Relating Adversarially Robust Generalization to Flat Minima

David Stutz [1] **Matthias Hein** [2] **Bernt Schiele** [1]

Abstract

Adversarial training (AT) has become the de-facto standard to obtain models robust against adversarial examples. However, AT exhibits severe robust overfitting: cross-entropy loss on adversarial examples (robust loss) decreases continuously on training examples, while eventually increasing on test examples. This leads to poor robust generalization, i.e., low adversarial robustness on new examples. We study the relationship between robust generalization and flatness of the robust loss landscape in weight space, i.e., whether robust loss changes significantly when perturbing weights. To this end, we propose a metric to measure "robust flatness" and find a strong **correlation between good robust generalization and flatness**. Throughout training, flatness reduces during overfitting, i.e., early stopping effectively finds flatter minima and AT variants such as AT-AWP or TRADES and simple regularization techniques such as AutoAugment or label noise that improve robustness also correspond to flatter minima.

1. Introduction

In order to obtain robustness against adversarial examples (Szegedy et al., 2014), *adversarial training (AT)* (Madry et al., 2018) augments training with adversarial examples generated on-the-fly. AT is known to require more training data (Schmidt et al., 2018), leading to generalization problems (Farnia et al., 2019). *Robust overfitting* (Rice et al., 2020) has been identified as the main obstacle: adversarial robustness on test examples eventually starts to decrease, while robustness on training examples continues to increase (cf. Fig. 2). This is observed as increasing *robust loss (RLoss)* or *robust test error (RErr)*, i.e., (cross-entropy) loss and test error on adversarial examples. Despite recent work (Singla et al., 2021; Wu et al., 2020; Hwang et al., 2020), it remains an open problem.

[1]Max Planck Institute for Informatics, Saarland Informatics Campus, Germany [2]University of Tübingen, Germany. Correspondence to: David Stutz <david.stutz@mpi-inf.mpg.de>.

Presented at the ICML 2021 Workshop on Uncertainty and Robustness in Deep Learning., Copyright 2021 by the author(s).

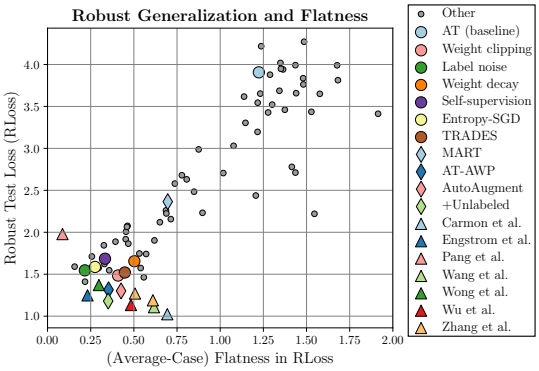

Figure 1: **Robust Generalization and Flatness:** Robust loss (RLoss ↓, y-axis), i.e., cross-entropy loss on PGD adversarial examples (Madry et al., 2018), against our flatness measure in weight space (lower is "flatter", x-axis). Popular AT variants improving adversarial robustness on CIFAR10, e.g., TRADES (Zhang et al., 2019) or AT-AWP (Wu et al., 2020), correspond to flatter minima. Vice-versa, explicitly regularizing flatness, e.g., Entropy-SGD (Chaudhari et al., 2017), improves robustness. There is a **clear relationship between good robust generalization and flatness in RLoss.** ●,♦ Our models, without early stopping. ▲ RobustBench (Croce et al., 2020) models *with* early stopping.

In *"clean"* generalization, overfitting is well-studied and commonly tied to flatness of the loss landscape in weight space, both visually (Li et al., 2018) and empirically (Neyshabur et al., 2017; Keskar et al., 2017). In general, the optimal weights on test examples do not coincide with the minimum found on training examples. Flatness ensures that the loss does *not* increase significantly in a neighborhood around the found minimum. Therefore, flatness leads to good generalization because the loss on test examples does not increase significantly, cf. Fig. 3 (right). (Li et al., 2018) showed that *visually* flatter minima correspond to better generalization. (Neyshabur et al., 2017; Keskar et al., 2017) formalize this idea by measuring the change in loss within a local neighborhood. Furthermore, explicitly encouraging flatness during training has been shown to be successful in practice (Cicek & Soatto, 2019; Lin et al., 2020; Chaudhari et al., 2017; Izmailov et al., 2018).

