DATA AUGMENTATION CAN IMPROVE ROBUSTNESS

Sylvestre-Alvise Rebuffi*, Sven Gowal*, Dan A. Calian, Florian Stimberg, Olivia Wiles & Timothy Mann
DeepMind, London, UK
{sylvestre,sgowal}@google.com

ABSTRACT

Adversarial training suffers from *robust overfitting*, a phenomenon where the robust test accuracy starts to decrease during training. In this paper, we focus on reducing robust overfitting by using common data augmentation schemes. We demonstrate that, contrary to previous findings, when combined with model weight averaging, data augmentation can significantly boost robust accuracy. We evaluate our approach on CIFAR-10 against ℓ_∞ and ℓ_2 norm-bounded perturbations of size $\epsilon = 8/255$ and $\epsilon = 128/255$, respectively. We show large absolute improvements of +2.93% and +2.16% in robust accuracy compared to previous state-of-the-art methods. In particular, against ℓ_∞ norm-bounded perturbations of size $\epsilon = 8/255$, our model reaches 60.07% robust accuracy without using any external data.

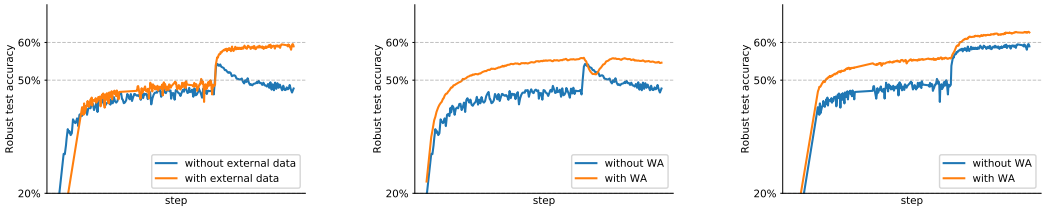
1 INTRODUCTION

Despite their success, neural networks are not intrinsically robust. In particular, it has been shown that the addition of imperceptible deviations to the input, called adversarial perturbations, can cause neural networks to make incorrect predictions with high confidence (Carlini & Wagner, 2017a;b; Goodfellow et al., 2015; Kurakin et al., 2016; Szegedy et al., 2014). Starting with Szegedy et al. (2014), there has been a lot of work on understanding and generating adversarial perturbations (Carlini & Wagner, 2017b; Athalye & Sutskever, 2018), and on building defenses that are robust to such perturbations (Goodfellow et al., 2015; Papernot et al., 2016; Madry et al., 2018; Kannan et al., 2018). Among successful defenses are robust optimization techniques like the one developed by Madry et al. (2018) that learn robust models by finding worst-case adversarial perturbations at each training step. Since Madry et al. (2018), various modifications to their original implementation have been proposed (Zhang et al., 2019; Xie et al., 2019; Pang et al., 2020; Huang et al., 2020; Rice et al., 2020; Gowal et al., 2020).

Notably, Hendrycks et al. (2019); Carmon et al. (2019); Uesato et al. (2019); Zhai et al. (2019); Najafi et al. (2019) showed that using additional data improves adversarial robustness, while Rice et al. (2020); Wu et al. (2020); Gowal et al. (2020) found that data augmentation techniques did not boost robustness. This dichotomy motivates this paper. In particular, we explore whether it is possible to fix the training procedure such that data augmentation becomes useful (in the setting without additional data). By making the observation that model weight averaging (WA) (Izmailov et al., 2018) helps robust generalization to a wider extent when robust overfitting is minimized, we propose to combine model weight averaging with data augmentation techniques. Overall, we make the following contributions:

- We demonstrate how, when combined with model weight averaging, data augmentation techniques such as *Cutout* (DeVries & Taylor, 2017), *CutMix* (Yun et al., 2019) and *MixUp* (Zhang et al., 2018) can improve robustness.
- To the contrary of Rice et al. (2020); Wu et al. (2020); Gowal et al. (2020) which all tried data augmentation techniques without success, we are able to use any of these three aforementioned techniques to obtain new state-of-the-art robust accuracies. We find *CutMix* to be the most effective method by reaching a robust accuracy of 60.07% on CIFAR-10 against ℓ_∞ perturbations of size $\epsilon = 8/255$ (an improvement of +2.93% upon the state-of-the-art).

*Equal contribution



(a) Adversarial training with and without additional external data from 80M-TI

(b) Effect of WA without external data

(c) Effect of WA with external data

Figure 1: We compare the robust accuracy against $\epsilon_\infty = 8/255$ on CIFAR-10 of an adversarially trained Wide ResNet (WRN)-28-10. Panel (a) shows the impact of using additional external data from 80M-TI (Torralba et al., 2008) and illustrates *robust overfitting*. Panel (b) shows the benefit of *model weight averaging* (WA) despite robust overfitting. Panel (c) shows that WA remains effective and useful even when robust overfitting disappears. The graphs show the evolution of the robust accuracy as training progresses (against PGD^{40}). The jump in robust accuracy two-thirds through training is due to a drop in learning rate.

