# Relating Adversarially Robust Generalization to Flat Minima

David Stutz[1] Matthias Hein[2] Bernt Schiele[1]
[1]Max Planck Institute for Informatics, Saarbrücken, {dstutz,schiele}@mpi-inf.mpg.de
[2]University of Tübingen, Tübingen, matthias.hein@uni-tuebingen.de

## 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. Similarly, 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.*

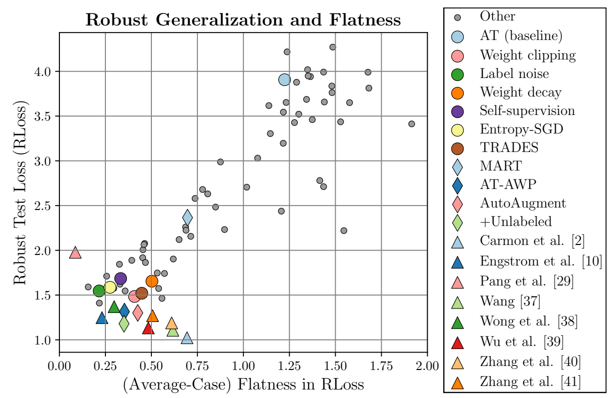


Figure 1: **Robust Generalization and Flatness:** Robust loss (RLoss, lower is more robust, y-axis), i.e., cross-entropy loss on PGD adversarial examples [26], against our flatness measure of RLoss in weight space (lower is "flatter", x-axis). Popular AT variants improving adversarial robustness on CIFAR10, e.g., TRADES [40] or AT-AWP [39], also correspond to flatter minima. Vice-versa, explicitly regularizing flatness, e.g., Entropy-SGD [3], also improves robustness. Across all models, there is a **clear relationship between good robust generalization and flatness in RLoss.** ●,♦ Our models, without early stopping. ▲ RobustBench [5] models *with* early stopping.

## 1. Introduction

In order to obtain robustness against adversarial examples [36], *adversarial training (AT)* [26] augments training with adversarial examples generated on-the-fly. AT is known to require more training data [21, 31], generally leading to generalization problems [11]. *Robust overfitting* [30] 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 typically observed as increasing *robust loss (RLoss)* or *robust test error (RErr)*, i.e., (cross-entropy) loss and test error on adversarial examples. As a result, the *robust generalization gap*, i.e., the difference between test and training robustness, tends to be large. [30], uses early stopping as a simple strategy to avoid robust overfitting. Nevertheless, despite recent work [32, 39, 17], it remains an open and poorly understood problem.

In *"clean"* generalization (i.e., on natural examples), overfitting is well-studied and commonly tied to flatness of the loss landscape in weight space, both visually [24] and empirically [28, 20, 19]. 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 (i.e., small generalization gap, cf. Fig. 3, right). [24] showed that *visually* flatter minima correspond to better generalization. [28, 20] 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 [42, 4, 25, 3, 18].

Recently, [39] applied the idea of flat minima to AT: through *adversarial weight perturbations*, AT is regularized to find flatter minima of the *robust* loss landscape. This reduces the impact of robust overfitting and improves robust

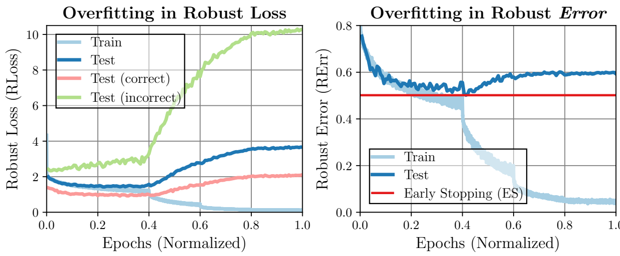


Figure 2: **Robust Overfitting:** Robust loss (RLoss, left) and robust error (RErr, right) over normalized epochs on CIFAR10. **Left:** Training RLoss (light blue) reduces continuously throughout training, while test RLoss (dark blue) eventually increases again. Robust overfitting is *not* limited to 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).

