Improving Robustness Without Sacrificing Accuracy
with Patch Gaussian Augmentation

Raphael Gontijo Lopes1 2  Dong Yin2  Ben Poole2  Justin Gilmer2  Ekin D. Cubuk2

Abstract
Deploying machine learning systems in the real world requires both high accuracy on clean data and robustness to naturally occurring corruptions. While modern advances in architectures have led to improved accuracy, building robust models remains challenging. Previous work claims that there is an inherent trade-off between the two. This potential trade-off is apparent with standard data augmentation techniques such as Cutout, which improves clean accuracy but not robustness, and Gaussian, which improves robustness at the cost of accuracy. Here, we present a simple augmentation scheme, Patch Gaussian, that adds noise to randomly selected patches in an input image. Models trained with Patch Gaussian achieve state of the art on the CIFAR and ImageNet Common Corruptions benchmarks while also improving accuracy on clean data.

1. Introduction
Modern deep neural networks can achieve impressive performance at classifying images in curated datasets (Karpathy, 2011; Krizhevsky et al., 2012; Huang et al., 2018). Yet, their performance is not robust to corruptions that typically occur in real-world settings. For example, neural networks are sensitive to small translations and changes in scale (Azaualy & Weiss, 2018), blurring and additive noise (Dodge & Karam, 2017), small objects placed in images (Rosenfeld et al., 2018), and even different images from a similar distribution of the training set (Reht et al., 2019; 2018). For models to be useful in the real world, they need to be both accurate on a high-quality held-out set of images, which we refer to as “clean accuracy,” and robust on corrupted images, which we refer to as “robustness.” Most of the literature in machine learning has focused on architectural changes (Simonyan & Zisserman, 2015; Szegedy et al., 2015; He et al., 2016; Zoph & Le, 2017; Szegedy et al., 2017; Han et al., 2017; Zoph et al., 2017; Hu et al., 2017; Liu et al., 2018) to improve clean accuracy but has recently become interested in robustness as well.

Research in neural network robustness has tried to quantify the problem by establishing benchmarks that directly measure it (Hendrycks & Dietterich, 2018; Gu et al., 2019) and comparing the performance of humans and neural networks (Geirhos et al., 2018b). Others have tried to understand robustness by highlighting systemic failure modes of current learning methods. For instance, networks exhibit excessive invariance to visual features (Jacobsen et al., 2018), texture bias (Geirhos et al., 2018a), sensitivity to worst-case (adversarial) perturbations (Goodfellow et al., 2014), and a propensity to rely solely on non-robust, but highly predictive features for classification (Doersch et al., 2015; Ilyas et al., 2019). Of particular relevance to our work, Ford et al. (2019) show that in order to achieve adversarial robustness, one needs to achieve robustness to noise-corrupted data.

Another line of work has attempted to increase model robustness performance, either by directly projecting out superficial statistics (Wang et al., 2019), via architectural improvements (Cubuk et al., 2017), pre-training schemes (Hendrycks et al., 2019), or through the use of data augmentations. Data augmentation increases the size and diversity of the training set, and provides a simple method for learning invariances that are challenging to encode architecturally (Cubuk et al., 2018). Recent work in this area include learning better transformations (DeVries & Taylor, 2017; Zhang et al., 2017; Zhong et al., 2017), inferring combinations of transformations (Cubuk et al., 2018), unsupervised methods (Xie et al., 2019), theory of data augmentation (Dao et al., 2018), applications for one-shot learning (Asano et al., 2019), among others.

Despite these advances, individual data augmentation methods that improve robustness do so at the expense of reduced clean accuracy. Some have even claimed that there exists a fundamental trade-off between the two (Tsipras et al., 2018). Because of this, many recent works focus on improving either one or the other (Madry et al., 2017; Geirhos et al.,...
In this work we propose a data augmentation that overcomes this trade-off, achieving both improved robustness and clean accuracy. Our contributions are as follows:

- We characterize a trade-off between robustness and accuracy among two standard data augmentations: Cutout and Gaussian (Sec. 2.1).
- We devise a simple data augmentation method (which we term Patch Gaussian) that enables interpolation between the two augmentations above. (Sec. 3.1)
- We find that Patch Gaussian allows us to overcome the observed trade-off (Section 4.1), and achieves a new state of the art in the Common Corruptions benchmark (Hendrycks & Dietterich, 2018) on CIFAR-C and ImageNet-C. (Sec. 4.2)

2. Preliminaries

We start by considering two data augmentations: Cutout (DeVries & Taylor, 2017) and Gaussian (Grandvalet & Canu, 1997). The former sets a random patch of the input images to a constant (the mean pixel in the dataset) and is successful in reaching higher clean accuracy. The latter works by adding independent Gaussian noise to each pixel of the input image, which can increase robustness to Gaussian noise directly.