Recently, (Wu et al., 2020) applied the idea of flat minima to AT: AT-AWP regularizes AT with *adversarial weight perturbations* to find flatter minima of the *robust* loss land-

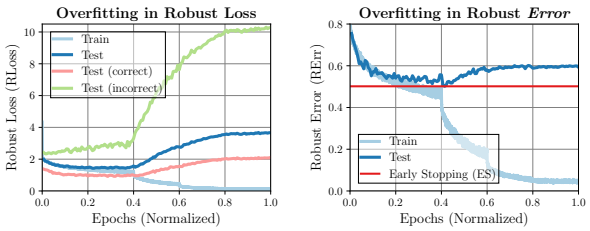

Figure 2: **Robust Overfitting:** Robust loss (RLoss, left) and robust error (RErr, right) over normalized epochs on CIFAR10. **Left:** Training RLoss (light blue) reduces continuously throughout training, while test RLoss (dark blue) eventually increases again. Robust overfitting is *not* limited to incorrectly classified examples (green), but also affects correctly classified ones (rose). **Right:** Similar behavior, but less pronounced, can be observed considering RErr. We also show RErr obtained through early stopping (red).

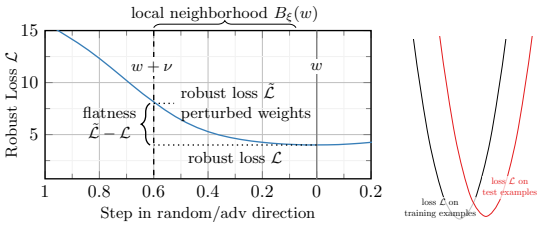

Figure 3: **Measuring Flatness. Left:** Measuring flatness in a random direction (blue) by computing the difference between RLoss $\tilde{\mathcal{L}}$ *after* perturbing weights (i.e., $w+\nu$) and the "reference" RLoss $\mathcal{L}$ given a local neighborhood $B_\xi(w)$ around the found weights w. In practice, we average across several random directions. **Right:** Large changes in RLoss around the "sharp" minimum causes poor generalization from training (black) to test examples (red).

scape. This reduces the impact of robust overfitting but does not *avoid* robust overfitting – early stopping is still necessary. Flatness is only assessed visually. Similarly, (Gowal et al., 2020) shows that weight averaging (Izmailov et al., 2018) improves robust generalization, indicating that flatness might be beneficial in general. This raises the question whether other "tricks" (Pang et al., 2020; Gowal et al., 2020), e.g., different activation functions (Singla et al., 2021) or approaches such as AT with self-supervision (Hendrycks et al., 2019)/unlabeled examples (Carmon et al., 2019) are successful *because of* finding flatter minima.

Contributions: We study **whether flatness of the robust loss (RLoss) in weight space improves robust generalization**. To this end, we propose a scale-invariant (Dinh et al., 2017) flatness measures for the *robust* case and show that **robust generalization generally improves alongside flatness** and vice-versa: Fig. 1 plots RLoss (lower is more robust, y-axis) against flatness in RLoss (lower is flatter, x-axis), showing a clear relationship. This trend is also present when considering the robust generalization gap (i.e., test - train RLoss). Furthermore, our experiments cover a wide range of AT variants and regularization schemes (cf. Tab. 1) on CIFAR10. Furthermore, we consider hyper-parameters such as learning rate schedule, weight decay or activation functions (Elfwing et al., 2018; Misra, 2020; Hendrycks & Gimpel, 2016), and methods explicitly improving flatness (Chaudhari et al., 2017; Izmailov et al., 2018). This is a short version of our pre-print (Stutz et al., 2021b).