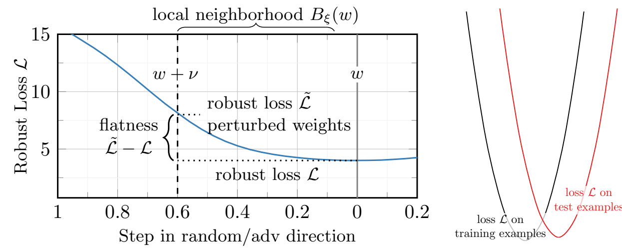


Figure 3: **Measuring Flatness. Left:** Measuring flatness in a random direction (blue) by computing the difference between RLoss $\tilde{\mathcal{L}}$ *after* perturbing weights (i.e., $w + \nu$) and the "reference" RLoss $\mathcal{L}$ given a local neighborhood $B_\xi(w)$ around the found weights $w$, see Sec. 2.1. 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).

generalization, but does not *avoid* robust overfitting. As result, early stopping is still necessary. Unfortunately, flatness is only assessed visually. Similarly, [12] shows that weight averaging [18] improves robust generalization, indicating that flatness might be beneficial in general. This raises the question whether other "tricks" [29, 12], e.g., different activation functions [32], label smoothing [35], or approaches such as AT with self-supervision [15]/unlabeled examples [2] 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 [8] 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 covers a wide range of AT variants on CIFAR10 [39, 40, 37, 15, 2, 1] and various regularization schemes, including AutoAugment [7], label smoothing/noise [35] or weight clipping [33]. Furthermore, we consider hyper-parameters such as learning rate schedule, weight decay or activation functions [9, 27, 14], and methods explicitly improving flatness [3, 18].

This paper is a short version of [34]. It is intended to be self-contained, but we refer to [34] for further discussion.

## 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 [16, 24, 28, 20], 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 (deep) 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 $\delta$ 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) [26]. We focus on $p = \infty$ as this constrains the maximum change per feature/pixel, e.g., $\epsilon = {}^8/_{255}$ on CIFAR10. For evaluation, we consider both RLoss, approximated using PGD, and robust test error (RErr), computed using AutoAttack [6].

**Robust Overfitting:** Following [30], Fig. 2 illustrates the problem of *robust* overfitting, plotting RLoss (left) and RErr (right) over epochs, which we normalize by the total number of epochs for clarity. 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 [30], 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.

### 2.1. Flatness Measure

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 the loss

| Model | RErr ↓ |
|---|---|
| AT (baseline) | 62.8 |
| Scaled ×0.5 | 62.8 |
| Scaled ×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 |

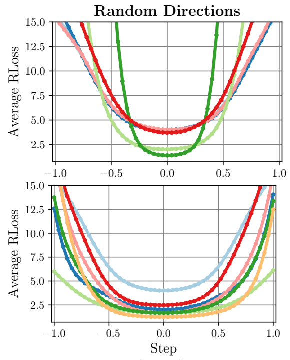


Figure 4: **Visualizing Flatness:** RLoss landscape across 10 random directions for AT and scaled variants (×2, ×0.5). Training with Adam [22] or MiSH [27] improves adversarial robustness (lower RErr vs. AutoAttack [6]) but do *not* result in (visually) flatter minima. In contrast, AT-AWP [39] or Entropy-SGD [3] improve robustness *and* flatness.

