DEFENDING AGAINST IMAGE CORRUPTIONS THROUGH ADVERSARIAL AUGMENTATIONS

Dan A. Calian **Florian Stimberg** **Sylvestre-Alvise Rebuffi** **Olivia Wiles**
András György **Timothy Mann** **Sven Gowal**
DeepMind
{dancalian,stimberg,sylvestre,oawiles,agyorgy,timothymann,sgowal}@google.com

ABSTRACT

Modern neural networks excel at image classification, achieving superhuman performance, yet they remain vulnerable to common image corruptions such as blur, speckle noise or fog. Recent methods that focus on this problem, such as *AugMix* and *DeepAugment*, introduce defenses that operate *in expectation* over a distribution of image corruptions. In contrast, the literature on ℓ_p-norm bounded perturbations focuses on defenses against *worst-case* corruptions. In this work, we reconcile both approaches by proposing *AdversarialAugment*, a technique which optimizes the parameters of image-to-image models to generate adversarially corrupted augmented images. Our classifiers improve upon the state-of-the-art on common image corruption benchmarks conducted in expectation on CIFAR-10-C and improve worst-case performance against ℓ_p-norm bounded perturbations.

1 INTRODUCTION

By following a process known as Empirical Risk Minimization (ERM) (Vapnik, 1998), neural networks are trained to minimize the average error on a training set. ERM has enabled breakthroughs in a wide variety of fields and applications (Goodfellow et al., 2016; Krizhevsky et al., 2012; Hinton et al., 2012), ranging from ranking content on the web (Covington et al., 2016) to autonomous driving (Bojarski et al., 2016) via medical diagnostics (De Fauw et al., 2018). ERM is based on the principle that the data used during training is independently drawn from the same distribution as the one encountered during deployment. In practice, however, training and deployment data may differ and models can fail catastrophically. Such occurrence is commonplace as training data is often collected through a biased process that highlights confounding factors and spurious correlations (Torralba et al., 2011; Kuehlkamp et al., 2017), which can lead to undesirable consequences (e.g., `http://gendershades.org`).

As such, it has become increasingly important to ensure that deployed models are robust and generalize to various input corruptions. Unfortunately, even small corruptions can significantly affect the performance of existing classifiers. For example, Recht et al. (2019); Hendrycks et al. (2019) show that the accuracy of IMAGENET models is severely impacted by changes in the data collection process, while 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). Methods to counteract such effects, which mainly consist of using random or adversarially-chosen *data augmentations*, struggle. Training against corrupted data only forces the memorization of such corruptions and, as a result, these models fail to generalize to new corruptions (Vasiljevic et al., 2016; Geirhos et al., 2018). For an extended discussion of related work please see section B in the appendix.

Recent work from Hendrycks et al. (2020b) (also known as *AugMix*) argues that basic corruptions can be composed to improve the robustness of models to common corruptions. While the method performs well in expectation on the common corruptions present in CIFAR-10-C and IMAGENET-C, it generalizes poorly to the adversarial setting. Most recently, Laidlaw et al. (2021) proposed an adversarial training method based on bounding a neural perceptual distance (i.e., an approximation of the true perceptual distance), under the acronym of PAT for Perceptual Adversarial Training.

Their method performs well against five diverse adversarial attacks, but, as it specifically addresses robustness to pixel-level attacks that directly manipulate image pixels, it performs worse than *AugMix* on common corruptions. In this work, we address this gap. We focus on training models that are robust to adversarially-chosen corruptions that preserve semantic content. We go beyond conventional *random data augmentation* schemes (exemplified by Hendrycks et al., 2020b;a) and *adversarial training* (exemplified by Madry et al., 2018; Gowal et al., 2019; Laidlaw et al., 2021) by leveraging image-to-image models that can produce a wide range of semantically-preserving corruptions. Our contributions are as follows:

- We formulate an adversarial training procedure, named *AdversarialAugment* (or *AdA* for short), which is based on *DeepAugment* (Hendrycks et al., 2020a). To the contrary of *DeepAugment*, our approach generalizes to adversarial settings while maintaining accuracy on common corruptions in expectation. We also establish links to Invariant Risk Minimization (IRM) (Arjovsky et al., 2020), Adversarial Mixing (*AdvMix*) (Gowal et al., 2019) and Perceptual Adversarial Training (PAT) (Laidlaw et al., 2021)
- We improve upon the known state-of-the-art on CIFAR-10-C by achieving a mean corruption error (mCE) of **7.83%** when using our method *in conjunction* with others (vs. 23.51% for PAT, 10.90% for AugMix and 8.11% for DeepAugment). On ℓ_2 and ℓ_∞ norm-bounded perturbations we significantly improve upon *DeepAugment* and *AugMix*. Our method (coupled with *AugMix*) also slightly improves generalization performance on CIFAR-10.1.

2 Defense Against Adversarial Corruptions

In this section, we introduce *AdA*, our approach for training models robust to image corruptions through the use of adversarial augmentations. For an explanation of how our work relates to IRM (Arjovsky et al., 2020), *AdvMix* (Gowal et al., 2019) and PAT (Laidlaw et al., 2021), see section C in the appendix.

Corrupted adversarial risk. We consider a model $f_\theta : \mathcal{X} \to \mathcal{Y}$ parametrized by θ. Given a data distribution $\mathcal{D} \subset \mathcal{X} \times \mathcal{Y}$ over pairs of examples x and corresponding labels y, we would like to find the parameters θ which minimize the *corrupted adversarial risk*:

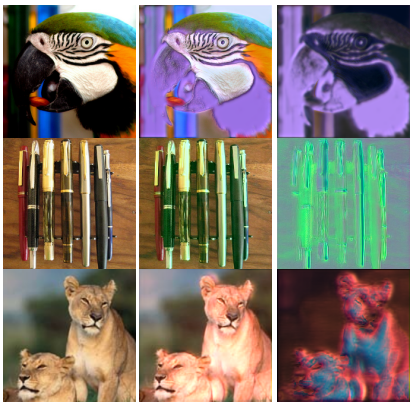
$$\mathbb{E}_{(x,y)\sim\mathcal{D}}\Big[\max_{x'\in\mathcal{C}(x)} L(f_\theta(x'), y)\Big] \tag{1}$$

where L is a suitable loss function, e.g. the 0-1 loss for classification, and $\mathcal{C} : \mathcal{X} \to 2^{\mathcal{X}}$ outputs a corruption set for a given example x. For example, in the case of an image x, a plausible corruption set $\mathcal{C}(x)$ could contain blurred, pixelized and noisy variants of x.