2 PRELIMINARIES AND HYPOTHESIS

Adversarial training. Madry et al. (2018) formulate a saddle point problem to find model parameters $\boldsymbol{\theta}$ that minimize the adversarial risk:

$$\arg\min_{\boldsymbol{\theta}} \mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}} \left[\max_{\boldsymbol{\delta}\in\mathbb{S}} l(f(\boldsymbol{x}+\boldsymbol{\delta};\boldsymbol{\theta}),y) \right] \tag{1}$$

where $\mathcal{D}$ is a data distribution over pairs of examples $\boldsymbol{x}$ and corresponding labels y, $f(\cdot;\boldsymbol{\theta})$ is a model parametrized by $\boldsymbol{\theta}$, l is a suitable loss function (such as the $0-1$ loss in the context of classification tasks), and $\mathbb{S}$ defines the set of allowed perturbations. For ℓ_p norm-bounded perturbations of size ϵ, the adversarial set is defined as $\mathbb{S}_p = \{\boldsymbol{\delta} \mid \|\boldsymbol{\delta}\|_p \leq \epsilon\}$. In the rest of this manuscript, we will use ϵ_p to denote ℓ_p norm-bounded perturbations of size ϵ (e.g., $\epsilon_\infty = 8/255$) and for the inner optimization, we use the Projected Gradient Descent (PGD) with K steps which we refer to as PGD^K.

Robust overfitting. To the contrary of standard training, which often shows no *overfitting* in practice (Zhang et al., 2017), adversarial training suffers from *robust overfitting* (Rice et al., 2020). Robust overfitting is the phenomenon by which robust accuracy on the test set quickly degrades while it continues to rise on the train set (clean accuracy on both sets continues to improve as well). Rice et al. (2020) propose to use early stopping as the main contingency against robust overfitting, and demonstrate that it also allows to train models that are more robust than those trained with other regularization techniques (such as data augmentation or increased ℓ_2-regularization). They observed that some of these other regularization techniques could reduce the impact of overfitting at the cost of producing models that are over-regularized and lack overall robustness and accuracy. There is one notable exception which is the addition of external data (Carmon et al., 2019; Uesato et al., 2019). Fig. 1(a) shows how the robust accuracy (evaluated on the test set) evolves as training progresses on CIFAR-10 against $\epsilon_\infty = 8/255$. Without external data, robust overfitting is clearly visible and appears shortly after the learning rate is dropped (the learning rate is decayed by $10\times$ two-thirds through training in a schedule is similar to Rice et al., 2020 and commonly used since Madry et al., 2018). Robust overfitting completely disappears when an additional set of 500K pseudo-labeled images from 80M-TI (Torralba et al., 2008) is introduced.

Model weight averaging. Model weight averaging (WA) (Izmailov et al., 2018) can be implemented using an exponential moving average $\boldsymbol{\theta}'$ of the model parameters $\boldsymbol{\theta}$ with a decay rate τ (i.e., $\boldsymbol{\theta}' \leftarrow \tau \cdot \boldsymbol{\theta}' + (1-\tau)\cdot\boldsymbol{\theta}$ at each training step). During evaluation, the weighted parameters $\boldsymbol{\theta}'$ are used instead of the trained parameters $\boldsymbol{\theta}$. Gowal et al. (2020); Chen et al. (2021) discovered that model weight averaging can significantly improve robustness on a wide range of models and datasets. Chen et al. (2021) argue (similarly to Wu et al., 2020) that WA leads to a flatter adversarial

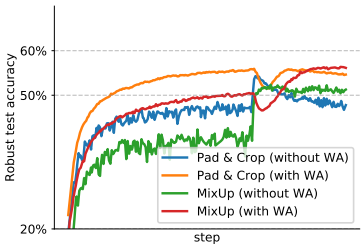


Figure 2: Accuracy against $\epsilon_\infty = 8/255$ on CIFAR-10 with and without using model weight averaging (WA) when using *MixUp* or *random padding-and-cropping* (*Pad & Crop*). The graph shows the robust test accuracy against PGD^{40}.

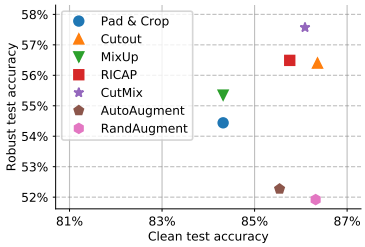


Figure 3: Clean (without adversarial attacks) accuracy and robust accuracy (against AA+MT) for a WRN-28-10 trained against $\epsilon_\infty = 8/255$ on CIFAR-10 for different data augmentation techniques.

loss landscape, and thus a smaller robust generalization gap. Gowal et al. (2020) also explain that, in addition to improved robustness, WA reduces sensitivity to early stopping. While this is true, it is important to note that WA is still prone to robust overfitting. This is not surprising, since the exponential moving average "forgets" older model parameters as training goes on. Fig. 1(b) shows how the robust accuracy evolves as training progresses when using WA. We observe that, after the change of learning rate, the averaged weights are increasingly affected by overfitting, thus resulting in worse robust accuracy for the averaged model.

