A simple way to make neural networks robust against diverse image corruptions

Evgenia Rusak[1,2*], Lukas Schott[1,2*], Roland S. Zimmermann[1,2*], Julian Bitterwolf[2], Oliver Bringmann[1†], Matthias Bethge[1,2†], and Wieland Brendel[1,2†]

[1] University of Tübingen
[2] International Max Planck Research School for Intelligent Systems
[*] joint first / [†] joint senior authors
{first.last}@uni-tuebingen.de

Abstract. The human visual system is remarkably robust against a wide range of naturally occurring variations and corruptions like rain or snow. In contrast, the performance of modern image recognition models strongly degrades when evaluated on previously unseen corruptions. Here, we demonstrate that a simple but properly tuned training with additive Gaussian and Speckle noise generalizes surprisingly well to unseen corruptions, easily reaching the state of the art on the corruption benchmark ImageNet-C (with ResNet50) and on MNIST-C. We build on top of these strong baseline results and show that an adversarial training of the recognition model against locally correlated worst-case noise distributions leads to an additional increase in performance. This regularization can be combined with previously proposed defense methods for further improvement.

Keywords: Image corruptions, robustness, generalization, adversarial training

1 Introduction

While Deep Neural Networks (DNNs) have surpassed the functional performance of humans in a range of complex cognitive tasks [12], [44], [38], [2], [30], they still lag behind humans in numerous other aspects. One fundamental shortcoming of machines is their lack of robustness against input perturbations. Even minimal perturbations that are hardly noticeable for humans can derail the predictions of high-performance neural networks.

For the purpose of this paper, we distinguish between two types of input perturbations. One type are minimal image-dependent perturbations specifically designed to fool a neural network with the smallest possible change to the input. These so-called *adversarial perturbations* have been the subject of hundreds of papers in the past five years, see e.g. [39], [21], [35], [11]. Another, much less studied type are *common corruptions*. These perturbations occur naturally

in many applications and include simple Gaussian or Salt and Pepper noise; natural variations like rain, snow or fog; and compression artifacts such as those caused by JPEG encoding. All of these corruptions do not change the semantic content of the input, and thus, machine learning models should not change their decision-making behavior in their presence. Nonetheless, high-performance neural networks like ResNet50 [12] are easily confused by small deformations [1]. The juxtaposition of adversarial examples and common corruptions was explored in [8] where the authors discuss the relationship between both and encourage researchers working in the field of adversarial robustness to cross-evaluate the robustness of their models towards common corruptions.

We argue that in many practical applications, robustness to common corruptions is often more relevant than robustness to artificially designed adversarial perturbations. Autonomous cars should not change their behavior in the face of unusual weather conditions such as hail or sand storms or small pixel defects in their sensors. Not-Safe-For-Work filters should not fail on images with unusual compression artifacts. Likewise, speech recognition algorithms should perform well regardless of the background music or sounds.

Besides its practical relevance, robustness to common corruptions is also an excellent target in its own right for researchers in the field of adversarial robustness and domain adaptation. Common corruptions can be seen as distributional shifts or as a weak form of adversarial examples that live in a smaller, constrained subspace.

Despite their importance, common corruptions have received relatively little attention so far. Only recently, a modification of the ImageNet dataset [34] to benchmark model robustness against common corruptions and perturbations has been published [13] and is referred to as ImageNet-C. Now, this scheme has also been applied to other common datasets resulting in Pascal-C, Coco-C and Cityscapes-C [25] and MNIST-C [29].

Our contributions are as follows:

- We demonstrate that data augmentation with Gaussian or Speckle noise serves as a simple yet very strong baseline that is sufficient to surpass almost all previously proposed defenses against common corruptions on ImageNet-C for ResNet50. We further show that the magnitude of the additive noise is a crucial hyper-parameter to reach optimal robustness.
- Motivated by our strong results with baseline noise augmentations, we introduce a neural network-based *adversarial noise generator* that can learn arbitrary uncorrelated noise distributions that maximally fool a given recognition network when added to their inputs. We denote the resulting noise patterns as *adversarial noise*.
- We design and validate a constrained Adversarial Noise Training (ANT) scheme through which the recognition network learns to become robust against adversarial i.i.d. noise. We demonstrate that our ANT reaches state-of-the-art robustness on the corruption benchmark ImageNet-C for the commonly used ResNet50 architecture and on MNIST-C, even surpassing the already strong baseline noise augmentations. This result is not due to overfitting on the

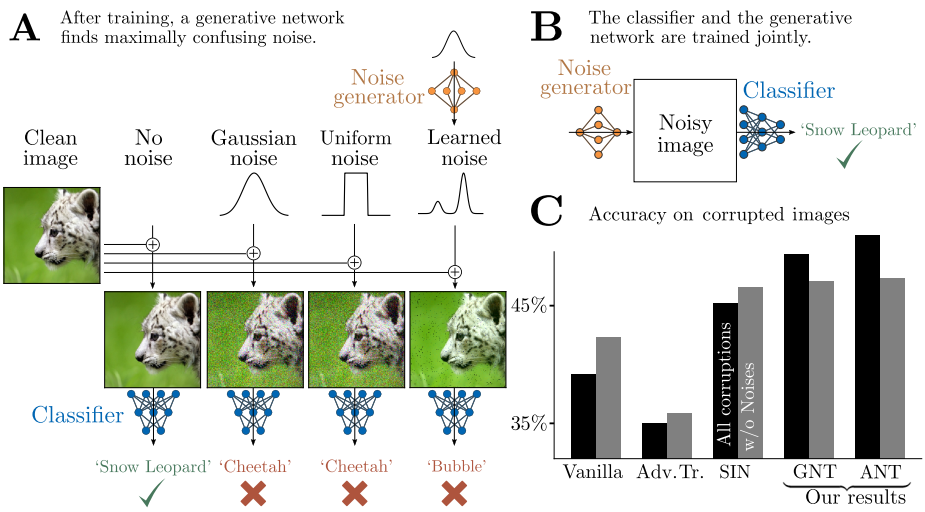

Fig. 1. Outline of our approach. A: First, we train a generative network against a vanilla trained classifier to find the adversarial noise. B: To achieve robustness against adversarial noise, we train the classifier and the noise generator jointly. C: We measure the robustness against common corruptions for a vanilla, adversarially trained (Adv. Tr.), trained on Stylized ImageNet (SIN), trained via Gaussian data augmentation (GNT) and trained with the means of Adversarial Noise Training (ANT). With our methods, we achieve the highest accuracy on common corruptions, both on all and non-noise categories.

noise categories of the respective benchmarks since we find equivalent results on the non-noise corruptions as well.