2. Robust Generalization and Flat Minima

We consider robust generalization and overfitting in the context of flatness of the *robust* loss landscape in weight space, i.e., w.r.t. changes in the weights. While flat minima have consistently been linked to standard generalization (Hochreiter & Schmidhuber, 1997; Li et al., 2018; Neyshabur et al., 2017; Keskar et al., 2017), this relationship remains unclear for adversarial robustness. We briefly provide some background and discuss robust overfitting before introducing our flatness measure based on the change in robust loss along random weight directions in a local neighborhood.

Notation: Let f be a neural network taking input $x \in [0,1]^D$ and weights $w \in \mathbb{R}^W$ and predicting a label $f(x;w)$. Given a true label y, an adversarial example is a perturbation $\tilde{x} = x + \delta$ such that $f(\tilde{x};w) \neq y$. The perturbation δ is enforced to be nearly imperceptible using a L_p constraint: $\|\delta\|_p \leq \epsilon$. To improve robustness, AT injects adversarial examples during training and minimizes robust loss (RLoss), i.e., $\max_{\|\delta\|_\infty \leq \epsilon} \mathcal{L}(f(x+\delta;w),y)$ with $\mathcal{L}$ being the cross-entropy loss. The inner maximization is tackled using projected gradient descent (PGD) (Madry et al., 2018). We focus on $p=\infty$, e.g., $\epsilon = 8/255$ on CIFAR10, and we consider both RLoss, approximated using PGD, and robust test error (RErr), using AutoAttack (Croce & Hein, 2020).

Robust Overfitting: Following (Rice et al., 2020), Fig. 2 illustrates the problem of *robust* overfitting, plotting RLoss (left) and RErr (right) over epochs. Shortly after the first learning rate drop (at epoch 60, i.e., 40% of training), test RLoss and RErr start to increase significantly, while robustness on training examples continues to improve. In contrast to (Rice et al., 2020), mostly focusing on RErr, Fig. 1 shows that RLoss overfits more severely. For now, RLoss and RErr do clearly not move "in parallel" and RLoss, reaching values around 4, is higher than for a random classifier (which is possible considering *adversarial* examples). This is primarily due to an extremely high RLoss on incorrectly classified test examples (which are "trivial" adversarial examples). We emphasize, however, that robust overfitting also occurs on correctly classified test examples.

Flatness: We consider how RLoss changes w.r.t. perturbations in the weights w. Generally, we expect flatter minima to generalize better as the loss does not change significantly within a neighborhood around the found weights. Even if

Model	RErr ↓
AT (baseline)	62.8
Scaled $\times 0.5$	62.8
Scaled $\times 2$	62.8
MiSH	59.8
Batch size 8	58.2
Adam	57.5

Model	RErr ↓
Label smoothing	61.2
MART	61
Entropy-SGD	58.6
Self-supervision	57.1
TRADES	56.7
AT-AWP	54.3

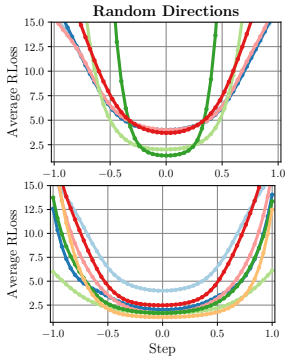

Figure 4: **Visualizing Flatness:** RLoss landscape across 10 random directions for AT and scaled variants ($\times 2$, $\times 0.5$). Training with Adam (Kingma & Ba, 2015) or MiSH (Misra, 2020) improve adversarial robustness (lower RErr vs. AutoAttack (Croce & Hein, 2020)) but do *not* result in (visually) flatter minima. In contrast, AT-AWP or Entropy-SGD (Chaudhari et al., 2017) improve robustness *and* flatness.

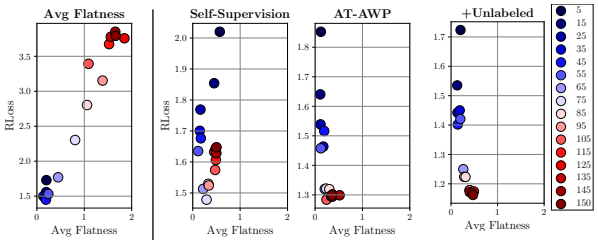

Figure 5: **Flatness Throughout Training.** Test RLoss (y-axis) plotted against flatness in RLoss (x-axis) during training, showing a clear correlation. AT with self-supervision reduces the impact of robust overfitting (RLoss increases less) and simultaneously favors flatter minima. This behavior is pronounced for AT-AWP, explicitly optimizing flatness, and AT with additional unlabeled examples.

