \pm 0.10, \pm 0.15, \pm 0.20\}$ |

Table 2. Best hyper-parameter configurations on CIFAR-10.

| MODEL          | METHOD         | BEST CONFIG. |
|----------------|----------------|--------------|
| ResNet-32      | SGD + noise    | $\alpha = 2.0$ |
| SGD + clipping | $c = \pm 0.15$ |
| SAM            | $\rho = 0.05$  |
| SAM + noise    | $\rho = 0.05, \alpha = 2.0$ |
| SAM + clipping | $\rho = 0.05, c = \pm 0.2$ |
| ASAM           | $\rho = 1.0$   |
| ASAM + noise   | $\rho = 0.5, \alpha = 2.0$ |
| ASAM + clipping| $\rho = 1.0, c = \pm 0.2$ |

| MODEL          | METHOD         | BEST CONFIG. |
|----------------|----------------|--------------|
| MobileNetV2    | SGD + noise    | $\alpha = 2.0$ |
| SGD + clipping | $c = \pm 0.15$ |
| SAM            | $\rho = 0.1$   |
| SAM + noise    | $\rho = 0.1, \alpha = 2.0$ |
| SAM + clipping | $\rho = 0.1, c = \pm 0.2$ |
| ASAM           | $\rho = 1.0$   |
| ASAM + noise   | $\rho = 1.0, \alpha = 2.0$ |
| ASAM + clipping| $\rho = 1.0, c = \pm 0.2$ |

| MODEL          | METHOD         | BEST CONFIG. |
|----------------|----------------|--------------|
| VGG-13         | SGD + noise    | $\alpha = 2.0$ |
| SGD + clipping | $c = \pm 0.15$ |
| SAM            | $\rho = 0.1$   |
| SAM + noise    | $\rho = 0.1, \alpha = 2.0$ |
| SAM + clipping | $\rho = 0.1, c = \pm 0.2$ |
| ASAM           | $\rho = 0.5$   |
| ASAM + noise   | $\rho = 0.5, \alpha = 2.0$ |
| ASAM + clipping| $\rho = 0.5, c = \pm 0.2$ |

Table 3. Best hyper-parameter configurations on CIFAR-100.

| MODEL          | METHOD         | BEST CONFIG. |
|----------------|----------------|--------------|
| ResNet-18      | SGD + noise    | $\alpha = 2.5$ |
| SGD + clipping | $c = \pm 0.15$ |
| SAM            | $\rho = 0.1$   |
| SAM + noise    | $\rho = 0.1, \alpha = 2.0$ |
| SAM + clipping | $\rho = 0.1, c = \pm 0.2$ |
| ASAM           | $\rho = 0.5$   |
| ASAM + noise   | $\rho = 0.5, \alpha = 2.0$ |
| ASAM + clipping| $\rho = 0.5, c = \pm 0.2$ |

Table 4. Best hyper-parameter configurations on ImageNet.

B. Additional sharpness experiments

Visual correlations between loss sharpness and model robustness using SAM’s $m$-sharpness on models trained on CIFAR10/100, optimized by Adam, and finetuned ImageNet are presented in Figs. 11, 13, and 15, respectively. Results using Keskar et al. [21]’s sharpness are also shown on 10, 12, and 14. Overall, we observe that SAM’s $m$-sharpness shows better visual correlation than Keskar et al. [21]’s notion of sharpness. The best neighborhood configurations for SAM and ASAM’s $m$-sharpness are shown in Fig. 16. We observe that ASAM’s $m$-sharpness shows the most visual correlation compared to the two other sharpness metrics, as also discussed in the main paper.
Figure 10. Correlation between Keskar et al. [21]’s sharpness and the performance gap of the different methods. For each experiment, we plot the mean and standard deviation over 10 inference runs.
Figure 11. Correlation between SAM’s $m$-sharpness (Eq. (6), $\rho = 0.05$) and the performance gap of the different methods. For each experiment, we plot the mean and standard deviation over 10 inference runs.
Figure 12. Correlation between Keskar et al. [21]’s and the performance gap of the different methods on ResNet-32 trained on CIFAR-10 using the Adam optimizer. For each experiment, we plot the mean and standard deviation over 10 inference runs.

Figure 13. Correlation between SAM’s $m$-sharpness (Eq. (6), $\rho = 0.05$) and the performance gap of the different methods on ResNet-32 trained on CIFAR-10 using the Adam optimizer. For each experiment, we plot the mean and standard deviation over 10 inference runs.

Figure 14. Correlation between Keskar et al. [21]’s sharpness and the performance gap of the different methods on ResNet-18 finetuned on ImageNet. For each experiment, we plot the mean and standard deviation over 3 inference runs.

Figure 15. Correlation between SAM’s $m$-sharpness (Eq. (6), $\rho = 0.05$) and the performance gap of the different methods on ResNet-18 finetuned on ImageNet. For each experiment, we plot the mean and standard deviation over 3 inference runs.
Figure 16. Correlation between sharpness and robustness, i.e., performance gap between $\sigma_c = 0.0$ and $\sigma_c = 0.4$, on the ResNet-32 models trained by the different methods on CIFAR-10. For SAM and ASAM’s $m$-sharpness, we plot the neighborhood size for which sharpness achieved the highest Pearson correlation.