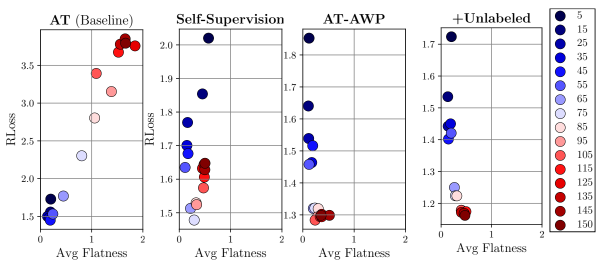


Figure 5: **Flatness Throughout Training.** Test RLoss (y-axis) plotted against flatness in RLoss (x-axis) during training, showing a clear correlation. AT with self-supervision reduces the impact of robust overfitting (RLoss increases less) and simultaneously favors flatter minima. This behavior is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

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, e.g., following [24], 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 [22] or small batch size improves adversarial robustness, but the found minima look less flat (top). For other approaches, e.g., TRADES [40] or AT-AWP [39], 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:** Thus, to objectively measure and compare flatness, we draw inspiration from [28] and propose an "average-case" flatness measures adapted to

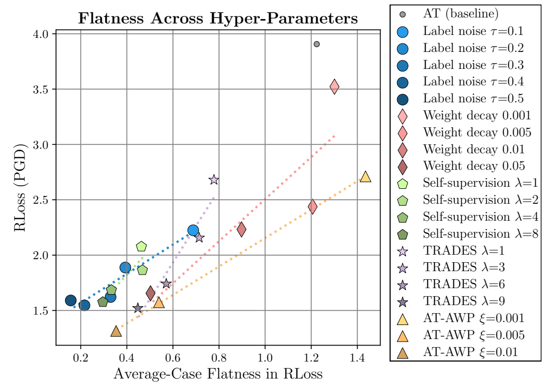


Figure 6: **Flatness Across Hyper-Parameters:** RLoss (y-axis) vs. flatness (x-axis) for selected methods and hyper-parameters (cf. supplementary material). For example, we consider different strengths of weight decay (rose) or sizes $\xi$ of adversarial weight perturbations for AT-AWP (orange). For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

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

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

averaged over test examples $x, y$, as illustrated in Fig. 3. We define $B_\xi(w)$ using *relative* $L_2$-balls per layer as in [39]:

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

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, $\xi$ 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 [23], where our *AT baseline* uses ResNet-18 [13] and is trained using SGD and a multi-step learning rate schedule. For PGD, we use 7 iterations and $\epsilon = 8/255$ for $L_\infty$ 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 (balanced) test examples, we run PGD with 20 iterations, 10 random restarts to estimate RLoss and AutoAttack [6] 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 Tab. 1. We also use models from RobustBench [5], obtained using early stopping.

| Model | Robustness | | Flatness ↓ | Early Stop. |
|---|---|---|---|---|
| (sorted asc. by test RErr) | RErr ↓ | RErr ↓ | | RErr ↓ |
| (split at 70%/30% percentiles) | (test) | (train) | (RLoss) | (early stop) |
| +Unlabeled [2, 1] | 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 [7] | 54.0 | 47.9 (-6.1) | 0.49 | 53.5 (-0.5) |
| AT-AWP [39] | 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 [33] | 56.5 | 39.0 (-17.5) | 0.41 | 56.5 (-0.0) |
| TRADES [40] | 56.7 | 15.8 (-40.9) | 0.57 | 53.4 (-3.3) |
| Self-supervision [15] | 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 [3] | 58.6 | 46.1 (-12.5) | 0.28 | 56.9 (-1.7) |
| MiSH [27] | 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 [9] | 60.0 | 5.6 (-54.4) | 1.71 | 53.7 (-6.3) |
| Weight averaging [18] | 60.0 | 10.0 (-50.0) | 1.28 | 53.0 (-7.0) |
| Larger $\epsilon{=}9/255$ | 60.9 | 11.1 (-49.8) | 1.33 | 53.8 (-7.1) |
| MART [37] | 61.0 | 20.8 (-40.2) | 0.73 | 54.7 (-6.3) |
| GeLU [14] | 61.1 | 3.2 (-57.9) | 1.55 | 56.7 (-4.4) |
| Label smoothing [35] | 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, w/ early stopping, fourth column) and flatness in RLoss (third column) for selected methods. RErr may be slightly higher than reported in the literature due to our setup, e.g., 7 iterations PGD during training. We split methods into good, average, and poor robustness using the 30% and 70% percentiles. Most methods improve adversarial robustness alongside flatness.