the loss landscape on test examples changes, loss remains small, ensuring good generalization. The contrary case is illustrated in Fig. 3 (right). The easiest way to "judge" flatness is visual inspection (Li et al., 2018) where the loss landscape is visualized along random directions after normalizing the weights *per-filter*. The normalization is important to handle difference scales (cf. Fig. 4), i.e., weight distributions, and allows comparison across models. However, as shown in Fig. 4, judging flatness visually is difficult: Considering random weight directions, AT with Adam (Kingma & Ba, 2015) or small batch size improves adversarial robustness, but the found minima look less flat (top). For other approaches, e.g., TRADES (Zhang et al., 2019) or AT-AWP (Wu et al., 2020), results look indeed flatter while also improving robustness (bottom). Furthermore, not only flatness but also the vertical "height" of the loss landscape matters and it is impossible to tell "how much" flatness is necessary.

Average-Case Flatness Measure: Thus, to objectively measure and compare flatness, we draw inspiration from (Neyshabur et al., 2017) and propose an "average-case"

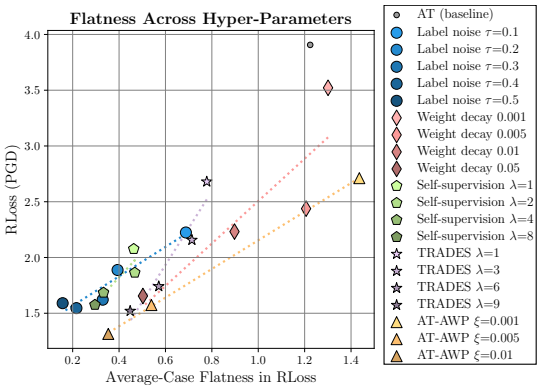

Figure 6: **Flatness Across Hyper-Parameters:** RLoss (y-axis) vs. flatness (x-axis) for selected methods and hyper-parameters (cf. supplementary material). For example, we consider different strengths of weight decay (rose) or sizes ξ of adversarial weight perturbations for AT-AWP (orange). For clarity, we plot (dotted) lines representing the trend per method. Clearly, improved adversarial robustness, i.e., low RLoss, is related to improved flatness.

flatness measures adapted to the robust loss. Considering random weight perturbations $\nu \in B_\xi(w)$ within the ξ-neighborhood of w, flatness is computed as

$$\mathbb{E}_\nu\Big[\max_{\|\delta\|_\infty \leq \epsilon} \mathcal{L}(f(x+\delta; w+\nu), y)\Big] - \max_{\|\delta\|_\infty \leq \epsilon} \mathcal{L}(f(x+\delta; w), y) \quad (1)$$

averaged over test examples x, y, as illustrated in Fig. 3. We define $B_\xi(w)$ using *relative* L_2-balls per layer:

$$B_\xi(w) = \{w + \nu : \|\nu^{(l)}\|_2 \leq \xi \|w^{(l)}\|_2 \forall \text{ layers } l\}. \quad (2)$$

Note that the second term in Eq. (1), i.e., the "reference" robust loss, is important to make the measure independent of the absolute loss (i.e., corresponding to the vertical shift in Fig. 3, left). In practice, ξ can be as large as 0.5. We refer to Eq. (1) as **flatness in RLoss**. By construction, Eq. (2) is scale-invariant as the weight neighborhood is defined *relative* to the L_2 norm of the weights.