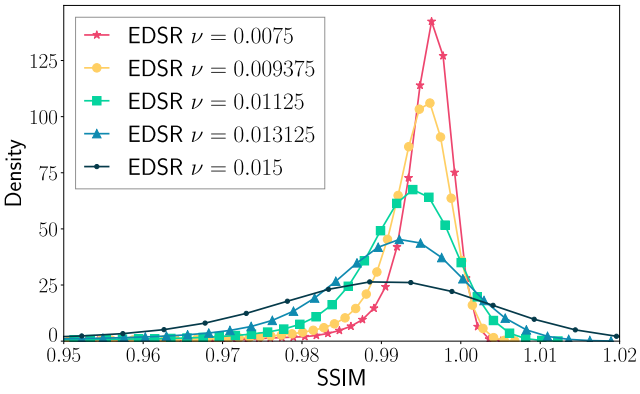
In other words, we seek the optimal parameters θ^* which minimize the corrupted adversarial risk so that f_{θ^*} is invariant to corruptions, i.e. $\forall_{x'\in\mathcal{C}(x)} :\ f_{\theta^*}(x') = f_{\theta^*}(x)$. For example if x is an image classified to be a horse by f_{θ^*}, then this prediction should not be affected by the image being slightly corrupted by camera blur, Poisson noise or JPEG compression artifacts.

AdversarialAugment (AdA). Inspired by *DeepAugment*, we use image-to-image models to generate augmented corrupted images. However, instead of making use of heuristically-defined editing operations, we optimize over the parameters of the image-to-image models directly. We dub these image-to-image models *corruption networks*. We experiment with both the super-resolution EDSR model (Lim et al., 2017) and the compressive autoencoder (CAE) model (Theis et al., 2017), the same ones used in *DeepAugment*. Since we do not observe any improvements in performance when using *CAE* on CIFAR-10-C, we only use *EDSR* in the remainder of this paper.

Formally, let $c_\phi : \mathcal{X} \to \mathcal{X}$ be a *corruption network* with parameters ϕ which acts upon clean examples by corrupting them. Let β be a weight perturbation, so that a corrupted variant of x can be generated by $c_{\phi+\beta}(x)$. Clearly, using unconstrained perturbations can result in exceedingly corrupted images which have lost all discriminative information and are not useful for training. For example, if c_ϕ is a multi-layer perceptron, trivially setting $\beta = -\phi$ would yield fully-zero, uninformative, outputs. Hence, we restrict the corruption sets by defining a maximum perturbation radius $\nu > 0$. We denote *AdA*'s corruption set by $\mathcal{C}(x) = \{c_{\phi+\beta}(x) \mid \|\beta\|_2 \leq \nu\}$. In practice, as c_ϕ is a deep neural network, we set a global absolute perturbation radius which is converted into relative perturbation radii for the weights and biases (separately) of each network layer independently. For example, if ϕ_i are the i-th layer parameters, then the effective perturbation radius of these parameters is $\nu_i = \nu \cdot \|\phi_i\|_2$.



(a) Adversarial examples using *Ada*.



(b) Distributions of SSIM scores.

Figure 1: The left plot (a) shows adversarial examples generated using the *AdA* attack for a nominally trained *ResNet50* through the *EDSR* backbone (the image pairs show the original, adversarial example and exaggerated differences respectively). In this case *AdA* introduces local and global color shifts while preserving high-frequency details. The right plot (b) shows the (estimated using KDE) distribution of SSIM scores under different perturbation radii. The SSIM distribution with a small perturbation radius ($\nu = 0.0075$) is highly concentrated and close to 1.0 SSIM yielding very close to clean images; increasing ν slightly, dissipates density rapidly.

Finding adversarial corruptions. For a clean image x with label y, a corrupted adversarial example within a bounded corruption distance ν is a corrupted image $x' = c_{\phi+\beta}(x)$ generated by the corruption network c with bounded parameter offsets $||\beta||_2 \leq \nu$ which causes f_θ to misclassify x: $f_\theta(x') \neq y$. Similarly to Madry et al. (2018), we find an adversarial corruption by maximizing a surrogate loss $\tilde{L}$ to L, e.g., the cross-entropy loss between the corrupted image predicted logits and its clean label. We optimize over the perturbation β to c's parameters ϕ:

$$\max_{\|\beta\|_2 \leq \nu} \tilde{L}(f_\theta(c_{\phi+\beta}(x)), y). \tag{2}$$

We solve this optimization problem using projected gradient ascent to enforce that perturbations β lie within the feasible set $\|\beta\|_2 \leq \nu$. Examples of corrupted images obtained by *AdA* are shown in Figure 1 (a). Note that the pre-trained corruption network parameters (ϕ) are kept fixed and we only optimize over their perturbation (β).

Adversarial training. Given the model f parametrized by θ, minimizing the corrupted adversarial risk from Equation 1 results in parameters θ^* obtained by solving the following saddle point problem:

$$\theta^* = \arg\min_\theta \mathbb{E}_{(x,y)\sim\mathcal{D}} \Big[\max_{\|\beta\|_2 \leq \nu} \tilde{L}(f_\theta(c_{\phi+\beta}(x)), y) \Big]. \tag{3}$$

Meaningful corruptions. A crucial element of *AdA* is setting the perturbation radius ν to ensure that corruptions are varied enough to constitute a strong defense against common corruptions, while still being meaningful (i.e., without destroying semantics). We measure the extent of corruption induced by a given ν through the structural similarity index measure (SSIM) (Wang et al., 2004) between clean and corrupted images. We plot the distributions of SSIM over various perturbation radii in Figure 1 (b). We find that a perturbation radius of $\nu = .015$ yields enough corruptions variety (having large SSIM variance compared to, e.g., $\nu = 0.009375$) without destroying semantic meaning (having a high mean SSIM of .99). We guard against very unlikely severe corruptions which have SSIM less than 0.3 using an efficient approximate line search procedure (see Appendix G for details).