### 3.1. Robust Generalization and Flatness in RLoss

Recent work [39, 12], 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 $\sim$2.3 have, on average, only small impact on RErr (for 10 classes). Pushing RLoss below 2.3, in contrast, directly translates to better RErr. This is illustrated in Fig. 7 which plots RErr vs. RLoss for all evaluated models. To avoid this "kink" in the dotted red lines around RLoss≈2.3, 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. Note that Fig. 1 highlights selected models from literature (colored), e.g., from [5] obtained with early stopping, while the described relationship is mostly observed across models without early stopping and with varying hyper-parameters, cf. Fig. 4. We found that this also translates to RErr, 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.

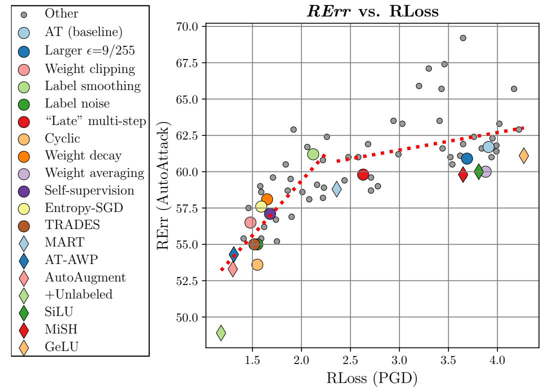


Figure 7: **RLoss and RErr:** RErr plotted against RLoss, showing that improved RLoss does not directly translate to reduced RErr for large RLoss. Here, reducing RLoss mainly means reducing the confidence of adversarial examples, which is necessary to improve adversarial robustness.

## 4. Conclusion

We studied the relationship between adversarial robustness, also considering robust overfitting [30], and flatness of the robust loss (RLoss) 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 [40], MART [37], AT-AWP [39] AT with self-supervision [15] or additional unlabeled examples [2]. Our experiments reveal a **clear relationship between adversarial robustness and flatness** in RLoss: more robust methods predominantly find flatter minima and, vice versa, approaches known to improve flatness help AT improve robustness.

## References


[1] Jean-Baptiste Alayrac, Jonathan Uesato, Po-Sen Huang, Alhussein Fawzi, Robert Stanforth, and Pushmeet Kohli. Are labels required for improving adversarial robustness? In *NeurIPS*, 2019. 2, 4

[2] Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C. Duchi, and Percy Liang. Unlabeled data improves adversarial robustness. In *NeurIPS*, 2019. 2, 4

[3] Pratik Chaudhari, Anna Choromanska, Stefano Soatto, Yann LeCun, Carlo Baldassi, Christian Borgs, Jennifer T. Chayes, Levent Sagun, and Riccardo Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. In *ICLR*, 2017. 1, 2, 3, 4

[4] Safa Cicek and Stefano Soatto. Input and weight space smoothing for semi-supervised learning. In *ICCV Workshops*, 2019. 1

[5] Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv.org*, abs/2010.09670, 2020. 1, 3, 4

[6] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020. 2, 3

[7] Ekin Dogus Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation policies from data. *arXiv.org*, abs/1805.09501, 2018. 2, 4

[8] Laurent Dinh, Razvan Pascanu, Samy Bengio, and Yoshua Bengio. Sharp minima can generalize for deep nets. In *ICML*, 2017. 2

[9] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *NN*, 107, 2018. 2, 4

[10] Logan Engstrom, Andrew Ilyas, Hadi Salman, Shibani Santurkar, and Dimitris Tsipras. Robustness (python library), 2019.