3. Experiments

We conduct experiments on CIFAR10 (Krizhevsky, 2009), where our *AT baseline* uses ResNet-18 (He et al., 2016) and is trained using SGD and a multi-step learning rate schedule. For PGD, we use 7 iterations and $\epsilon = 8/255$ for L_∞ adversarial examples. PGD-7 is also used for early stopping on the last 500 test examples. We do *not* use early stopping by default. For evaluation on the first 1000 test examples, we run PGD with 20 iterations, 10 random restarts to estimate RLoss and AutoAttack (Croce & Hein, 2020) to estimate RErr. In Eq. (1), we use 10 random weight perturbations with $\xi = 0.5$. We consider various AT variants, hyper-parameters and optimization strategies as summarized in

Model (sorted asc. by test RErr) (split at 70%/30% percentiles)	Robustness RErr ↓ (test)	RErr ↓ (train)	Flatness ↓ (RLoss)	Early Stop. RErr ↓ (early stop)
+Unlabeled (Carmon et al., 2019)	48.9	43.2 (-5.7)	0.32	48.9 (-0.0)
Cyclic	53.6	35.4 (-18.2)	0.35	53.6 (-0.0)
AutoAugment (Cubuk et al., 2018)	54.0	47.9 (-6.1)	0.49	53.5 (-0.5)
AT-AWP (Wu et al., 2016)	54.3	43.1 (-11.2)	0.35	53.6 (-0.7)
Label noise	56.2	30.0 (-26.2)	0.33	55.5 (-0.7)
Weight clipping (Stutz et al., 2021a)	56.5	39.0 (-17.5)	0.41	56.5 (-0.0)
TRADES (Zhang et al., 2019)	56.7	15.8 (-40.9)	0.57	53.4 (-3.3)
Self-supervision (Hendrycks et al., 2019)	57.1	45.0 (-12.1)	0.33	56.8 (-0.3)
Weight decay	58.1	32.8 (-25.3)	0.50	54.8 (-3.3)
Entropy-SGD (Chaudhari et al., 2017)	58.6	46.1 (-12.5)	0.28	56.9 (-1.7)
MiSH (Misra, 2020)	59.8	5.3 (-54.5)	1.56	53.7 (-6.1)
"Late" multi-step	59.8	18.4 (-41.4)	0.80	57.8 (-2.0)
SiLU (Elfwing et al., 2018)	60.0	5.6 (-54.4)	1.71	53.7 (-6.3)
Weight averaging (Izmailov et al., 2018)	60.0	10.0 (-50.0)	1.28	53.0 (-7.0)
Larger ϵ=9/255	60.9	11.1 (-49.8)	1.33	53.8 (-7.1)
MART (Wang et al., 2020)	61.0	20.8 (-40.2)	0.73	54.7 (-6.3)
GeLU (Hendrycks & Gimpel, 2016)	61.1	3.2 (-57.9)	1.55	56.7 (-4.4)
Label smoothing (Szegedy et al., 2016)	61.2	8.0 (-53.2)	0.65	54.0 (-7.2)
AT (baseline)	62.8	10.7 (-52.1)	1.21	54.6 (-8.2)

Table 1: **Quantitative Results:** Test and train RErr (first, second column) and flatness in RLoss (third column) for selected methods, corresponding to Fig. 7. We also report RErr after early stopping (fourth column). We split methods into good, average, and poor robustness using the 30% and 70% percentiles. Most methods improve adversarial robustness alongside flatness. Commonly, this leads to increased train RErr, i.e., smaller robust generalization gap.

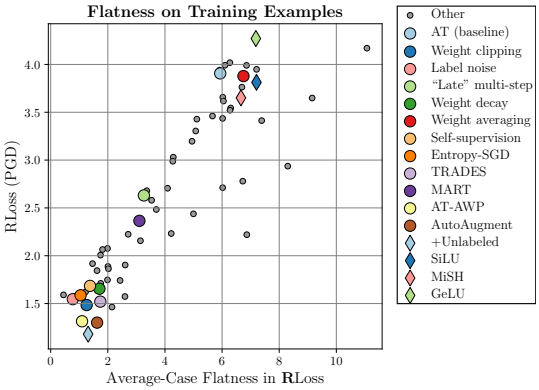

Figure 7: **Flatness on Training Set:** RLoss (on test examples) plotted against flatness measured on *training* examples. The clear relationship between robustness and flatness is preserved, i.e., flatness on the training set is a good predictor of robustness (i.e., RLoss) at test time.

Tab. 1. We also use models from RobustBench (Croce et al., 2020), obtained using early stopping.