3 Results

We evaluate the robustness of models trained with *AdA*. Table 1 shows performance on robustness to common image corruptions on CIFAR-10-C. The CIFAR-10-C benchmark (Hendrycks & Dietterich, 2019) contains corrupted variants of the CIFAR-10 test set using 15 synthetic corruptions (including various manipulations as shown in the header of Table 1) at five levels of severity. Table 2 shows performance on robustness to ℓ_p-norm bounded perturbations and, for completeness, on CIFAR-10.1.

Table 1: **Robustness to common image corruptions (CIFAR-10-C).** The table shows mean corruption error on CIFAR-10-C and individual corruption errors for each corruption type (averaged across all severities). We note that AugMix with an additional regularizing loss (JSD) and a ResNeXt architecture obtains 10.90% mCE (Hendrycks et al., 2020b) and sets the known state-of-the-art mCE on CIFAR-10-C prior to our result herein.

		Noise			Blur				Weather				Digital			
Setup	mCE	Gauss	Shot	Impulse	Defocus	Glass	Motion	Zoom	Snow	Frost	Fog	Bright	Contrast	Elastic	Pixel	JPEG
AdA + DeepAugment (10×) + AugMix	**7.83%**	8.8	7.8	11.2	5.9	**10.7**	**7.3**	6.5	8.5	6.7	8.7	5.2	6.2	8.5	7.7	**7.8**
AdA + DeepAugment + AugMix	8.58%	9.8	8.7	12.4	6.3	11.6	8.0	6.8	9.4	7.5	8.8	5.4	6.6	9.2	9.7	8.6
AdA + AugMix	8.80%	16.4	12.8	14.4	5.7	10.9	7.5	6.3	**7.7**	6.7	6.6	**3.9**	**4.7**	**8.1**	11.9	8.4
AdA	12.49%	25.8	19.8	29.9	9.3	15.6	10.9	9.6	9.3	8.1	8.8	4.3	11.0	9.3	**6.7**	9.2
DeepAugment	11.67%	12.1	10.2	15.2	7.1	20.5	12.2	8.4	12.3	9.8	9.3	6.7	13.1	11.8	14.1	12.2
DeepAugment + AugMix	10.15%	10.9	9.4	10.8	7.0	18.2	10.4	7.7	10.6	8.7	9.3	6.3	8.6	11.4	12.3	10.7
DeepAugment (10×)	8.30%	9.0	**7.6**	11.5	**5.4**	14.0	7.9	**6.1**	8.6	**6.6**	7.1	5.0	7.3	9.1	10.0	9.4
DeepAugment (10×) + AugMix	8.11%	**8.8**	7.7	**8.6**	5.7	14.4	7.9	6.2	8.4	6.7	7.3	5.1	6.5	9.2	10.4	8.9
AugMix	13.13%	26.2	18.3	10.6	7.2	32.8	10.4	9.0	10.5	10.1	**6.3**	4.7	6.3	11.4	20.1	13.0
Nominal	25.17%	58.3	44.6	49.1	15.4	44.7	19.7	19.4	15.6	17.9	10.0	5.3	17.2	15.3	24.7	19.8

Table 2: **Robustness to adversarial ℓ_p-norm bounded perturbations and distribution shift (CIFAR-10).** The table shows clean and robust top-1 accuracy against ℓ_∞ and ℓ_2 attacks on CIFAR-10, and clean accuracy on CIFAR-10.1 (trained on the CIFAR-10 training set only).

			ℓ_2		ℓ_∞		
Setup	Clean	CIFAR-10.1	$\epsilon = 0.5$	$\epsilon = 1.0$	$\epsilon = 1/255$	$\epsilon = 2/255$	$\epsilon = 4/255$
AdA + DeepAugment (10×) + AugMix	94.93%	87.35%	**18.63%**	**0.99%**	77.72%	**49.95%**	**13.88%**
AdA + DeepAugment + AugMix	94.72%	88.20%	13.25%	0.30%	75.23%	44.78%	9.68%
AdA + AugMix	**96.26%**	**91.05%**	2.26%	0.00%	68.32%	26.45%	0.99%
AdA	96.18%	90.50%	8.61%	0.03%	**77.78%**	44.39%	5.89%
DeepAugment	94.27%	87.10%	0.55%	0.00%	55.18%	14.43%	0.28%
DeepAugment + AugMix	94.25%	87.60%	0.56%	0.00%	49.63%	12.86%	0.15%
DeepAugment (10×)	95.57%	88.75%	1.56%	0.00%	64.72%	21.35%	0.70%
DeepAugment (10×) + AugMix	95.21%	89.35%	1.16%	0.00%	55.58%	17.07%	0.49%
AugMix	96.05%	90.40%	0.00%	0.00%	28.48%	1.53%	0.00%
Nominal	95.77%	89.65%	0.00%	0.00%	24.23%	0.85%	0.00%

Experimental setup. We train pre-activation *ResNet50* (He et al., 2016b) models (as in Wong et al. (2020)) on the clean training set of CIFAR-10; see Appendix A for full details. For measuring robustness to ℓ_p-norm bounded perturbations (in Table 2) we attack our models with one of the strongest available combinations of attacks: AutoAttack & MultiTargeted as in (Gowal et al., 2020).

Competing methods and evaluation. We compare *AdA* to nominal training, and to two challenging data augmentation methods: DeepAugment (Hendrycks et al., 2020a) and AugMix (Hendrycks et al., 2020b). These augmentation methods have been shown to greatly improve robustness against image corruptions. We summarize performance on CIFAR-10-C using the mean corruption error (mCE) introduced in (Hendrycks & Dietterich, 2019). mCE measures and aggregates top-1 classifier error across all corruption types and severities. For a given corruption c the corruption error across the five severities is $E_c = \sum_{s=1}^{5} E_{c,s}$. We report mCE as the mean over the corruption errors E_c, and top-1 errors averaged across all severities for each individual corruption type (no IMAGENET-style normalization of top-1 errors is done). For ℓ_p-norm robustness we summarize performance as robust top-1 accuracy at different perturbation radii.