- We extend the adversarial noise generator towards locally correlated noise thereby enabling it to learn more diverse noise distributions. Performing ANT with the modified noise generator, we observe an increase in robustness for the 'snow' corruption which is visually similar to our learned noise.
- We demonstrate a further increase in robustness when combining ANT with previous defense methods.
- We substantiate the claim that increased robustness against regular or universal adversarial perturbations does not imply increased robustness against common corruptions. This is not necessarily true vice-versa: Our noise trained recognition network has high accuracy on ImageNet-C and also slightly improved accuracy on adversarial attacks on clean ImageNet compared to a vanilla trained ResNet50.

We released our model weights along with the full training code on GitHub. [1]

[1] github.com/bethgelab/game-of-noise

2 Related work

Robustness against common corruptions Several recent publications study the vulnerability of DNNs to common corruptions.

Two recent studies compare humans and DNNs on recognizing corrupted images, showing that DNN performance drops much faster than human performance for increased perturbation sizes [5], [10]. Hendrycks et al. introduce corrupted versions of standard datasets denoted as ImageNet-C, Tiny ImageNet-C and CIFAR10-C as standardized benchmarks for machine learning models [13]. Similarly, common corruptions have been applied to and evaluated on COCO-C, Pascal-C, Cityscapes-C [25] and MNIST-C [29].

There have been attempts to increase robustness against common corruptions. Zhang et al. integrate an anti-aliasing module from the signal processing domain in the ResNet50 architecture to restore the shift-equivariance which can get lost in deep CNNs and report an increased accuracy on clean data and better generalization to corrupted image samples [45]. Concurrent work to ours demonstrates that having more training data [43], [22] or using stronger backbones [43], [25], [18] can significantly improve model performance on common corruptions.

A popular method to decrease overfitting and help the network generalize better to unseen data is to augment the training dataset by applying a set of (randomized) manipulations to the images [26]. Furthermore, augmentation methods have also been applied to make the models more robust against image corruptions [9]. Augmentation with Gaussian [8], [19] or uniform noise [10] has been tried to increase model robustness. Conceptually, Ford et al. is the closest study to our work, since they also apply Gaussian noise to images to increase corruption robustness [8]. They use a different architecture (InceptionV3 versus our ResNet50). Also, they train a new model from scratch solely on images perturbed by Gaussian noise whereas we fine-tune a pretrained model on a mixture of clean and noisy images. They observe a low relative improvement in accuracy on corrupted images whereas we were able to outperform all previous baselines on the commonly used ResNet50 architecture.[2] Lopes et al. restrict the Gaussian noise to small image patches, which improves accuracy but does not yield state-of-the-art performance on the ResNet50 architecture [19]. Geirhos et al. train ImageNet classifiers against a fixed set of corruptions but find no generalized robustness against unseen corruptions [10]. However, they considered vastly higher noise levels than us. Considering the efficacy of Gaussian or uniform data augmentation to increase model robustness, the main difference to our work is that other works have used either munch larger [10] or smaller [8], [19] values for the standard deviation σ. A too large σ leads to an overfitting to the used noise distribution whereas a too small σ leads to noise levels that are not different enough from the clean images. We show that taking σ from the intermediate regime works best for generalization both to other noise types and non-noise corruptions.

[2] To compare with Ford et al., we evaluate our approach for an InceptionV3 architecture, see our results in Appendix H.

Link between adversarial robustness and common corruptions There is currently no agreement on whether adversarial training increases robustness against common corruptions in the literature. Hendrycks et al. report a robustness increase on common corruptions due to adversarial logit pairing on Tiny ImageNet-C [13]. Ford et al. suggest a link between adversarial robustness and robustness against common corruptions, claim that increasing one robustness type should simultaneously increase the other, but report mixed results on MNIST and CIFAR10-C [8]. Additionally, they also observe large drops in accuracy for adversarially trained networks and networks trained with Gaussian data augmentation compared to a vanilla classifier on certain corruptions. On the other hand, Engstrom et al. report that increasing robustness against adversarial ℓ_∞ attacks does not increase robustness against translations and rotations, but they do not present results on noise [7]. Kang et al. study robustness transfer between models trained against ℓ_1, ℓ_2, ℓ_∞ adversaries / elastic deformations and JPEG artifacts [17]. They observe that adversarial training increases robustness against elastic and JPEG corruptions on a 100-class subset of ImageNet. This result contradicts our findings on full ImageNet as we see a slight decline in accuracy on those two classes for the adversarially trained model from [42] and severe drops in accuracy on other corruptions. Jordan et al. show that adversarial robustness does not transfer easily between attack classes [16]. Tramèr et al. [40] also argue in favor of a trade-off between different robustness types. For a simple and natural classification task, they prove that adversarial robustness towards l_∞ perturbations does neither transfer to l_1 nor to input rotations and translations, and vice versa and support their formal analysis with experiments on MNIST and CIFAR10.

3 Methods

3.1 Training with Gaussian noise

As discussed in section 2, several researchers have tried using Gaussian noise as a method to increase robustness towards common corruptions with mixed results. In this work, we revisit the approach of Gaussian data augmentation and increase its efficacy. We treat the standard deviation σ of the distribution as a hyper-parameter of the training and measure its influence on robustness.