Recent work (Wu et al., 2020; Gowal et al., 2020), and Tab. 1 (fourth column), suggest that robust overfitting can be mitigated using regularization. We hypothesize that this is because strong regularization helps to find flatter minima in the RLoss landscape.

Flatness in RLoss "Explains" Overfitting: Considering Fig. 5, we find that flatness reduces significantly during robust overfitting. Namely, flatness "explains" the increased RLoss caused by overfitting very well. We explicitly plot RLoss (y-axis) against flatness in RLoss (x-axis) across epochs (dark blue to dark red): RLoss and flatness clearly worsen "alongside" each other during overfitting. Methods such as AT with self-supervision, AT-AWP or AT with unlabeled examples avoid both robust overfitting *and* sharp minima (right). This relationship generalizes to different hyper-parameter choices of these methods: Fig. 6 plots RLoss (y-axis) vs. flatness (x-axis) across different hyper-parameters. Again, e.g., for TRADES or AT-AWP, hyper-parameters with lower RLoss also correspond to flatter minima. In fact, Fig. 6 indicates that the connection between robustness and flatness also generalizes *across* different methods (and individual models).

Improved Robustness Through Flatness: Indeed, across all trained models, we found a **strong correlation between robust generalization and flatness**. Here, we mainly consider RLoss to assess robust generalization as improvements in RLoss above ∼2.3 have, on average, only small impact on RErr (for 10 classes), see (Stutz et al., 2021b) for a discussion. Thus, Fig. 1 plots *RLoss* (y-axis) against *average-case* flatness in RLoss (x-axis), highlighting selected models. This reveals a *clear correlation between robustness and flatness*: More robust methods, e.g., AT with unlabeled examples or AT-AWP, correspond to flatter minima. Similarly, methods improving flatness, e.g., Entropy-SGD, weight decay or weight clipping, improve adversarial robustness. This also translates to RErr (middle right), subject to the described bend at RLoss≈2.3. These results are summarized in tabular form in Tab. 1: Grouping methods by good, average or poor robustness, we find that methods need some degree of flatness to be successful. Overall, flatness in RLoss has clear advantages in terms of robust generalization, i.e., low RLoss on test examples. We emphasize that this relationship is preserved when evaluating flatness on training examples, see Fig. 7, or plotting flatness against the robust generalization gap (i.e., test - train RLoss), as detailed in (Stutz et al., 2021b).

4. Conclusion

We studied the relationship between adversarial robustness, also considering robust overfitting (Rice et al., 2020), and flatness of the robust loss landscape w.r.t. random perturbations in the weight space. We introduced a scale-invariant measure of robust flatness and considered popular adversarial training (AT) variants, e.g., TRADES (Zhang et al., 2019), AT-AWP (Wu et al., 2020) AT with self-supervision (Hendrycks et al., 2019) or unlabeled examples (Carmon et al., 2019). We show a **clear relationship between adversarial robustness and flatness**: more robust methods predominantly find flatter minima and, vice versa, approaches known to improve flatness help AT improve robustness.

References

Carmon, Y., Raghunathan, A., Schmidt, L., Duchi, J. C., and Liang, P. Unlabeled data improves adversarial robustness. In *NeurIPS*, 2019.

Chaudhari, P., Choromanska, A., Soatto, S., LeCun, Y., Baldassi, C., Borgs, C., Chayes, J. T., Sagun, L., and Zecchina, R. Entropy-sgd: Biasing gradient descent into wide valleys. In *ICLR*, 2017.

Cicek, S. and Soatto, S. Input and weight space smoothing for semi-supervised learning. In *ICCV Workshops*, 2019.

Croce, F. and Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.

Croce, F., Andriushchenko, M., Sehwag, V., Flammarion, N., Chiang, M., Mittal, P., and Hein, M. Robustbench: a standardized adversarial robustness benchmark. *arXiv.org*, abs/2010.09670, 2020.

Cubuk, E. D., Zoph, B., Mané, D., Vasudevan, V., and Le, Q. V. Autoaugment: Learning augmentation policies from data. *arXiv.org*, abs/1805.09501, 2018.

Dinh, L., Pascanu, R., Bengio, S., and Bengio, Y. Sharp minima can generalize for deep nets. In *ICML*, 2017.