Hypothesis. As WA results in flatter, wider solutions compared to the steep decrease in robust accuracy observed for Stochastic Gradient Descent (SGD) (Chen et al., 2021), it is natural to ask ourselves whether WA remains useful in cases that do not exhibit robust overfitting. Fig. 1(c) shows how the robust accuracy evolves as training progresses when using WA and additional external data (for which standard SGD does not show signs of overfitting). We notice that the robust performance in this setting is not only preserved but even boosted when using WA. Hence, we formulate the hypothesis that model weight averaging helps robustness to a greater effect when robust accuracy between model iterations can be maintained.

3 DATA AUGMENTATIONS

Limiting robust overfitting without external data. Rice et al. (2020) show that combining data augmentation methods such as *Cutout* or *MixUp* with early stopping does not improve robustness upon early stopping alone. While, these methods do not improve upon the "best" robust accuracy, they reduce the extent of robust overfitting, thus resulting in a slower decrease in robust accuracy compared to classical adversarial training (which uses random crops and weight decay). This can be seen in Fig. 2 where *MixUp* without WA exhibits no decrease in robust accuracy, whereas the robust accuracy of the standard combination of *random padding-and-cropping* without WA (*Pad & Crop*) decreases immediately after the change of learning rate.

Verifying the hypothesis. Since *MixUp* preserves robust accuracy, albeit at a lower level than the "best" obtained by *Pad & Crop*, it can be used to evaluate the hypothesis that WA is more beneficial when the performance between model iterations is maintained. Therefore, we compare in Fig. 2 the effect of WA on robustness when using *MixUp*. We observe that, when using WA, the performance of *MixUp* surpasses the performance of *Pad & Crop*. Indeed, the robust accuracy obtained by the averaged weights of *Pad & Crop* (in blue) slowly decreases after the change of learning rate, while the one obtained by *MixUp* (in green) increases throughout training. Ultimately, *MixUp* with WA obtains a higher robust accuracy despite the fact that the non-averaged model has a significantly lower "best" robust accuracy than the non-averaged *Pad & Crop* model. This finding is notable as it demonstrates for the first time the benefits of data augmentation schemes for adversarial training (this contradicts to some extent the findings from three recent publications: Rice et al., 2020; Wu et al., 2020; Gowal et al., 2020).

Setup	Pad & Crop Clean	Pad & Crop Robust	CutMix Clean	CutMix Robust
Varying the model size				
WRN-28-10	84.32%	54.44%	86.09%	**57.50%**
WRN-34-10	84.89%	55.13%	86.18%	**58.09%**
WRN-34-20	85.80%	55.69%	87.80%	**59.25%**
WRN-70-16	86.02%	57.17%	87.25%	**60.07%**

Table 1: Robust test accuracy (against AA+MT) against $\epsilon_\infty = 8/255$ on CIFAR-10 as the model size increases. We compare *Pad & Crop* and *CutMix*.

Setup	ℓ_∞ Clean	ℓ_∞ Robust	ℓ_2 Clean	ℓ_2 Robust
WRN-28-10 (if not specified)				
Wu et al. (2020) (WRN-34-10)	85.36%	56.17%	88.51%	73.66%
Gowal et al. (2020) (trained by us)	84.32%	54.44%	88.60%	72.56%
Ours (CutMix)	86.22%	**57.50%**	91.35%	**76.12%**
WRN-70-16				
Gowal et al. (2020)	85.29%	57.14%	90.90%	74.50%
Ours (CutMix)	87.25%	**60.07%**	92.43%	**76.66%**

Table 2: Clean (without adversarial attacks) accuracy and robust accuracy (against AA+MT) on CIFAR-10 as we both test against $\epsilon_\infty = 8/255$ and $\epsilon_2 = 128/255$.

Exploring data augmentations. After verifying our hypothesis for *MixUp*, we investigate in Sec. 4 if other augmentations can help maintain robust accuracy and also be combined with WA to improve robustness. We concentrate on image patching techniques like *Cutout* (DeVries & Taylor, 2017), *CutMix* (Yun et al., 2019) and *RICAP* (Takahashi et al., 2018). We also evaluate automated augmentation strategies like *AutoAugment* (Cubuk et al., 2019), *RandAugment* (Cubuk et al., 2020).

4 Experimental results

Experimental setup. In all the experiments we use model weight averaging (WA) (Izmailov et al., 2018) with a decay rate $\tau = 0.999$. All the technical details, hyperparameters, architecture and evaluation procedure are described in the appendix.

Experimental results. We consider as baseline the *Pad & Crop* augmentation which reproduces the current state-of-the-art set by Gowal et al. (2020). In Fig. 3, we compare this baseline with various data augmentations, *MixUp*, *Cutout*, *CutMix* and *RICAP*, as well as learned augmentation policies with *AutoAugment* and *RandAugment*. Three clusters are clearly visible. The first cluster, containing *AutoAugment* and *RandAugment*, increases the clean accuracy compared to the baseline but, most notably, reduces the robust accuracy. Indeed, these automated augmentation strategies have been tuned for standard classification, and should be adapted to the robust classification setting. The second cluster, containing *RICAP*, *Cutout* and *CutMix*, includes the three methods that occlude local information with patching and provide a significant boost upon the baseline with +3.06% in robust accuracy for *CutMix* and an average improvement of +1.79% in clean accuracy. The last cluster, with *MixUp*, only improves the robust accuracy upon the baseline by a small margin of +0.91%. A possible explanation lies in the fact that *MixUp* tends to either produce images that are far from the original data distribution (when α is large) or too close to the original samples (when α is small). The appendix contains more ablation analysis on all methods.