To formally introduce the objective, let $\mathcal{D}$ be the data distribution over input pairs $(\boldsymbol{x}, y)$ with $\boldsymbol{x} \in \mathbb{R}^N$ and $y \in \{1, ..., k\}$. We train a differentiable classifier $f_\theta(\boldsymbol{x})$ by minimizing the risk on a dataset with additive Gaussian noise

$$\mathop{\mathbb{E}}_{\boldsymbol{x},y\sim\mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{\delta}\sim\mathcal{N}(\mathbf{0},\sigma^2\mathbb{1})} \left[\mathcal{L}_{\mathrm{CE}}\left(f_\theta(\mathrm{clip}(\boldsymbol{x}+\boldsymbol{\delta})), y\right)\right], \tag{1}$$

where σ is the standard deviation of the Gaussian noise and $\boldsymbol{x} + \boldsymbol{\delta}$ is clipped to the input range $[0, 1]^N$. The standard deviation is either kept fixed or is chosen uniformly from a fixed set of standard deviations. In both cases, the possible standard deviations are chosen from a small set of nine values inspired by the

noise variance in the ImageNet-C dataset (cf. section 3.3). To maintain high accuracy on clean data, we only perturb 50% of the training data with Gaussian noise within each batch.

3.2 Adversarial noise

Learning Adversarial Noise Our goal is to find a noise distribution $p_\phi(\boldsymbol{\delta})$, $\boldsymbol{\delta} \in \mathbb{R}^N$ such that noise samples added to $\boldsymbol{x}$ maximally confuse the classifier f_θ. More concisely, we optimize

$$\max_{\phi} \mathop{\mathbb{E}}_{\boldsymbol{x},y\sim\mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{\delta}\sim p_\phi(\boldsymbol{\delta})} \left[\mathcal{L}_{\text{CE}}\left(f_\theta(\text{clip}(\boldsymbol{x}+\boldsymbol{\delta})), y\right)\right], \tag{2}$$

where clip is an operator that clips all values to the valid interval (i.e. $\text{clip}(\boldsymbol{x}+\boldsymbol{\delta}) \in [0,1]^N$) and restricts their norm $||\boldsymbol{\delta}||_2 = \epsilon$. [3]

We follow the literature of implicit generative models [28], [4] as we do not have to explicitly model the probability density function $p_\phi(\boldsymbol{\delta})$ since optimizing Eq. (2) only involves samples drawn from $p_\phi(\boldsymbol{\delta})$. We model the samples from $p_\phi(\boldsymbol{\delta})$ as the output of a neural network $g_\phi : \mathbb{R}^N \to \mathbb{R}^N$ which gets its input from a normal distribution $\boldsymbol{\delta} = g_\phi(\boldsymbol{z})$ where $\boldsymbol{z} \sim \mathcal{N}(\boldsymbol{0}, \mathbb{1})$. We enforce the independence property of $p_\phi(\boldsymbol{\delta}) = \prod_n p_\phi(\delta_n)$ by constraining the network architecture of the noise generator g_ϕ to only consist of convolutions with 1x1 kernels. Lastly, the projection onto a sphere $||\boldsymbol{\delta}||_2 = \epsilon$ is achieved by scaling the generator output with a scalar while clipping $\boldsymbol{x} + \boldsymbol{\delta}$ to the valid range $[0,1]^N$. This fixed size projection (hyper-parameter) is motivated by the fact that Gaussian noise training with a single, fixed σ achieved the highest accuracy. [4]

The noise generator g_ϕ has four 1x1 convolutional layers with ReLU activations and one residual connection from input to output. The weights of the layers are initialized to small numbers; for this initialization, the input is passed through the residual connection to the output. Since we use Gaussian noise as input, the noise generator outputs Gaussian noise at initialization. During training, the weights change and the generator learns to produce more diverse distributions.

Adversarial Noise Training To increase robustness, we now train the classifier f_θ to minimize the risk under adversarial noise distributions jointly with the noise generator

$$\min_{\theta} \max_{\phi} \mathop{\mathbb{E}}_{\boldsymbol{x},y\sim\mathcal{D}} \mathop{\mathbb{E}}_{\boldsymbol{\delta}\sim p_\phi(\boldsymbol{\delta})} \left[\mathcal{L}_{\text{CE}}\left(f_\theta(\text{clip}(\boldsymbol{x}+\boldsymbol{\delta})), y\right)\right], \tag{3}$$

where again $\boldsymbol{x} + \boldsymbol{\delta} \in [0,1]^N$ and $||\boldsymbol{\delta}||_2 = \epsilon$. For a joint adversarial training, we alternate between an outer loop of classifier update steps and an inner loop of

[3] We apply the method derived in [32] and rescale the perturbation by a factor γ to obtain the desired ℓ_2 norm; despite the clipping, the squared ℓ_2 norm is a piece-wise linear function of γ^2 that can be inverted to find the correct scaling factor γ.

[4] We also experimented with an adaptive sphere radius ϵ which grows with the classifier's accuracy. However, we did not see any improvements and followed Occam's razor.

generator update steps. Note that in regular adversarial training, e.g. [21], $\boldsymbol{\delta}$ is optimized directly whereas we optimize a constrained distribution over $\boldsymbol{\delta}$.

To maintain high classification accuracy on clean samples, we sample every mini-batch so that they contain 50% clean data and perturb the rest. The current state of the noise generator is used to perturb 30% of this data and the remaining 20% are augmented with samples chosen randomly from previous distributions. For this, the noise generator states are saved at regular intervals. The latter method is inspired by experience replay from reinforcement learning [27] and is used to keep the classifier from forgetting previous adversarial noise patterns. To prevent the noise generator from being stuck in a local minimum, we halt the Adversarial Noise Training (ANT) at regular intervals and train a new noise generator from scratch. This noise generator is trained against the current state of the classifier to find a current optimum. The new noise generator replaces the former noise generator in the ANT. This technique has been crucial to train a robust classifier.