4 CONCLUSION

Models trained with *AdA* (coupled with AugMix) obtain very good performance against common image corruptions in expectation, as shown in Table 1. Combining *AdA* with increasingly more complex methods results in monotonic improvements to mCE; i.e., coupling *AdA* with AugMix improves mCE from 12.49% to 8.80%; adding DeepAugment further pushes mCE to 8.58%. Finally, adding 10× more DeepAugment samples pushes mCE to 7.83% establishing a new state-of-the-art for CIFAR-10-C. In the adversarial setting (Table 2) models trained with *AdA* perform best across all metrics; these models gain a limited form of robustness to ℓ_p-norm perturbations despite not training *directly* to defend against this type of attack. Interestingly, models trained with *AdA* (to be robust to a synthetic distribution shift) generalize better than all baselines to CIFAR-10.1 (a natural distribution shift), with the best variant achieving 91.05% accuracy.

We have shown that our method, *AdA*, can be used to defend against common image corruptions by training robust models, obtaining a new state-of-the-art mean corruption error on CIFAR-10-C, improving upon *DeepAugment*'s performance on ℓ_p-norm bounded perturbations and generalizing better to CIFAR-10.1. We hope our method will inspire future work on robustness to common image corruptions. We are actively working on training and evaluating *AdA* models on IMAGENET-C.

REFERENCES

Martin Arjovsky, Léon Bottou, Ishaan Gulrajani, and David Lopez-Paz. Invariant risk minimization. *arXiv preprint arXiv:1907.02893*, 2020. URL https://arxiv.org/pdf/1907.02893.

Shumeet Baluja and Ian Fischer. Adversarial transformation networks: Learning to generate adversarial examples. *arXiv preprint arXiv:1703.09387*, 2017. URL https://arxiv.org/pdf/1703.09387.

Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D. Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, Xin Zhang, Jake Zhao, and Karol Zieba. End to end learning for self-driving cars. *NIPS Deep Learning Symposium*, 2016.

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.

Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for YouTube recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, 2016.

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark, 2020.

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.

Jeffrey De Fauw, Joseph R Ledsam, Bernardino Romera-Paredes, Stanislav Nikolov, Nenad Tomasev, Sam Blackwell, Harry Askham, Xavier Glorot, Brendan O'Donoghue, Daniel Visentin, George van den Driessche, Balaji Lakshminarayanan, Clemens Meyer, Faith Mackinder, Simon Bouton, Kareem Ayoub, Reena Chopra, Dominic King, Alan Karthikesalingam, Cían O Hughes, Rosalind Raine, Julian Hughes, Dawn A Sim, Catherine Egan, Adnan Tufail, Hugh Montgomery, Demis Hassabis, Geraint Rees, Trevor Back, Peng T Khaw, Mustafa Suleyman, Julien Cornebise, Pearse A Keane, and Olaf Ronneberger. Clinically applicable deep learning for diagnosis and referral in retinal disease. In *Nature Medicine*, 2018.

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

Robert Geirhos, Carlos RM Temme, Jonas Rauber, Heiko H Schütt, Matthias Bethge, and Felix A Wichmann. Generalisation in humans and deep neural networks. In *Advances in Neural Information Processing Systems*, pp. 7538–7550, 2018.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/pdf?id=Bygh9j09KX.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. URL http://www.deeplearningbook.org.

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

Sven Gowal, Chongli Qin, Po-Sen Huang, Taylan Cemgil, Krishnamurthy Dvijotham, Timothy Mann, and Pushmeet Kohli. Achieving Robustness in the Wild via Adversarial Mixing with Disentangled Representations. *arXiv preprint arXiv:1912.03192*, 2019. URL https://arxiv.org/pdf/1912.03192.

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.

Hongyu Guo, Yongyi Mao, and Richong Zhang. Mixup as locally linear out-of-manifold regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2019. URL https://ojs.aaai.org/index.php/AAAI/article/download/4256/4134.

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

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European conference on computer vision*, pp. 630–645. Springer, 2016b.

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

Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *arXiv preprint arXiv:1907.07174*, 2019.

Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The Many Faces of Robustness: A Critical Analysis of Out-of-Distribution Generalization. *arXiv preprint arXiv:2006.16241*, 2020a. URL https://arxiv.org/pdf/2006.16241.

Dan Hendrycks, Norman Mu, Ekin D. Cubuk, Barret Zoph, Justin Gilmer, and Balaji Lakshminarayanan. Augmix: A simple data processing method to improve robustness and uncertainty. *Int. Conf. Learn. Represent.*, 2020b.

Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, and others. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6):82–97, 2012.

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

Klim Kireev, Maksym Andriushchenko, and Nicolas Flammarion. On the effectiveness of adversarial training against common corruptions, 2021.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Adv. Neural Inform. Process. Syst.*, 2012.

Andrey Kuehlkamp, Benedict Becker, and Kevin Bowyer. Gender-from-iris or gender-from-mascara? In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 1151–1159. IEEE, 2017.

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

Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. Perceptual adversarial robustness: Defense against unseen threat models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/pdf?id=dFwBosAcJkN.

Bee Lim, Sanghyun Son, Heewon Kim, Seungjun Nah, and Kyoung Mu Lee. Enhanced deep residual networks for single image super-resolution, 2017.

Raphael Gontijo Lopes, Dong Yin, Ben Poole, Justin Gilmer, and Ekin D. Cubuk. Improving robustness without sacrificing accuracy with patch gaussian augmentation. *arXiv preprint arXiv:1906.02611*, 2019. URL https://arxiv.org/pdf/1906.02611.

Ilya Loshchilov and Frank Hutter. SGDR: stochastic gradient descent with warm restarts. In *Int. Conf. Learn. Represent.*, 2017.

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

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Jonathan Uesato, and Pascal Frossard. Robustness via curvature regularization, and vice versa. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

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

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.