Table 1 shows the performance of *CutMix* and the *Pad & Crop* baseline when varying the model size. *CutMix* consistently outperforms the baseline by at least +2.90% in robust accuracy across all the model sizes. Table 2 shows the performance of *CutMix* on CIFAR-10 against $\epsilon_\infty = 8/255$ and $\epsilon_2 = 128/255$. We observe that using *CutMix* provides a significant boost in robust accuracy for both threat models with up to +2.93% (in the ℓ_∞ setting) and +2.16% (in the ℓ_2 setting) when training a WRN-70-16. Finally, we show the generality of our approach as using *CutMix* on CIFAR-100 significantly improves on the state-of-the-art with our best model reaching 32.43% against AutoAttack (in the ℓ_∞ setting). We refer to the appendix for more details on the CIFAR-100 experiments.

5 Conclusion

Contrary to previous works (Rice et al., 2020; Gowal et al., 2020; Wu et al., 2020), which have tried data augmentation techniques to train adversarially robust models without success, we demonstrate that combining data augmentations with model weight averaging can significantly improve robustness. Our work provides novel insights into the effect of model weight averaging on robustness, which we hope can further our understanding of robustness. All our models are available online at https://github.com/deepmind/deepmind-research/tree/master/adversarial_robustness/iclrw2021data.

REFERENCES

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square Attack: a query-efficient black-box adversarial attack via random search. *Eur. Conf. Comput. Vis.*, 2020.

Anish Athalye and Ilya Sutskever. Synthesizing robust adversarial examples. *Int. Conf. Mach. Learn.*, 2018.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. *Int. Conf. Mach. Learn.*, 2018.

Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pp. 3–14. ACM, 2017a.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy*, 2017b.

Yair Carmon, Aditi Raghunathan, Ludwig Schmidt, John C Duchi, and Percy S Liang. Unlabeled data improves adversarial robustness. In *Adv. Neural Inform. Process. Syst.*, 2019.

Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Robust overfitting may be mitigated by properly learned smoothening. In *Int. Conf. Learn. Represent.*, 2021. URL https://openreview.net/pdf?id=qZzy5urZw9.

Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. *arXiv preprint arXiv:1907.02044*, 2020a. URL https://arxiv.org/pdf/1907.02044.

Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. *arXiv preprint arXiv:2003.01690*, 2020b.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

Ekin D. Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V. Le. Randaugment: Practical automated data augmentation with a reduced search space. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

Jiequan Cui, Shu Liu, Liwei Wang, and Jiaya Jia. Learnable boundary guided adversarial training. *arXiv preprint arXiv:2011.11164*, 2020. URL https://arxiv.org/pdf/2011.11164.

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

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting Adversarial Attacks with Momentum. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *Int. Conf. Learn. Represent.*, 2015.

Sven Gowal, Jonathan Uesato, Chongli Qin, Po-Sen Huang, Timothy Mann, and Pushmeet Kohli. An Alternative Surrogate Loss for PGD-based Adversarial Testing. *arXiv preprint arXiv:1910.09338*, 2019.

Sven Gowal, Chongli Qin, Jonathan Uesato, Timothy Mann, and Pushmeet Kohli. Uncovering the limits of adversarial training against norm-bounded adversarial examples. *arXiv preprint arXiv:2010.03593*, 2020. URL https://arxiv.org/pdf/2010.03593.

Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016.

Dan Hendrycks, Kimin Lee, and Mantas Mazeika. Using Pre-Training Can Improve Model Robustness and Uncertainty. *Int. Conf. Mach. Learn.*, 2019.

Lang Huang, Chao Zhang, and Hongyang Zhang. Self-Adaptive Training: beyond Empirical Risk Minimization. *arXiv preprint arXiv:2002.10319*, 2020.

Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging Weights Leads to Wider Optima and Better Generalization. *Uncertainty in Artificial Intelligence*, 2018.

Harini Kannan, Alexey Kurakin, and Ian Goodfellow. Adversarial Logit Pairing. *arXiv preprint arXiv:1803.06373*, 2018.

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

Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *ICLR workshop*, 2016.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *Int. Conf. Learn. Represent.*, 2018.

Marius Mosbach, Maksym Andriushchenko, Thomas Trost, Matthias Hein, and Dietrich Klakow. Logit Pairing Methods Can Fool Gradient-Based Attacks. *arXiv preprint arXiv:1810.12042*, 2018.

Amir Najafi, Shin-ichi Maeda, Masanori Koyama, and Takeru Miyato. Robustness to adversarial perturbations in learning from incomplete data. *Adv. Neural Inform. Process. Syst.*, 2019.

Yurii Nesterov. A method of solving a convex programming problem with convergence rate $o(1/k^2)$. In *Sov. Math. Dokl*, 1983.