Elfwing, S., Uchibe, E., and Doya, K. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *NN*, 107, 2018.

Engstrom, L., Ilyas, A., Salman, H., Santurkar, S., and Tsipras, D. Robustness (python library), 2019. URL https://github.com/MadryLab/robustness.

Farnia, F., Zhang, J. M., and Tse, D. Generalizable adversarial training via spectral normalization. In *ICLR*, 2019.

Gowal, S., Qin, C., Uesato, J., Mann, T. A., and Kohli, P. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv.org*, abs/2010.03593, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *CVPR*, 2016.

Hendrycks, D. and Gimpel, K. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *arXiv.org*, abs/1606.08415, 2016.

Hendrycks, D., Mazeika, M., Kadavath, S., and Song, D. Using self-supervised learning can improve model robustness and uncertainty. In *NeurIPS*, 2019.

Hochreiter, S. and Schmidhuber, J. Flat minima. *NC*, 9, 1997.

Hwang, J., Lee, Y., Oh, S., and Bae, Y.-S. Adversarial training with stochastic weight average. *arXiv.org*, abs/2009.10526, 2020.

Izmailov, P., Podoprikhin, D., Garipov, T., Vetrov, D. P., and Wilson, A. G. Averaging weights leads to wider optima and better generalization. In *UAI*, 2018.

Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. On large-batch training for deep learning: Generalization gap and sharp minima. In *ICLR*, 2017.

Kingma, D. P. and Ba, J. Adam: A method for stochastic optimization. In *ICLR*, 2015.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, 2009.

Li, H., Xu, Z., Taylor, G., and Goldstein, T. Visualizing the loss landscape of neural nets. In *NeurIPS*, 2018.

Lin, T., Stich, S. U., Patel, K. K., and Jaggi, M. Don't use large mini-batches, use local SGD. In *ICLR*, 2020.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. *ICLR*, 2018.

Misra, D. Mish: A self regularized non-monotonic activation function. In *BMVC*, 2020.

Neyshabur, B., Bhojanapalli, S., McAllester, D., and Srebro, N. Exploring generalization in deep learning. In *NeurIPS*, 2017.

Pang, T., Yang, X., Dong, Y., Su, H., and Zhu, J. Bag of tricks for adversarial training. *arXiv.org*, abs/2010.00467, 2020.

Rice, L., Wong, E., and Kolter, J. Z. Overfitting in adversarially robust deep learning. In *ICML*, 2020.

Schmidt, L., Santurkar, S., Tsipras, D., Talwar, K., and Madry, A. Adversarially robust generalization requires more data. In *NeurIPS*, 2018.

Singla, V., Singla, S., Jacobs, D., and Feizi, S. Low curvature activations reduce overfitting in adversarial training. *arXiv.org*, abs/2102.07861, 2021.

Stutz, D., Chandramoorthy, N., Hein, M., and Schiele, B. Bit error robustness for energy-efficient dnn accelerators. In *MLSys*, 2021a.

Stutz, D., Hein, M., and Schiele, B. Relating adversarially robust generalization to flat minima. *arXiv.org*, abs/2104.04448, 2021b.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I. J., and Fergus, R. Intriguing properties of neural networks. In *ICLR*, 2014.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., and Wojna, Z. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., and Gu, Q. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*, 2020.

Wong, E., Rice, L., and Kolter, J. Z. Fast is better than free: Revisiting adversarial training. *arXiv.org*, abs/2001.03994, 2020.

Wu, D., Xia, S., and Wang, Y. Adversarial weight perturbation helps robust generalization. In *NeurIPS*, 2020.

Wu, J., Zhang, C., Xue, T., Freeman, B., and Tenenbaum, J. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *NeurIPS*, 2016.

Zhang, H., Yu, Y., Jiao, J., Xing, E. P., Ghaoui, L. E., and Jordan, M. I. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019.

Zhang, J., Xu, X., Han, B., Niu, G., Cui, L., Sugiyama, M., and Kankanhalli, M. S. Attacks which do not kill training make adversarial learning stronger. In *ICML*, 2020.