Chongli Qin, James Martens, Sven Gowal, Dilip Krishnan, Krishnamurthy Dvijotham, Alhussein Fawzi, Soham De, Robert Stanforth, and Pushmeet Kohli. Adversarial Robustness through Local Linearization. *Adv. Neural Inform. Process. Syst.*, 2019.

Haonan Qiu, Chaowei Xiao, Lei Yang, Xinchen Yan, Honglak Lee, and Bo Li. SemanticAdv: Generating Adversarial Examples via Attribute-conditional Image Editing. *arXiv preprint arXiv:1906.07927*, 2019. URL `https://arxiv.org/pdf/1906.07927`.

Aditi Raghunathan, Sang Michael Xie, Fanny Yang, John Duchi, and Percy Liang. Adversarial training can hurt generalization. In *ICML 2019 Workshop on Identifying and Understanding Deep Learning Phenomena*, 2019. URL `https://openreview.net/pdf?id=SyxM3J256E`.

Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet Classifiers Generalize to ImageNet? *arXiv preprint arXiv:1902.10811*, 2019.

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

Yang Song, Rui Shu, Nate Kushman, and Stefano Ermon. Constructing unrestricted adversarial examples with generative models. In *Advances in Neural Information Processing Systems*, pp. 8312–8323, 2018.

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.

Rohan Taori, Achal Dave, Vaishaal Shankar, Nicholas Carlini, Benjamin Recht, and Ludwig Schmidt. Measuring robustness to natural distribution shifts in image classification. *Advances in Neural Information Processing Systems*, 33, 2020.

Lucas Theis, Wenzhe Shi, Andrew Cunningham, and Ferenc Huszár. Lossy image compression with compressive autoencoders, 2017.

Yuji Tokozume, Yoshitaka Ushiku, and Tatsuya Harada. Between-class learning for image classification. *IEEE Conf. Comput. Vis. Pattern Recog.*, 2018.

Antonio Torralba, Alexei A Efros, and others. Unbiased look at dataset bias. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2011.

Vladimir Vapnik. *Statistical learning theory*. Wiley New York, 1998.

Igor Vasiljevic, Ayan Chakrabarti, and Gregory Shakhnarovich. Examining the impact of blur on recognition by convolutional networks. *arXiv preprint arXiv:1611.05760*, 2016.

Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

Eric Wong and J Zico Kolter. Learning perturbation sets for robust machine learning. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/pdf?id=MIDckA56aD.

Eric Wong, Leslie Rice, and J. Zico Kolter. Fast is better than free: Revisiting adversarial training. *Int. Conf. Learn. Represent.*, 2020.

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

Chaowei Xiao, Bo Li, Jun-Yan Zhu, Warren He, Mingyan Liu, and Dawn Song. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*, 2018. URL https://arxiv.org/pdf/1801.02610.

Dong Yin, Raphael Gontijo Lopes, Jonathon Shlens, Ekin D. Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. *arXiv preprint arXiv:1906.08988*, 2019. URL https://arxiv.org/pdf/1906.08988.

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.

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.

APPENDIX

A EXPERIMENTAL SETUP DETAILS

Training and evaluation details. We train pre-activation *ResNet50* (He et al., 2016b) models (as in Wong et al. (2020)); as in previous work (Hendrycks et al., 2020b) our models use 3×3 kernels for the first convolutional layer. We use standard CIFAR-10 data augmentation consisting of padding by 4 pixels, randomly cropping back to 32×32 and randomly flipping left-to-right.

Outer minimization. We minimize the corrupted adversarial risk by optimizing the classifier's parameters using stochastic gradient descent with Nesterov momentum (Polyak, 1964; Nesterov, 1983). We train for 300 epochs with a batch size of 256 and use a global weight decay of 10^{-4}. We use a cosine learning rate schedule (Loshchilov & Hutter, 2017), without restarts, with 5 warm-up epochs, with an initial learning rate of 0.1 which is decayed to 0 at the end of training. We scale all learning rates using the linear scaling rule of Goyal et al. (2017), i.e., effective LR $= \max(\text{LR} \times \text{batch size}/256, \text{LR})$.

Inner maximization. Corrupted adversarial examples are obtained by maximizing the cross-entropy between the classifier's predictions on the corrupted inputs (by passing them through the corruption network) and their labels. We initialize the perturbations to the corruption network parameters randomly within the feasible region. We optimize the perturbations using 10 steps of iterated FGSM (Goodfellow et al., 2015; Kurakin et al., 2016)[1]. We project (i.e., clip) the optimization iterates to stay within the feasible region. We use a step size equal to $1/4$ of the median perturbation radius over all parameter blocks.

Combining data augmentation methods. We view the process of combining data augmentation methods as a data pipeline, where the output from each stage is fed as input to the next stage. We first draw random samples either from the clean training dataset, or from the DeepAugment-processed training set if DeepAugment is used (this is how DeepAugment is used in the original paper (Hendrycks et al., 2020a)). Then we apply standard data augmentation (random pad and crop for CIFAR-10). If *AdA* is used, we apply the method now in the pipeline (followed by the SSIM line-search procedure). If AugMix is used, we apply it as the final step in the data pipeline.

B RELATED WORK

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., 2016a). 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; Tokozume et al., 2018) (which linearly interpolates between two images) all demonstrate extremely compelling results. Guo et al. (2019) improved upon *mixup* by proposing an adaptive mixing policy. Works, such as *AutoAugment* (Cubuk et al., 2019) and related *RandAugment* (Cubuk et al., 2020), learn augmentation policies from data directly. These methods are tuned to improve standard classification accuracy and have been shown to work well on CIFAR-10, CIFAR-100, SVHN and IMAGENET. However, these approaches do not necessarily generalize well to larger data shifts and perform poorly on benign corruptions (e.g., blur or speckle noise) (Taori et al., 2020).