Tianyu Pang, Xiao Yang, Yinpeng Dong, Kun Xu, Hang Su, and Jun Zhu. Boosting Adversarial Training with Hypersphere Embedding. *Adv. Neural Inform. Process. Syst.*, 2020.

Nicolas Papernot, Patrick McDaniel, Xi Wu, Somesh Jha, and Ananthram Swami. Distillation as a defense to adversarial perturbations against deep neural networks. *IEEE Symposium on Security and Privacy*, 2016.

Boris T Polyak. Some methods of speeding up the convergence of iteration methods. *USSR Computational Mathematics and Mathematical Physics*, 1964.

Leslie Rice, Eric Wong, and J. Zico Kolter. Overfitting in adversarially robust deep learning. *Int. Conf. Mach. Learn.*, 2020.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *Int. Conf. Learn. Represent.*, 2014.

Ryo Takahashi, Takashi Matsubara, and Kuniaki Uehara. Ricap: Random image cropping and patching data augmentation for deep cnns. *Asian Conf. Mach. Learn.*, 2018.

Antonio Torralba, Rob Fergus, and William T. Freeman. 80 million tiny images: a large dataset for non-parametric object and scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 2008.

Jonathan Uesato, Brendan O'Donoghue, Aaron van den Oord, and Pushmeet Kohli. Adversarial Risk and the Dangers of Evaluating Against Weak Attacks. *Int. Conf. Mach. Learn.*, 2018.

Jonathan Uesato, Jean-Baptiste Alayrac, Po-Sen Huang, Robert Stanforth, Alhussein Fawzi, and Pushmeet Kohli. Are labels required for improving adversarial robustness? *Adv. Neural Inform. Process. Syst.*, 2019.

Dongxian Wu, Shu-tao Xia, and Yisen Wang. Adversarial weight perturbation helps robust generalization. *Adv. Neural Inform. Process. Syst.*, 2020.

Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. *Int. Conf. Comput. Vis.*, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *Brit. Mach. Vis. Conf.*, 2016.

Runtian Zhai, Tianle Cai, Di He, Chen Dan, Kun He, John Hopcroft, and Liwei Wang. Adversarially Robust Generalization Just Requires More Unlabeled Data. *arXiv preprint arXiv:1906.00555*, 2019.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *Int. Conf. Learn. Represent.*, 2017. URL https://openreview.net/pdf?id=Sy8gdB9xx.

Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric P. Xing, Laurent El Ghaoui, and Michael I. Jordan. Theoretically Principled Trade-off between Robustness and Accuracy. *Int. Conf. Mach. Learn.*, 2019.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *Int. Conf. Learn. Represent.*, 2018.

Data Augmentation Can Improve Robustness (Supplementary Material)

A Related work

Adversarial training. The adversarial training procedure (Madry et al., 2018) feeds adversarially perturbed examples back into the training data. It has been augmented in different ways – with changes in the attack procedure (e.g., by incorporating momentum; Dong et al., 2018), loss function (e.g., logit pairing; Mosbach et al., 2018) or model architecture (e.g., feature denoising; Xie et al., 2019). Another notable work by Zhang et al. (2019) proposed TRADES, which balances the trade-off between standard and robust accuracy, and achieved state-of-the-art performance against ℓ_∞ norm-bounded perturbations on CIFAR-10. More recently, the work from Rice et al. (2020) studied *robust overfitting* and demonstrated that improvements similar to TRADES could be obtained more easily using classical adversarial training with early stopping. This later study revealed that early stopping was competitive with many other regularization techniques and demonstrated that data augmentation schemes beyond the typical *random padding-and-cropping* were ineffective on CIFAR-10. Finally, Gowal et al. (2020) highlighted how different hyper-parameters (such as network size and model weight averaging) affect robustness. They were able to obtain models that significantly improved upon the state-of-the-art, but lacked a thorough investigation on data augmentation schemes. Similarly to Rice et al. (2020), they also make the conclusion that data augmentations beyond *random padding-and-cropping* do not improve robustness.

Data augmentation. Data augmentation has been shown to reduce the generalization error of standard (non-robust) training. For image classification tasks, random flips, rotations and crops are commonly used He et al. (2016). More sophisticated techniques such as *Cutout* (DeVries & Taylor, 2017) (which produces random occlusions), *CutMix* (Yun et al., 2019) (which replaces parts of an image with another) and *MixUp* (Zhang et al., 2018) (which linearly interpolates between two images) all demonstrate extremely compelling results. As such, it is rather surprising that they remain ineffective when training adversarially robust networks.

B Experimental setup

Architecture. We use WRNs (He et al., 2016; Zagoruyko & Komodakis, 2016) as our backbone network. This is consistent with prior work (Madry et al., 2018; Rice et al., 2020; Zhang et al., 2019; Uesato et al., 2019; Gowal et al., 2020) which use diverse variants of this network family. Furthermore, we adopt the same architecture details as Gowal et al. (2020) with Swish/SiLU (Hendrycks & Gimpel, 2016) activation functions. Most of the experiments are conducted on a WRN-28-10 model which has a depth of 28, a width multiplier of 10 and contains 36M parameters. To evaluate the effect of data augmentations on wider and deeper networks, we also run several experiments using WRN-70-16, which contains 267M parameters.