[11] Farzan Farnia, Jesse M. Zhang, and David Tse. Generalizable adversarial training via spectral normalization. In *ICLR*, 2019. 1

[12] Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy A. Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv.org*, abs/2010.03593, 2020. 2, 4

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 3

[14] Dan Hendrycks and Kevin Gimpel. Bridging nonlinearities and stochastic regularizers with gaussian error linear units. *arXiv.org*, abs/1606.08415, 2016. 2, 4

[15] Dan Hendrycks, Mantas Mazeika, Saurav Kadavath, and Dawn Song. Using self-supervised learning can improve model robustness and uncertainty. In *NeurIPS*, 2019. 2, 4

[16] S. Hochreiter and J. Schmidhuber. Flat minima. *NC*, 9, 1997. 2

[17] J. Hwang, Youngwan Lee, Sungchan Oh, and Yu-Seok Bae. Adversarial training with stochastic weight average. *arXiv.org*, abs/2009.10526, 2020. 1

[18] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry P. Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *UAI*, 2018. 1, 2, 4

[19] Yiding Jiang, Behnam Neyshabur, Hossein Mobahi, Dilip Krishnan, and Samy Bengio. Fantastic generalization measures and where to find them. In *ICLR*, 2020. 1

[20] 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 *ICLR*, 2017. 1, 2

[21] Marc Khoury and Dylan Hadfield-Menell. On the geometry of adversarial examples. *arXiv.org*, abs/1811.00525, 2018. 1

[22] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 3

[23] Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, 2009. 3

[24] Hao Li, Zheng Xu, G. Taylor, and T. Goldstein. Visualizing the loss landscape of neural nets. In *NeurIPS*, 2018. 1, 2, 3

[25] Tao Lin, Sebastian U. Stich, Kumar Kshitij Patel, and Martin Jaggi. Don't use large mini-batches, use local SGD. In *ICLR*, 2020. 1

[26] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *ICLR*, 2018. 1, 2

[27] Diganta Misra. Mish: A self regularized non-monotonic activation function. In *BMVC*, 2020. 2, 3, 4

[28] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nati Srebro. Exploring generalization in deep learning. In *NeurIPS*, 2017. 1, 2, 3

[29] Tianyu Pang, Xian Yang, Yinpeng Dong, Hang Su, and Jun Zhu. Bag of tricks for adversarial training. *arXiv.org*, abs/2010.00467, 2020. 2

[30] Leslie Rice, Eric Wong, and J. Zico Kolter. Overfitting in adversarially robust deep learning. In *ICML*, 2020. 1, 2, 4

[31] Ludwig Schmidt, Shibani Santurkar, Dimitris Tsipras, Kunal Talwar, and Aleksander Madry. Adversarially robust generalization requires more data. In *NeurIPS*, 2018. 1

[32] Vasu Singla, Sahil Singla, David Jacobs, and Soheil Feizi. Low curvature activations reduce overfitting in adversarial training. *arXiv.org*, abs/2102.07861, 2021. 1, 2

[33] David Stutz, Nandhini Chandramoorthy, Matthias Hein, and Bernt Schiele. Bit error robustness for energy-efficient dnn accelerators. In *MLSys*, 2021. 2, 4

[34] David Stutz, Matthias Hein, and Bernt Schiele. Relating adversarially robust generalization to flat minima. *arXiv.org*, abs/2104.04448, 2021. 2

[35] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 2, 4

[36] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014. 1

[37] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In *ICLR*, 2020. 2, 4

[38] Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. *arXiv.org*, abs/2001.03994, 2020.

[39] Dongxian Wu, Shu-Tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. In *NeurIPS*, 2020. 1, 2, 3, 4

[40] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically principled trade-off between robustness and accuracy. In *ICML*, 2019. 1, 2, 3, 4

[41] Jingfeng Zhang, Xilie Xu, Bo Han, Gang Niu, Lizhen Cui, Masashi Sugiyama, and Mohan S. Kankanhalli. Attacks which do not kill training make adversarial learning stronger. In *ICML*, 2020.

[42] Yaowei Zheng, Richong Zhang, and Yongyi Mao. Regularizing neural networks via adversarial model perturbation. *arXiv.org*, abs/2010.04925, 2020. 1