Robustness to synthetic and natural data shift. Several works argue that training against corrupted data only forces the memorization of such corruptions and, as a result, models fail to generalize to new corruptions (Vasiljevic et al., 2016; Geirhos et al., 2018). This has not prevented Geirhos et al. (2019); Yin et al. (2019); Hendrycks et al. (2020b); Lopes et al. (2019); Hendrycks et al. (2020a) from demonstrating that some forms of data augmentation can improve the robustness of models on IMAGENET-C, despite not being directly trained on these common corruptions. Most works on the

[1]We also experimented with Adam, SGD and normalized gradient ascent but we obtained the best results using FGSM.

topic focus on training models that perform well in expectation. Unfortunately, these models remain vulnerable to more drastic adversarial shifts (Taori et al., 2020).

Robustness to adversarial data shift. Adversarial data shift is extensively studied (Goodfellow et al., 2015; Kurakin et al., 2016; Szegedy et al., 2014; Moosavi-Dezfooli et al., 2019; Papernot et al., 2016; Madry et al., 2018). Most works focus the robustness of classifiers to ℓ_p-norm bounded perturbations. In particular, it is expected that a *robust* classifier be invariant to small perturbations in the pixel space (as defined by the ℓ_p-norm). Goodfellow et al. (2015) and Madry et al. (2018) laid down foundational principles to train robust networks, and recent works (Zhang et al., 2019; Qin et al., 2019; Rice et al., 2020; Wu et al., 2020; Gowal et al., 2020) continue to find novel approaches to enhance adversarial robustness. However, approaches focused on ℓ_p-norm bounded perturbations often sacrifice accuracy on non-adversarial images (Raghunathan et al., 2019). Several works (Baluja & Fischer, 2017; Song et al., 2018; Xiao et al., 2018; Qiu et al., 2019; Wong & Kolter, 2021; Laidlaw et al., 2021) go beyond these analytically defined perturbations and demonstrate that it is not only possible to maintain accuracy on non-adversarial images but also reduce the effect of spurious correlations and reduce bias (Gowal et al., 2019). Unfortunately, most aforementioned approaches perform poorly on CIFAR-10-C and IMAGENET-C.

C Relationship to Previous Works

Relationship to Invariant Risk Minimization. IRM proposed by Arjovsky et al. (2020) considers the case where there are multiple datasets $D_e = \{x_i, y_i\}_{i=1}^n$ drawn from different training environments $e \in \mathcal{E}$. The motivation behind IRM is to minimize the worst-case risk

$$\max_{e \in \mathcal{E}} \mathbb{E}_{(x,y) \in D_e} \left[L(f_\theta(x), y) \right]. \tag{4}$$

In this work, the environments are defined by the different corruptions x' resulting from adversarially choosing the parameter offsets β of ϕ. Given a dataset $\{x_i, y_i\}_{i=1}^n$, we can rewrite the *corrupted adversarial risk* shown in Equation 1 as Equation 4 by setting the environment set $\mathcal{E}$ to

$$\mathcal{E} = \{\{c_{\phi+\beta}(x_i), y_i\}_{i=1}^n | \|\beta\|_2 \leq \nu\}. \tag{5}$$

This effectively create an ensemble of datasets for all possible values of β for all examples. The crucial difference between IRM and *AdA* is in the formulation of the risk. In general, we expect *AdA* to be more risk-averse than IRM, as it considers individual examples to be independent from each other.

Relationship to Adversarial Mixing. Gowal et al. (2019) formulate a similar adversarial setup where image perturbations are generated by optimizing a subset of latents corresponding to pre-trained generative models. In this work, we can equivalently consider the parameters of our image-to-image models to be latents and could formulate *AdvMix* into the *AdA* framework. To contrary of *AdvMix*, we do not need to rely on a known partitioning of the latents (i.e., disentangled latents), but do need to restrict the feasible set of parameter offsets β.

Relationship to Perceptual Adversarial Training. Laidlaw et al. (2021) directly optimize input pixels and bound changes as to not exceed the perceptual distance (i.e., LPIPS) between a original clean image and its corrupted variant. Their setup requires a complex machinery to project corrupted images back to the feasible set of images (within a fixed perceptual distance). *AdA*, by construction, uses a well-defined perturbation set and projecting corrupted network's parameters is a trivial operation. This is only possible because perturbations are defined on weights/biases rather than input pixels.

D Comparison to Vanilla and Perceptual Adversarial Training

Perceptual Adversarial Training (PAT) (Laidlaw et al., 2021) proposes an adversarial training method based on bounding a neural perceptual distance (LPIPS). Appendix G (Table 10) of the PAT article shows the performance of various models on common image corruptions. However, performance is summarized using *relative* mCE, whereas we use *absolute* mCE throughout. The authors kindly provided us[2] with the raw corruption errors of their models at each severity and we reproduce their

[2]Personal communication.

results in (the top half of) Table 3. We observe that PAT has overall lower robustness to common image corruptions (best variant obtains 23.54% mCE) than AugMix (10.90% mCE) and than our best *AdA*-trained model (7.83% mCE).

PAT however, performs very well against other adversarial attacks[3], including ℓ_p-norm bounded perturbations. The best PAT model obtains 28.7% robust accuracy against ℓ_∞ attacks ($\epsilon = 8/255$) and 33.3% on ℓ_2 attacks ($\epsilon = 1$) while our best *AdA*-variant obtains less robust accuracy in each case (0.99% against ℓ_2 attacks and 13.88% against ℓ_∞ attacks with $\epsilon = 4/255$). This difference in performance against ℓ_p-norm attacks is not surprising, as PAT addresses robustness to pixel-level attacks (i.e., it manipulates image pixels directly); whereas *AdA* applies adversarial perturbations to the corruption function parameters (and not to the image pixels directly).

In similar spirit to PAT, (Kireev et al., 2021) introduce an efficient relaxation of adversarial training with LPIPS as the distance metric. Their best model, with a smaller architecture, RESNET18, obtains 11.47% mCE on CIFAR-10-C.

The authors of (Kireev et al., 2021) also show that models trained adversarially against ℓ_p-norm bounded perturbations can act as a strong baseline for robustness to common image corruptions. The strongest known[4] adversarially trained model against ℓ_p-norm bounded perturbations on common image corruptions is that of (Gowal et al., 2020) which obtains 12.32% mCE (training against ℓ_2-norm bounded perturbations with $\epsilon = 0.5$ while using extra-data).