Outer minimization. We use TRADES (Zhang et al., 2019) optimized using SGD with Nesterov momentum (Polyak, 1964; Nesterov, 1983) and a global weight decay of 5×10^{-4}. We train for 400 epochs with a batch size of 512, and the learning rate is initially set to 0.1 and decayed by a factor 10 two-thirds-of-the-way through training. We scale the learning rates using the linear scaling rule of Goyal et al. (2017) (i.e., effective LR $= \max(\text{LR} \times \text{batch size}/256, \text{LR})$). We also use model weight averaging (WA) (Izmailov et al., 2018). The decay rate of WA is set to $\tau = 0.999$.

Inner minimization. Adversarial examples are obtained by maximizing the Kullback-Leibler divergence between the predictions made on clean inputs and those made on adversarial inputs (Zhang et al., 2019). This optimization procedure is done using the Adam optimizer (Kingma & Ba, 2014) for 10 PGD steps. We take an initial step-size of 0.1 which is then decreased to 0.01 after 5 steps.

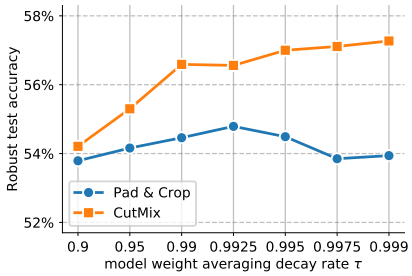


Figure 4: Robust test accuracy against AA+MT with $\epsilon_\infty = 8/255$ on CIFAR-10 as we vary the decay rate of the model weight averaging. The model is a WRN-28-10, which is trained either with *CutMix* or *Pad & Crop*.

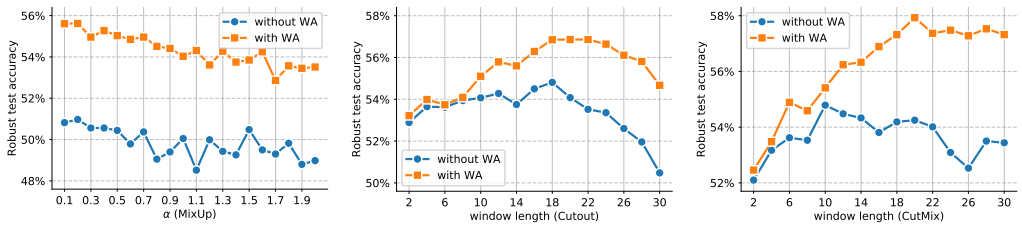
Evaluation. We follow the evaluation protocol designed by Gowal et al. (2020). Specifically, we train two (and only two) models for each hyperparameter setting, perform early stopping for each model on a separate validation set of 1024 samples using PGD^{40} similarly to Rice et al. (2020) and pick the best model by evaluating the robust accuracy on the same validation set . Finally, we report the robust test accuracy against a mixture of AUTOATTACK (Croce & Hein, 2020b) and MULTITARGETED (Gowal et al., 2019), which is denoted by AA+MT. This mixture consists in completing the following sequence of attacks: AUTOPGD on the cross-entropy loss with 5 restarts and 100 steps, AUTOPGD on the difference of logits ratio loss with 5 restarts and 100 steps and finally MULTITARGETED on the margin loss with 10 restarts and 200 steps. The training curves, such as those visible in Fig. 1, are always computed using PGD with 40 steps and the Adam optimizer (with step-size decayed by $10\times$ at step 20 and 30).

C ADDITIONAL EXPERIMENTS

Model weight averaging decay rate. In Fig. 4, we run an ablation study measuring the robust accuracy obtained when varying the decay rate τ of model weight averaging (WA) and using either *Pad & Crop* or *CutMix*. When using *CutMix*, the best robust accuracy is obtained at the highest decay rate $\tau = 0.999$. When using *Pad & Crop*, it is only obtained at a lower decay rate $\tau = 0.9925$. This is consistent with our observation from Sec. 3 that highlights how WA improves robustness to a greater extent when robust accuracy can be maintained throughout training. As larger decay rates average over longer time spans, they should better exploit the fact that *CutMix* maintains robust accuracy after the learning rate is dropped to the contrary of *Pad & Crop* (see Fig. 6).

Mixing rate of *MixUp*. For completeness, we also vary the different hyper-parameters that define the different data augmentations. In particular, for *MixUp*, we vary the mixing rate α. Remember that *MixUp* blends images by sampling an interpolation point $\lambda \sim \beta(\alpha, \alpha)$ from a Beta distribution with both its parameters set to α. Small values of α produce images near the original images, while larger values tend to blend images equally. In Fig. 5(a), we observe that smaller values of α are preferential (irrespective of whether we use model weight averaging). This conclusion is in line with the recommended settings from Zhang et al. (2018) for standard training, but contradicts the experiments made by Rice et al. (2020) who recommend a value of $\alpha = 1.4$ for robust training. We also note that using model weight averaging can increase robust accuracy by up to +5.79% when using *MixUp*.