Learning locally correlated adversarial noise We modify the architecture of the noise generator defined in Eq. 2 to allow for local spatial correlations and thereby enable the generator to learn more diverse distributions. Since we seek to increase model robustness towards image corruptions such as rain or snow that produce locally correlated patterns, it is natural to include local patterns in the manifold of learnable distributions. We replace the 1x1 kernels in one network layer with 3x3 kernels limiting the maximum correlation length of the output noise sample to 3x3 pixels. We indicate the correlation length of noise generator used for the constrained adversarial noise training as $\text{ANT}^{1\text{x}1}$ or $\text{ANT}^{3\text{x}3}$.

Combining Adversarial Noise Training with stylization As demonstrated by [9], using random stylization as data augmentation increases the accuracy on ImageNet-C due to a higher shape bias of the model. We combine our ANT and the stylization approach to achieve robustness gains from both in the following way: we split the samples in each batch into clean data (25%), stylized data (30%) and clean data perturbed by the noise generator (45%).

3.3 Evaluation on corrupted images

Evaluation of noise robustness We evaluate the robustness of a model by sampling a Gaussian noise vector $\boldsymbol{\delta}$ (covariance $\mathbb{1}$). We then do a line search along the direction $\boldsymbol{\delta}$ starting from the original image $\boldsymbol{x}$ until it is misclassified. We denote the resulting minimal perturbation as $\boldsymbol{\delta}_{\text{min}}$. The robustness of a model is then denoted by the median[5] over the test set

$$\epsilon^* = \underset{\boldsymbol{x},y\sim\mathcal{D}}{\text{median}} \, ||\boldsymbol{\delta}_{\text{min}}||_2, \tag{4}$$

[5] Samples for which no ℓ_2-distance allows us to manipulate the classifier's decision contribute a value of ∞ to the median.

with $f_\theta(\boldsymbol{x} + \boldsymbol{\delta}_{\min}) \neq y$ and $\boldsymbol{x} + \boldsymbol{\delta}_{\min} \in [0,1]^N$. Note that a higher ϵ^* denotes a more robust classifier. To test the robustness against adversarial noise, we train a new noise generator at the end of the Adversarial Noise Training until convergence and evaluate it according to Eq. (4).

ImageNet-C The ImageNet-C benchmark[6] [13] is a conglomerate of 15 diverse corruption types that were applied to the validation set of ImageNet. The corruptions are organized into four main categories: noise, blur, weather, and digital. The MNIST-C benchmark is created similarly to ImageNet-C with a slightly different set of corruptions [29]. We report the Top-1 and Top-5 accuracies as well as the 'mean Corruption Error' (mCE) on both benchmarks. We evaluate all proposed methods for ImageNet-C on the ResNet50 architecture for better comparability to previous methods, e.g. [9], [19], [45]. The clean ImageNet accuracy of the used architecture highly influences the results and could be seen as an upper bound for the accuracy on ImageNet-C. Note that our approach is independent of the used architecture and could be applied to any differentiable network.

4 Results

For our experiments on ImageNet, we use a classifier that was pretrained on ImageNet. For the experiments on MNIST, we use the architecture from [21] for comparability. All technical details, hyper-parameters and the architectures of the noise generators can be found in Appendix A-B. We use various open source software packages for our experiments, most notably Docker [24], scipy and numpy [41], PyTorch [31] and torchvision [23].

(In-)Effectiveness of regular adversarial training to increase robustness towards common corruptions In our first experiment, we evaluate whether robustness against regular adversarial examples generalizes to robustness against common corruptions. We display the Top-1 accuracy of vanilla and adversarially trained models in Table 1; detailed results on individual corruptions can be found in Appendix C. For all tested models, we find that regular ℓ_∞ adversarial training can strongly decrease the robustness towards common corruptions, especially for the corruption types Fog and Contrast. Universal adversarial training [37], on the other hand, leads to severe drops on some corruptions but the overall accuracy on ImageNet-C is slightly increased relative to the vanilla baseline model (AlexNet). Nonetheless, the absolute ImageNet-C accuracy of 22.2% is still very low. These results disagree with two previous studies which

[6] For the evaluation, we use the JPEG compressed images from github.com/hendrycks/robustness as is advised by the authors to ensure reproducibility. We note that Ford et al. report a decrease in performance when the compressed JPEG files are used as opposed to applying the corruptions directly in memory without compression artifacts [8].

Table 1: Top-1 accuracy on ImageNet-C and ImageNet-C without the noise category (higher is better). Regular adversarial training decreases robustness towards common corruptions; universal adversarial training seems to slightly increase it.

Model	IN-C	IN-C w/o noises
Vanilla RN50	39.2%	42.3%
Adv. Training [36]	29.1%	32.0%
Vanilla RN152	45.0%	47.9%
Adv. Training [42]	35.0%	35.9%
Vanilla AlexNet	21.1%	23.9%
Universal Adv. Training [37]	22.2%	23.1%

reported that (1) adversarial logit pairing[7] (ALP) increases robustness against common corruptions on Tiny ImageNet-C [13], and that (2) adversarial training can increase robustness on CIFAR10-C [8].

We evaluate adversarially trained models on MNIST-C and present the results and their discussion in Appendix E. The results on MNIST-C show the same tendency as on ImageNet-C: adversarially trained models have lower accuracy on MNIST-C and thus indicate that adversarial robustness does not transfer to robustness against common corruptions. This corroborates the results of Ford et al. [8] on MNIST who also found that an adversarially robust model had decreased robustness towards a set of common corruptions.

Effectiveness of Gaussian data augmentation to increase robustness towards common corruptions We fine-tune ResNet50 classifier pretrained on ImageNet with Gaussian data augmentation from the distribution $\mathcal{N}(\mathbf{0}, \sigma^2\mathbb{1})$ and vary σ. We try two different settings: in one, we choose a single noise level σ while in the second, we sample σ uniformly from a set of multiple possible values. The Top-1 accuracy of the fine-tuned models on ImageNet-C in comparison to a vanilla trained model is shown in Fig. 2. Each black point shows the performance of one model fine-tuned with one specific σ; the vanilla trained model is marked by the point at $\sigma = 0$. The horizontal lines indicate that the model is fine-tuned with Gaussian noise where σ is sampled from a set for each image. For example, for the dark green line, as indicated by the stars, we sample σ from the set $\{0.08, 0.12, 0.18, 0.26, 0.38\}$ which corresponds to the Gaussian corruption of ImageNet-C. Since Gaussian noise is part of the test set, we show both the results on the full ImageNet-C evaluation set and the results on ImageNet-C without noises (namely blur, weather and digital). To show how the different σ-levels manifest themselves in an image, we include example images in Appendix G.