To the best of our knowledge, our best performing model is more robust to common image corruptions on CIFAR-10 than all previous methods, obtaining a new state-of-the-art mCE of 7.83%.

Table 3: **Performance of Perceptual Adversarial Training on common image corruptions.** The table lists the performance of *ResNet50* models trained using Perceptual Adversarial Training by the original authors of (Laidlaw et al., 2021) on common image corruptions and two of our *AdA*-trained models. The table shows clean error, mean corruption error on CIFAR-10-C and individual corruption errors for each corruption type (averaged across all severities). "PAT-self" denotes the case where the same model is used for classification as well as for computing the LPIPS distance, while "PAT-AlexNet" denotes the case where the LPIPS distance is computed using a pre-trained CIFAR-10 AlexNet (Krizhevsky et al., 2012) classifier.

SETUP	CLEAN E	MCE	NOISE			BLUR				WEATHER				DIGITAL			
			GAUSS	SHOT	IMPULSE	DEFOCUS	GLASS	MOTION	ZOOM	SNOW	FROST	FOG	BRIGHT	CONTRAST	ELASTIC	PIXEL	JPEG
PAT MODELS (LAIDLAW ET AL., 2021)																	
Nominal Training	5.20%	25.80%	54.0	42.3	38.3	16.5	50.9	21.9	21.1	18.3	24.0	10.5	6.1	16.0	17.6	28.2	20.7
Adversarial Training ℓ_∞	13.20%	20.71%	18.3	17.0	22.3	16.9	19.8	20.4	17.5	17.0	18.2	32.9	13.7	47.9	18.1	15.0	15.3
Adversarial Training ℓ_2	15.00%	21.83%	18.4	17.5	21.4	18.3	20.2	21.0	18.7	19.2	20.5	35.9	16.0	47.1	20.0	16.6	16.5
PAT-self	17.60%	23.54%	22.5	21.1	25.7	20.0	22.5	22.2	20.3	23.7	23.6	33.5	19.8	38.8	21.7	18.7	19.1
PAT-AlexNet	28.40%	34.25%	33.2	31.9	36.3	30.9	34.3	33.0	32.0	33.9	35.0	43.7	30.5	46.5	32.6	29.9	29.8
SELECTION OF *AdA* MODELS (OURS)																	
AdA (EDSR)	3.82%	12.49%	25.8	19.8	29.0	9.3	15.6	10.9	9.6	9.3	8.1	8.8	4.3	11.0	9.3	6.7	9.2
AdA (EDSR) + DeepAugment (10×) + AugMix	5.07%	7.83%	8.8	7.8	11.2	5.9	10.7	7.3	6.5	8.5	6.7	8.7	5.2	6.2	8.5	7.7	7.8

E QUALITATIVE RESULTS

We visualize the performance of *AdA* trained models (best *AdA*-combination from Table 1) during training in Figure 2. Due to adversarial training, we expect the performance on each of the *-C corruptions to improve as training progresses, and this is indeed what we observe. The *AdA*-trained classifier performs consistently best on *Brightness*, especially at the beginning of training.

F RECONSTRUCTING COMMON CORRUPTIONS THROUGH *EDSR* AND *CAE*

Figure 3 shows how well the corruption networks used with *AdA* (*EDSR* and *CAE*) can be used to approximate the common and extra corruptions used in the CIFAR-10-C benchmark.

We optimize the perturbation to the corruption network (*EDSR* or *CAE*) parameters that best transform a clean image into its corrupted variant (with a small penalty on the perturbation norm). We solve the

[3] See Table 2 of (Laidlaw et al., 2021) for full details.

[4] See the ROBUSTBENCH (Croce et al., 2020) leaderboard: `https://robustbench.github.io`.

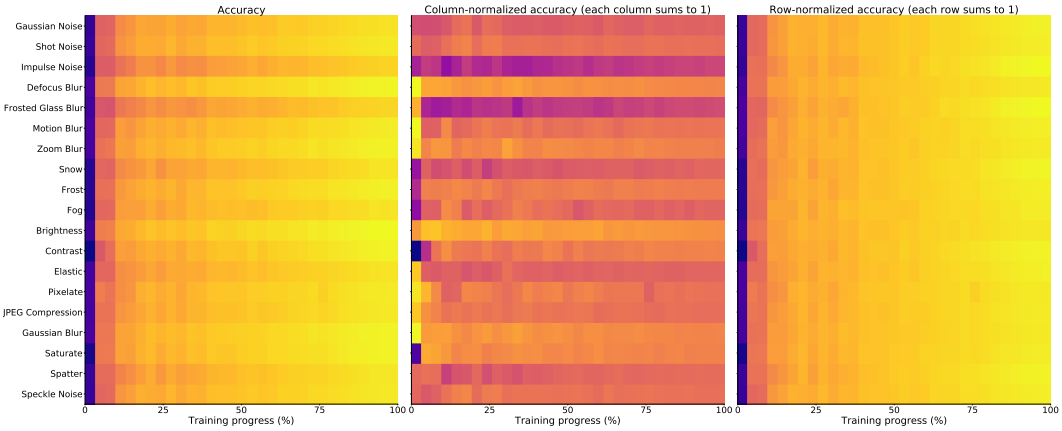


Figure 2: **Performance on image corruptions through training.** These plots visualize the performance of the best *AdA* combination on each of the common and extra *-C corruptions as training progresses. Each individual rectangle plots top-1 accuracy. Brighter is better. The accuracies are visualized raw (plots to the left), normalized over the columns (middle plots) or over the rows (plots to the right). Normalizing over the columns visualizes which corruption's performance is best at that point in training. Normalizing over the rows visualizes at which stage the classifier performs best on a given corruption.

following optimization problem for a random subset of 640 images from each CIFAR-10-C dataset:

$$\max_{\delta} \text{SSIM}(c_{\phi+\delta}(x), \hat{x}) - 10^{-5}\|\delta\|_2^2, \tag{6}$$

where δ is the perturbation to the corruption network's parameters, x is a clean example and $\hat{x}$ is its corrupted variant. We use 50 steps of Adam (Kingma & Ba, 2014) with a learning rate of 0.001.