Window length of *Cutout*. *CutOut* creates random occlusions (i.e., anywhere in the original image) of a fixed size (measured in pixels). Remember that CIFAR-10 images have a size of 32×32 pixels. The size of this occlusion is controlled by a parameter called the *window length*. Fig. 5(b) shows how the robust accuracy varies as a result of changing this parameter. We notice that the optimal window length is at 18 pixels whether model weight averaging (WA) is used or not. While WA is useful, it is noticeably less powerful when using *CutOut* (as opposed to *MixUp* and *CutMix*) bringing only an improvement of +2.05% in robust accuracy. This is clearly explained by the training curves shown in



(a) (b) (c)

Figure 5: Robust test accuracy against AA+MT with $\epsilon_\infty = 8/255$ on CIFAR-10 as we vary (a) the mixing rate α of *MixUp*, (b) the window length when using *CutOut* and (c) the window length when using *CutMix*. The model is a WRN-28-10 and we compare the settings without and with model weight averaging (in which case, we use $\tau = 0.999$). As a reference, the same model trained with *Pad & Crop* and model weight averaging reaches 54.44% robust accuracy.

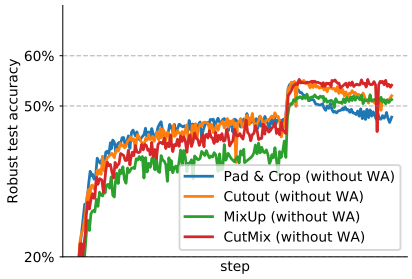


Figure 6: Accuracy against $\epsilon_\infty = 8/255$ on CIFAR-10 without using model weight averaging (WA) for different data augmentation schemes. The model is a WRN-28-10 and the curves show the evolution of the robust accuracy as training progresses (against PGD^{40}). The jump in robust accuracy two-thirds through training is due to a drop in learning rate.

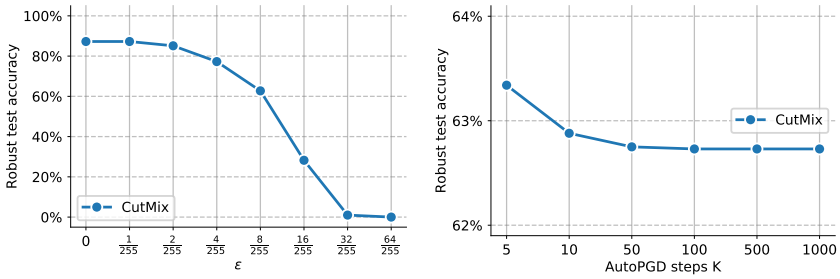
MODEL	CLEAN	AA+MT	AA
Cui et al. (2020) (WRN-34-10)	60.64%	–	29.33%
WRN-28-10 (retrained)	59.05%	28.75%	–
WRN-28-10 (CutMix)	62.97%	**30.50%**	**29.80%**
Gowal et al. (2020) (WRN-70-16)	60.86%	30.67%	30.03%
WRN-70-16 (retrained)	59.65%	30.62%	–
WRN-70-16 (CutMix)	65.76%	**33.24%**	**32.43%**

Table 3: Clean (without adversarial attacks) accuracy and robust accuracy (AA+MT) on CIFAR-100 against $\epsilon_\infty = 8/255$ obtained by different models. Robust accuracy against AUTOATTACK is also reported for select models.

Fig. 6 that demonstrate that *CutOut* suffers from *robust overfitting*. It also provides further evidence that support our hypothesis in Sec. 2.

Window length of *CutMix*. *CutMix* patches a rectangular cutout from one image onto another. In Yun et al. (2019), the area of this patch is sampled uniformly at random (this is the setting used throughout this paper). In this ablation experiment, however, we fix its size (i.e., window length) and observe its effect on robustness. In Fig. 5(c), we observe that the optimal size is not the same depending on whether model weight averaging (WA) is used. We also note that WA improves robust accuracy by +3.14%. Overall, *CutMix* obtains the highest robust accuracy of any of the four considered augmentations (including *MixUp*, *CutOut* and *Pad & Crop*).

CIFAR-100. Finally, to evaluate the generality of our approach, we evaluate *CutMix* on CIFAR-100. The results are shown in Table 3. Our best model reaches 32.43% against AUTOATTACK and improves noticeably on the state-of-the-art (in the setting that does not use any external data). It is worth noting that the currently best known result on CIFAR-100 against $\epsilon_\infty = 8/255$ when using external data is 36.88% against AUTOATTACK.



(a) (b)

Figure 7: Robust test accuracy measured by running AUTOPGD-CE with (a) different radii ϵ_∞ and (b) different number of steps K. The model is a WRN-70-16 network trained with *CutMix* against $\epsilon_\infty = 8/255$, which obtains 60.07% robust accuracy against AA+MT at $\epsilon_\infty = 8/255$.

D ANALYSIS OF MODELS

In this section, we perform additional diagnostics that give us confidence that our models are not doing any form of gradient obfuscation or masking (Athalye et al., 2018; Uesato et al., 2018).