There are three important results evident from Fig. 2:

[7] Note that ALP was later found to not increase adversarial robustness [6].

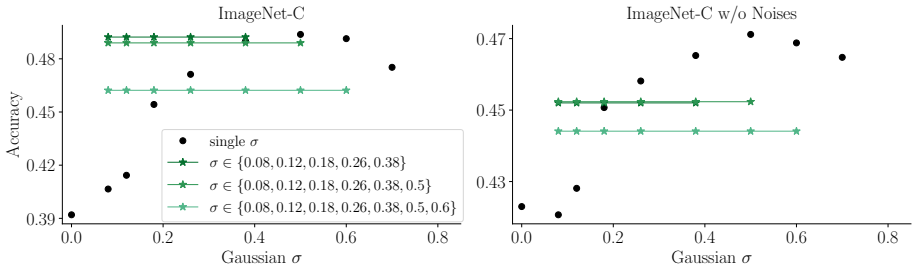

Fig. 2. Top-1 accuracy on ImageNet-C (left) and ImageNet-C without the noise corruptions (right) of a ResNet50 architecture fine-tuned with Gaussian data augmentation of varying σ. Each dot or green line represents one model. We train on Gaussian noise sampled from a distribution with a single σ (black dots) and on distributions where σ is sampled from different sets (green lines with stars). We also compare to a vanilla trained model at $\sigma = 0$.

1. Gaussian noise generalizes well to the non-noise corruptions of the ImageNet-C dataset and is a powerful baseline. This is surprising as it was shown in several recent works that training on Gaussian or uniform noise does not generalize to other corruption types [10], [19] or that the effect is weak [8].
2. The standard deviation σ is a crucial hyper-parameter and has an optimal value of about $\sigma = 0.5$ for ResNet50.
3. If σ is chosen well, using a single σ is enough and sampling from a set of σ values is detrimental for robustness against non-noise corruptions.

In the following Results sections, we will compare Gaussian data augmentation to our Adversarial Noise Training approach and baselines from the literature. For this, we will use the models with the overall best-performance: The model $\text{GN}_{0.5}$ that was trained with Gaussian data augmentation with a single $\sigma = 0.5$ and the model GN_{mult} where σ was sampled from the set $\{0.08, 0.12, 0.18, 0.26, 0.38\}$.

Evaluation of the severity of adversarial noise as an attack In this section, we focus on the question: Can we learn the most severe uncorrelated additive noise distribution for a classifier? Following the success of simple uncorrelated Gaussian noise data augmentation (section 4) and the ineffectiveness of regular adversarial training (section 4) which allows for highly correlated patterns, we restrict our learned noise distribution to be sampled independently for each pixel.

To measure the effectiveness of our adversarial noise, we report the median perturbation size ϵ^* that is necessary for a misclassification for each image in the test set as defined in section 3.3. We find $\epsilon^*_{\text{GN}} = 39.0$ for Gaussian noise, $\epsilon^*_{\text{UN}} = 39.1$ for uniform noise and $\epsilon^*_{\text{AN}} = 15.7$ for adversarial noise (see Fig. 1 for samples of each noise type). Thus, we see that our AN is much more effective at fooling the classifier compared to Gaussian and uniform noise.

Table 2: Accuracy on clean data and robustness of differently trained models as measured by the median perturbation size ϵ^*. A higher ϵ^* indicates a more robust model. We compute standard deviations for ϵ^*_{AN} for differently initialized generator networks. To provide an intuition for the perturbation sizes indicated by ϵ^*, we show example images for Gaussian noise below and a larger Figure for different noise types in Appendix I.

model	clean acc.	ϵ^*_{GN}	ϵ^*_{UN}	$\epsilon^*_{\text{AN1x1}}$
Vanilla RN50	**76.1%**	39.0	39.1	15.7 ± 0.6
GNT$\sigma_{0.5}$	75.9%	74.8	74.9	31.8 ± 3.9
GNT$_{\text{mult}}$	**76.1%**	130.1	130.7	24.0 ± 2.2
ANT$^{\text{1x1}}$	76.0%	**136.7**	**137.0**	**95.4 ± 5.7**

ϵ^*=15.0 ϵ^*=30.0 ϵ^*=60.0 ϵ^*=120.0

Evaluation of Adversarial Noise Training as a defense In the previous section, we established a method for learning the most adversarial noise distribution for a classifier. Now, we utilize it for a joint Adversarial Noise Training (ANT$^{\text{1x1}}$) where we simultaneously train the noise generator and classifier (see section 3.2). This leads to substantially increased robustness against Gaussian, uniform and adversarial noise, see Table 2. The robustness of models that were trained via Gaussian data augmentation also increases, but on average much less compared to the model trained with ANT$^{\text{1x1}}$. To evaluate the robustness against adversarial noise, we train four noise generators with different random seeds and measure $\epsilon^*_{\text{AN1x1}}$. We report the mean value and the standard deviation over the four runs. To visualize this effect, we visualize the temporal evolution of the probability density function $p_\phi(\delta_n)$ of uncorrelated noise during ANT$^{\text{1x1}}$ in Fig. 3A. This shows that the generator converges to different distributions and therefore, the classifier has been trained against a rich variety of distributions.