G APPROXIMATE SSIM LINE-SEARCH PROCEDURE

The adversarial examples produced by *AdA* can sometimes become too severely corrupted (i.e., left-tail of densities in Figure 1 (b) of the main manuscript). We guard against these unlikely events by using an efficient, approximate line-search procedure. We set a maximum threshold, denoted by t, on the SSIM distance between the clean example and the *AdA* adversarial example.

Denote by x_γ the linear combination of the clean example, x, and the corrupted example output by *AdA*, $\hat{x}$:

$$x_\gamma = (1-\gamma)\, x + \gamma\, \hat{x}.$$

When the deviation in SSIM between the clean and the corrupted example is greater than the threshold ($\text{SSIM}(x, \hat{x}) > t$), we find a scalar $\gamma^* \in [0, 1]$ such that the deviation between the corrected example, x^*_γ, and the clean example, x, is t: $\text{SSIM}(x, x^*_\gamma) = t$. We take 9 equally-spaced γ values in $[0, 1]$ and evaluate the SSIM distance between the clean example and x_γ for each considered γ. We fit a quadratic polynomial in γ to all the pairs of γ and the threshold-shifted SSIM deviation from the clean example $\text{SSIM}(x, x_\gamma) - t$. We then find the roots of this polynominal, clip them to $[0, 1]$, and take the γ closest to 1 as the desired γ^*. This corresponds to returning the most corrupted variant of x along the ray between x and $\hat{x}$ which obeys the SSIM threshold. The procedure is very efficient on accelerators (GPUs, TPUs) as it requires no iteration. It is approximate however because the quadratic polynomial sometimes underfits.

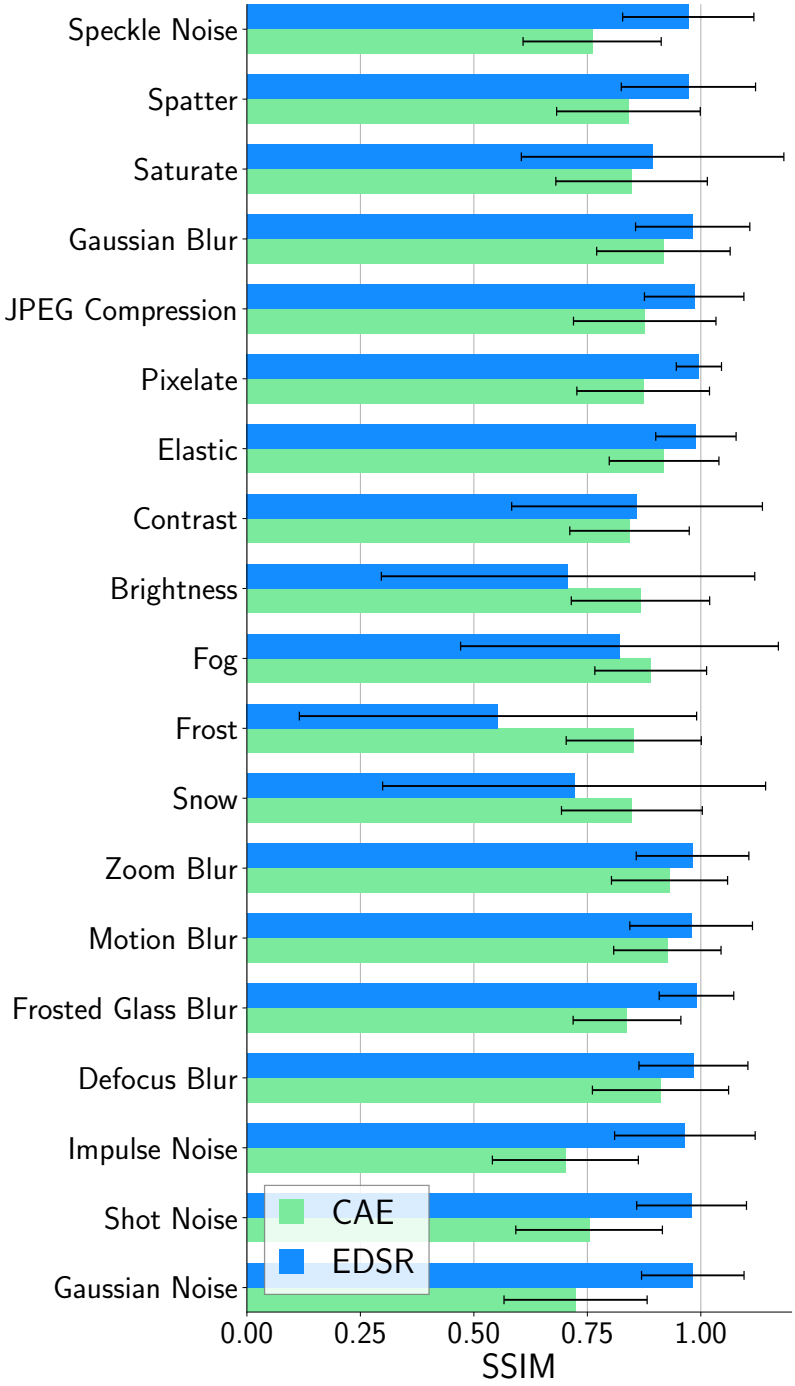


Figure 3: These bar plots show the extent to which the *EDSR* and *CAE* models can be used to approximate the effects of the corruptions present in CIFAR-10-C. Each bar shows the mean and std. dev. SSIM error (Wang et al., 2004) (higher is better, 1 is max.) between pairs of corrupted images and their reconstructions through *EDSR* or *CAE* respectively, optimized starting from the corresponding clean images. Both models can approximate most corruptions well, except for *Brightness* and *Snow*. Some corruption types (e.g. *Fog*, *Frost*, *Snow*) are better approximated by *CAE* (0.84 ± 0.16 overall SSIM) while most are better approximated by *EDSR* (0.91 ± 0.26 overall SSIM).