AUTOATTACK and robustness against black-box attacks. First, we report in Table 4 the robust accuracy obtained by our strongest models against a diverse set of attacks. These attacks are run as a cascade using the AUTOATTACK library available at https://github.com/fra31/auto-attack. The cascade is composed as follows:

- AUTOPGD-CE, an untargeted attack using PGD with an adaptive step on the cross-entropy loss (Croce & Hein, 2020b),
- AUTOPGD-T, a targeted attack using PGD with an adaptive step on the difference of logits ratio (Croce & Hein, 2020b),
- FAB-T, a targeted attack which minimizes the norm of adversarial perturbations (Croce & Hein, 2020a),
- SQUARE, a query-efficient black-box attack (Andriushchenko et al., 2020).

First, we observe that our combination of attacks, denoted AA+MT matches the final robust accuracy measured by AUTOATTACK. Second, we also notice that the black-box attack (i.e., SQUARE) does not find any additional adversarial examples. Overall, these results indicate that our empirical measurement of robustness is meaningful and that our models do not obfuscate gradients.

MODEL	NORM	RADIUS	AUTOPGD-CE	+ AUTOPGD-T	+ FAB-T	+ SQUARE	CLEAN	AA+MT
WRN-28-10 (CutMix)	ℓ_∞	$\epsilon = 8/255$	61.01%	57.61%	57.61%	57.61%	86.22%	57.50%
WRN-70-16 (CutMix)			62.65%	60.07%	60.07%	60.07%	87.25%	60.07%

Table 4: Clean (without adversarial attacks) accuracy and robust accuracy (against the different stages of AUTOATTACK) on CIFAR-10 obtained by different models. Refer to https://github.com/fra31/auto-attack for more details.

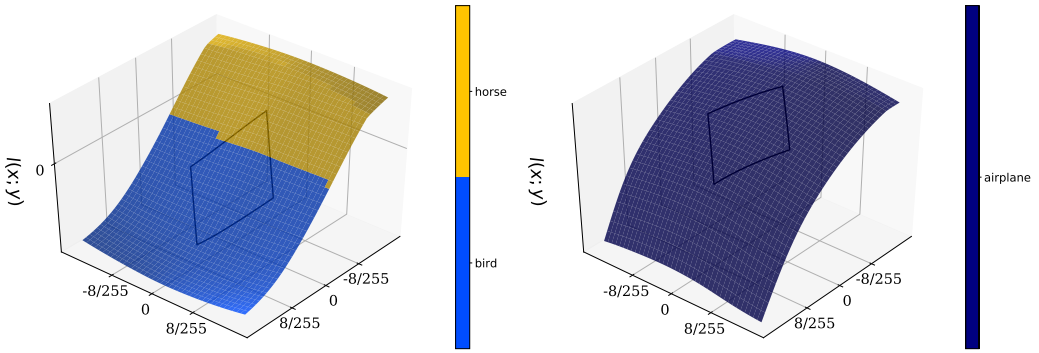
Further analysis of gradient obfuscation. In this paragraph, we consider a WRN-70-16 trained with *CutMix* against $\epsilon_\infty = 8/255$, which obtains 60.07% robust accuracy against AA+MT at $\epsilon_\infty = 8/255$.

In Fig. 7(a), we run AUTOPGD-CE with 100 steps and 1 restart and we vary the perturbation radius ϵ_∞ between zero and $64/255$. As expected, the robust accuracy gradually drops as the radius increases indicating that PGD-based attacks can find adversarial examples and are not hindered by gradient obfuscation.

In Fig. 7(b), we run AUTOPGD-CE with $\epsilon_\infty = 8/255$ and 1 restart and vary the number of steps K between five and 1000. We observe that the measured robust accuracy converges after 50 steps. This is further indication that attacks converge in 100 steps.

Loss landscapes. Finally, we analyze the adversarial loss landscapes of the model considered in the previous paragraph. To generate a loss landscape, we vary the network input along the linear space defined by the worse perturbation found by PGD^{40} (u direction) and a random Rademacher direction (v direction). The u and v axes represent the magnitude of the perturbation added in each of these directions respectively and the z axis is the adversarial margin loss (Carlini & Wagner, 2017b): $z_y - \max_{i \neq y} z_i$ (i.e., a misclassification occurs when this value falls below zero).

Fig. 8 shows the loss landscapes around the first 2 images of the CIFAR-10 test set. All landscapes are smooth and do not exhibit patterns of gradient obfuscation. Overall, it is difficult to interpret these figures further, but they do complement the numerical analyses done so far.



(a) Image of a horse

(b) Image of an airplane

Figure 8: Loss landscapes around the first two images from the CIFAR-10 test set for the WRN-70-16 network trained with *CutMix*. This model obtains 60.07% robust accuracy. It is generated by varying the input to the model, starting from the original input image toward either the worst attack found using PGD^{40} (u direction) or a random Rademacher direction (v direction). The loss used for these plots is the margin loss $z_y - \max_{i \neq y} z_i$ (i.e., a misclassification occurs when this value falls below zero). The diamond-shape represents the projected ℓ_∞ ball of size $\epsilon = 8/255$ around the nominal image.