Comparison of different methods to increase robustness towards common corruptions We now revisit common corruptions on ImageNet-C and compare the robustness of differently trained models. Since Gaussian noise is part of ImageNet-C, we train another baseline model with data augmentation using the Speckle noise corruption from the ImageNet-C holdout set. We later denote the cases where the corruptions present during training are part of the test set by putting corresponding accuracy values in brackets. Additionally, we compare our results with several baseline models from the literature:

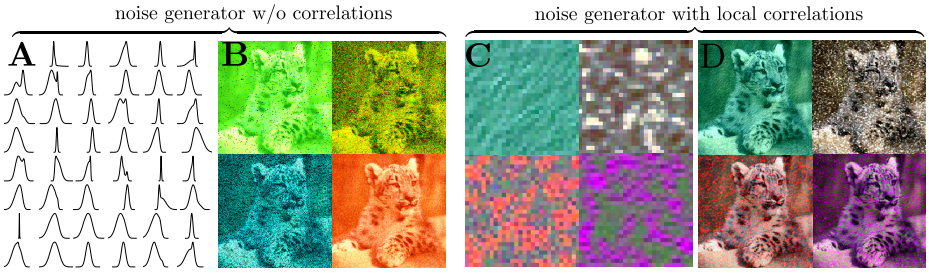

Fig. 3. A: Examples of learned probability densities over the grayscale version of the noise $\boldsymbol{\delta}_n$ during ANT$^{1\text{x}1}$ where each density corresponds to one local minimum; B: Example images with sampled uncorrelated adversarial noise; C: Example patches of locally correlated noise with a size of 28x28 pixels learned during ANT$^{3\text{x}3}$; D: Example images with sampled correlated adversarial noise.

1. Shift Inv: The model is modified to enhance shift-equivariance using anti-aliasing [45].[8]
2. Patch GN: The model was trained on Gaussian patches [19].[9]
3. SIN+IN: The model was trained on a stylized version of ImageNet [9].[10]
4. AugMix: [14] trained their model using diverse augmentations.[11] They use image augmentations from AutoAugment [3] and exclude contrast, color, brightness, sharpness, and Cutout operations to make sure that the test set of ImageNet-C is disjoint from the training set. We would like to highlight the difficulty in clearly distinguishing between the augmentations used during training and testing as there might be a certain overlap. This can be seen by the visual similarity between the Posterize operation and the JPEG corruption (see Appendix J).

The Top-1 accuracies on the full ImageNet-C dataset and ImageNet-C without the noise corruptions are displayed in Table 3; detailed results on individual corruptions in terms of accuracy and mCE are shown in Tables 3 and 4, Appendix D. We also calculate the accuracy on corruptions without the noise category since we observe that the generated noise can sometimes be close to the i.i.d. corruptions of ImageNet-C raising concerns about overfitting. Additionally, the expressiveness of the generated i.i.d. noise is quite limited compared to natural corruptions like 'snow'. We hence extend the ANT$^{1\text{x}1}$ procedure to include spatially correlated noise over 3x3 pixels. Samples are shown in Fig. 3C and Fig. 3D.

The results on full ImageNet-C are striking (see Table 3): a very simple baseline, namely a model trained with Speckle noise data augmentation, beats

[8] Weights were taken from github.com/adobe/antialiased-cnns.
[9] Since no model weights are released, we include the values reported in their paper.
[10] Weights were taken from github.com/rgeirhos/texture-vs-shape.
[11] Weights were taken from github.com/google-research/augmix.

Table 3: Average accuracy on clean data, average Top-1 and Top-5 accuracies on ImageNet-C and ImageNet-C without the noise category (higher is better); all values in percent. We compare the results obtained by the means of Gaussian (GNT) and Speckle noise data augmentation and with Adversarial Noise Training (ANT) to several baselines. Gray numbers in brackets indicate scenarios where a corruption from the test set was used during training.

	IN	IN-C		IN-C w/o noises	
model	clean acc.	Top-1	Top-5	Top-1	Top-5
Vanilla RN50	76.1	39.2	59.3	42.3	63.2
Shift Inv [45]	77.0	41.4	61.8	44.2	65.1
Patch GN [19]	76.0	(43.6)	(n.a.)	43.7	n.a.
SIN+IN [9]	74.6	45.2	66.6	46.6	68.2
AugMix [14]	**77.5**	48.3	69.2	50.4	71.8
Speckle	75.8	46.4	67.6	44.5	65.5
GNT_{mult}	76.1	(49.2)	(70.2)	45.2	66.2
$\text{GNT}\sigma_{0.5}$	75.9	(49.4)	(70.6)	47.1	68.3
$\text{ANT}^{1\text{x}1}$	76.0	(51.1)	(72.2)	47.7	68.8
$\text{ANT}^{1\text{x}1}$+SIN	74.9	(52.2)	(73.6)	49.2	70.6
$\text{ANT}^{1\text{x}1}$ w/o EP	75.7	(48.9)	(70.2)	46.5	67.7
$\text{ANT}^{3\text{x}3}$	76.1	50.4	71.5	47.0	68.1
$\text{ANT}^{3\text{x}3}$+SIN	74.1	**52.6**	**74.4**	**50.6**	**72.5**

almost all previous baselines reaching an accuracy of 46.4% which is larger than the accuracy of SIN+IN (45.2%) and close to AugMix (48.3%). The $\text{GN}\sigma_{0.5}$ surpasses SIN+IN not only on the noise category but also on almost all other corruptions, see a more detailed breakdown in Table 3, Appendix D.

The $\text{ANT}^{3\text{x}3}$+SIN model produces the best results on ImageNet-C both with and without noises. Thus, it is slightly superior to Gaussian data augmentation and pure $\text{ANT}^{3\text{x}3}$. Comparing $\text{ANT}^{1\text{x}1}$ and $\text{ANT}^{3\text{x}3}$, we observe that $\text{ANT}^{3\text{x}3}$ performs better than $\text{ANT}^{1\text{x}1}$ on the 'snow' corruption. We attribute this to the successful modeling capabilities of locally correlated patterns resembling snow of the 3x3 noise generator. We perform an ablation study to investigate the necessity of experience replay and note that we lose roughly 2% without it ($\text{ANT}^{1\text{x}1}$ w/o EP vs $\text{ANT}^{1\text{x}1}$). We also test how the classifier's performance changes if it is trained against adversarial noise sampled randomly from $p_\phi(\delta_n)$. The accuracy on ImageNet-C decreases slightly compared to regular $\text{ANT}^{1\text{x}1}$: 51.1%/ 71.9% (Top-1/ Top-5) on full ImageNet-C and 47.3%/ 68.3% (Top-1/ Top-5) on ImageNet-C without the noise category. We include additional results for $\text{ANT}^{1\text{x}1}$ with a DenseNet121 architecture [15] and for varying parameter counts of the noise generator in Appendix K.

For MNIST, we train a model with Gaussian data augmentation and via $\text{ANT}^{1\text{x}1}$. We achieve similar results with both approaches and report a new state-of-the-art accuracy on MNIST-C: 92.4%, see Appendix E for details.

Table 4: Adversarial robustness on ℓ_2 ($\epsilon = 0.12$) and ℓ_∞ ($\epsilon = 0.001$) compared to a Vanilla ResNet50 on ImageNet.

model	clean acc. [%]	ℓ_2 acc. [%]	ℓ_∞ acc. [%]
Vanilla RN50	**76.1**	41.1	18.1
GNT$\sigma_{0.5}$	75.9	49.0	28.1
ANT$^{1\text{x}1}$	76.0	50.1	28.6
Adv. Training [36]	60.5	**58.1**	**58.5**

Robustness towards adversarial perturbations As regular adversarial training can decrease the accuracy on common corruptions, it is also interesting to check what happens vice-versa: How does a model which is robust on common corruptions behave under adversarial attacks?

Both our ANT$^{1\text{x}1}$ and GNT models have slightly increased ℓ_2 and ℓ_∞ robustness scores compared to a vanilla trained model, see Table 4. We tested this using the white-box attacks PGD [20] and DDN [33]. Expectedly, an adversarially trained model has higher adversarial robustness compared to ANT$^{1\text{x}1}$ or GNT. In this experiment, we only verify that we do not unintentionally reduce adversarial robustness compared to a vanilla ResNet50. For details, see Appendix E for MNIST and Appendix F for ImageNet.

5 Conclusions

So far, attempts to use simple noise augmentations for general robustness against common corruptions have produced mixed results, ranging from no generalization from one noise to other noise types [10] to only marginal robustness increases [8], [19]. In this work, we demonstrate that carefully tuned additive noise patterns in conjunction with training on clean samples can surpass almost all current state-of-the-art defense methods against common corruptions. By drawing inspiration from adversarial training and experience replay, we additionally show that training against simple uncorrelated or locally correlated worst-case noise patterns outperforms our already strong baseline defense, with additional gains to be made in combination with previous defense methods like stylization [9].

There are still a few corruption types (e.g. Motion or Zoom blurs) on which our method is not state of the art, suggesting that additional gains are possible. Future extensions of this work may combine noise generators with varying correlation lengths, add additional interactions between noise and image (e.g. multiplicative interactions or local deformations) or take into account local image information in the noise generation process to further boost robustness across many types of image corruptions.

References

1. Azulay, A., Weiss, Y.: Why do deep convolutional networks generalize so poorly to small image transformations? (2018)
2. Campbell, M., Hoane, Jr., A.J., Hsu, F.h.: Deep blue. Artif. Intell. **134**(1-2), 57–83 (Jan 2002). https://doi.org/10.1016/S0004-3702(01)00129-1, http://dx.doi.org/10.1016/S0004-3702(01)00129-1
3. Cubuk, E.D., Zoph, B., Mane, D., Vasudevan, V., Le, Q.V.: Autoaugment: Learning augmentation policies from data. arXiv preprint arXiv:1805.09501 (2018)
4. Diggle, P.J., Gratton, R.J.: Monte carlo methods of inference for implicit statistical models. Journal of the Royal Statistical Society: Series B (Methodological) **46**(2), 193–212 (1984)
5. Dodge, S.F., Karam, L.J.: A study and comparison of human and deep learning recognition performance under visual distortions. CoRR **abs/1705.02498** (2017), http://arxiv.org/abs/1705.02498
6. Engstrom, L., Ilyas, A., Athalye, A.: Evaluating and understanding the robustness of adversarial logit pairing. CoRR **abs/1807.10272** (2018), https://arxiv.org/abs/1807.10272
7. Engstrom, L., Tsipras, D., Schmidt, L., Madry, A.: A rotation and a translation suffice: Fooling cnns with simple transformations. ICML (2019)
8. Ford, N., Gilmer, J., Carlini, N., Cubuk, D.: Adversarial examples are a natural consequence of test error in noise. ICML (2019)
9. Geirhos, R., Rubisch, P., Michaelis, C., Bethge, M., Wichmann, F.A., Brendel, W.: Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=Bygh9j09KX
10. Geirhos, R., Temme, C.R.M., Rauber, J., Schütt, H.H., Bethge, M., Wichmann, F.A.: Generalisation in humans and deep neural networks. In: Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R. (eds.) Advances in Neural Information Processing Systems 31, pp. 7538–7550. Curran Associates, Inc. (2018), http://papers.nips.cc/paper/7982-generalisation-in-humans-and-deep-neural-networks.pdf
11. Gilmer, J., Metz, L., Faghri, F., Schoenholz, S.S., Raghu, M., Wattenberg, M., Goodfellow, I.J.: Adversarial spheres. CoRR **abs/1801.02774** (2018), http://arxiv.org/abs/1801.02774
12. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
13. Hendrycks, D., Dietterich, T.: Benchmarking neural network robustness to common corruptions and perturbations. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=HJz6tiCqYm
14. Hendrycks, D., Mu, N., Cubuk, E.D., Zoph, B., Gilmer, J., Lakshminarayanan, B.: Augmix: A simple data processing method to improve robustness and uncertainty. In: International Conference on Learning Representations (2020), https://openreview.net/forum?id=S1gmrxHFvB
15. Huang, G., Liu, Z., Weinberger, K.Q.: Densely connected convolutional networks. CVPR (2017)
16. Jordan, M., Manoj, N., Goel, S., Dimakis, A.G.: Quantifying perceptual distortion of adversarial examples. arXiv preprint arXiv:1902.08265 (2019)

17. Kang, D., Sun, Y., Brown, T., Hendrycks, D., Steinhardt, J.: Transfer of adversarial robustness between perturbation types. CoRR **abs/1905.01034** (2019), http://arxiv.org/abs/1905.01034
18. Lee, J., Won, T., Hong, K.: Compounding the performance improvements of assembled techniques in a convolutional neural network. arXiv preprint arXiv:2001.06268 (2020)
19. Lopes, R.G., Yin, D., Poole, B., Gilmer, J., Cubuk, E.D.: Improving robustness without sacrificing accuracy with patch gaussian augmentation. CoRR **abs/1906.02611** (2019), http://arxiv.org/abs/1906.02611
20. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083 (2017)
21. Madry, A., Makelov, A., Schmidt, L., Tsipras, D., Vladu, A.: Towards deep learning models resistant to adversarial attacks. In: International Conference on Learning Representations (2018), https://openreview.net/forum?id=rJzIBfZAb
22. Mahajan, D., Girshick, R., Ramanathan, V., He, K., Paluri, M., Li, Y., Bharambe, A., van der Maaten, L.: Exploring the limits of weakly supervised pretraining. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 181–196 (2018)
23. Marcel, S., Rodriguez, Y.: Torchvision the machine-vision package of torch. In: ACM International Conference on Multimedia (2010)
24. Merkel, D.: Docker: Lightweight linux containers for consistent development and deployment. Linux J. **2014**(239) (Mar 2014)
25. Michaelis, C., Mitzkus, B., Geirhos, R., Rusak, E., Bringmann, O., Ecker, A.S., Bethge, M., Brendel, W.: Benchmarking robustness in object detection: Autonomous driving when winter is coming. arXiv preprint arXiv:1907.07484 (2019)
26. Mikołajczyk, A., Grochowski, M.: Data augmentation for improving deep learning in image classification problem. 2018 International Interdisciplinary PhD Workshop (IIPhDW) pp. 117–122 (2018)
27. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A.A., Veness, J., Bellemare, M.G., Graves, A., Riedmiller, M., Fidjeland, A.K., Ostrovski, G., et al.: Human-level control through deep reinforcement learning. Nature **518**(7540), 529 (2015)
28. Mohamed, S., Lakshminarayanan, B.: Learning in implicit generative models. arXiv preprint arXiv:1610.03483 (2016)
29. Mu, N., Gilmer, J.: MNIST-C: A robustness benchmark for computer vision. arXiv preprint arXiv:1906.02337 (2019)
30. OpenAI: Openai five. https://blog.openai.com/openai-five/ (2018)
31. Paszke, A., Gross, S., Chintala, S., Chanan, G., Yang, E., DeVito, Z., Lin, Z., Desmaison, A., Antiga, L., Lerer, A.: Automatic differentiation in PyTorch. In: NIPS Autodiff Workshop (2017)
32. Rauber, J., Bethge, M.: Fast differentiable clipping-aware normalization and rescaling. arXiv preprint arXiv:2007.07677 (2020), https://github.com/jonasrauber/clipping-aware-rescaling
33. Rony, J., Hafemann, L.G., Oliveira, L.S., Ayed, I.B., Sabourin, R., Granger, E.: Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4322–4330 (2019)
34. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M.S., Berg, A.C., Li, F.: Imagenet large scale visual recognition challenge. CoRR **abs/1409.0575** (2014), http://arxiv.org/abs/1409.0575

35. Schott, L., Rauber, J., Bethge, M., Brendel, W.: Towards the first adversarially robust neural network model on MNIST. In: International Conference on Learning Representations (2019), https://openreview.net/forum?id=S1EHOsC9tX
36. Shafahi, A., Najibi, M., Ghiasi, A., Xu, Z., Dickerson, J., Studer, C., Davis, L.S., Taylor, G., Goldstein, T.: Adversarial training for free! arXiv preprint arXiv:1904.12843 (2019)
37. Shafahi, A., Najibi, M., Xu, Z., Dickerson, J.P., Davis, L.S., Goldstein, T.: Universal adversarial training. CoRR **abs/1811.11304** (2018), http://arxiv.org/abs/1811.11304
38. Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L.R., Lai, M., Bolton, A., Chen, Y., Lillicrap, T.P., Hui, F.F.C., Sifre, L., van den Driessche, G., Graepel, T., Hassabis, D.: Mastering the game of go without human knowledge. Nature **550**, 354–359 (2017)
39. Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., Fergus, R.: Intriguing properties of neural networks. arXiv preprint arXiv:1312.6199 (2013)
40. Tramèr, F., Boneh, D.: Adversarial training and robustness for multiple perturbations. NeurIPS (2019), http://arxiv.org/abs/1904.13000
41. Virtanen, P., Gommers, R., Oliphant, T.E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S.J., Brett, M., Wilson, J., Jarrod Millman, K., Mayorov, N., Nelson, A.R.J., Jones, E., Kern, R., Larson, E., Carey, C., Polat, İ., Feng, Y., Moore, E.W., Vand erPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E.A., Harris, C.R., Archibald, A.M., Ribeiro, A.H., Pedregosa, F., van Mulbregt, P., Contributors, S...: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. Nature Methods **17**, 261–272 (2020). https://doi.org/https://doi.org/10.1038/s41592-019-0686-2
42. Xie, C., Wu, Y., van der Maaten, L., Yuille, A.L., He, K.: Feature denoising for improving adversarial robustness. CVPR (2019)
43. Xie, Q., Hovy, E., Luong, M.T., Le, Q.V.: Self-training with noisy student improves imagenet classification. arXiv preprint arXiv:1911.04252 (2019)
44. Xiong, W., Droppo, J., Huang, X., Seide, F., Seltzer, M., Stolcke, A., Yu, D., Zweig, G.: Achieving human parity in conversational speech recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing (2016)
45. Zhang, R.: Making convolutional networks shift-invariant again. ICML (2019)