To apply Gaussian, we uniformly sample a standard deviation $\sigma$ from 0 up to some maximum value $\sigma_{\text{max}}$, and add iid noise sampled from $\mathcal{N}(0, \sigma^2)$ to each pixel. To apply Cutout, we use a fixed patch size $W$, and randomly set a square region with size $W \times W$ to constant mean of each RGB channel in the dataset. As in (DeVries & Taylor, 2017), the patch location is randomly sampled and can lie outside of the $32 \times 32$ CIFAR-10 (or $224 \times 224$ ImageNet) image but its center is constrained to lie within it. $W$ and $\sigma_{\text{max}}$ are selected based on the method described in Sec. 3.2.

2.1. Cutout and Gaussian exhibit a trade-off between accuracy and robustness

We compare the effectiveness of Gaussian and Cutout data augmentation for accuracy and robustness by measuring the performance of models trained with each on clean data, as well as data corrupted by various standard deviations of Gaussian noise.

Figure 1 highlights an apparent trade-off in using these methods. In accordance to previous work (DeVries & Taylor, 2017), Cutout improves accuracy on clean test data. Despite this, we find it does not lead to increased robustness. Conversely, training with higher $\sigma$ of Gaussian can lead to increased robustness to Gaussian noise, but it also leads to decreased accuracy on clean data. Therefore, any robustness gains are offset by poor overall performance.

At first glance, these results seem to reinforce the findings of previous work (Tsipras et al., 2018), indicating that robustness comes at the cost of generalization. In the following sections, we will explore whether there exists augmentation strategies that do not exhibit this limitation.

3. Method

Each of the two methods seen so far achieves one half of our stated goal: either improving robustness or improving clean test accuracy, but never both. In an attempt to find a data augmentation based on the application of Gaussian noise that also led to improved performance on clean data, we devise a new technique: Patch Gaussian.

3.1. Patch Gaussian

Patch Gaussian works by adding an $N \times N$ patch of Gaussian noise to the image (Figure 2). As with Cutout, the center of the patch is sampled to be within the $32 \times 32$ CIFAR-10 (or $224 \times 224$ ImageNet) image. By varying $\sigma$.
the size of this patch and the maximum standard deviation of noise sampled $\sigma_{\text{max}}$, we can interpolate between Gaussian (which applies a Gaussian to the whole image) and an approximation of cutout (which erases all information from the input at a patch). See Fig. 6 for more examples.

All image transformations, including Patch Gaussian, are performed on images with unnormalized pixel values in $[0, 1]$ range. For all images, standard random flipping and cropping is applied immediately after any augmentations mentioned on CIFAR (before, on Imagenet). After noise-based augmentations, images are clipped to $[0, 1]$ range.

3.2. Hyper-parameter selection

In order to fairly select hyper-parameters for each augmentation, we select the model that is most robust, while still achieving $\geq Z$ accuracy on clean data. Values of $Z$ vary per dataset and model, and can be found in the Appendix. If no model has clean accuracy $\geq Z$, we report the model with highest clean accuracy, unless otherwise specified. Here, robustness is defined as average accuracy of the model, when tested on data corrupted by various $\sigma$ ($0.1, 0.2, 0.3, 0.5, 0.8, 1.0$) of Gaussian noise, relative to the clean accuracy. This metric is correlated with mCE (Ford et al., 2019), so it ensures model robustness is generally useful beyond Gaussian corruptions. By picking models based on their Gaussian noise robustness, we ensure that our selection process does not overfit to the Common Corruptions benchmark (Hendrycks & Dietterich, 2018). We find that patch sizes around 25 on CIFAR ($\leq 250$ for ImageNet, i.e.: uniformly sampled with maximum value 250) with $\sigma \leq 1.0$ generally perform the best. Selected hyper-parameters for all augmentations can be found in the Appendix.

3.3. Models, Datasets, & Implementation Details

We run our experiments on CIFAR-10 (Krizhevsky & Hinton, 2009) and ImageNet (Deng et al., 2009) datasets. On CIFAR-10, we use the Wide-ResNet-28-10 model (Zagoruyko & Komodakis, 2016), as well as the Shake-shake-112 model (Gastaldi, 2017), trained for 200 epochs and 600 epochs respectively. The Wide-ResNet model uses a initial learning rate of 0.1 with a cosine decay schedule. Weight decay is set to be $5e-4$ and batch size is 128. On ImageNet, we use the ResNet-50 and Resnet-200 models (He et al., 2016), trained for 90 epochs. We use a weight decay rate of $1e-4$, global batch size of 512 and learning rate of 0.2. The learning rate is decayed by 10 at epochs 30, 60, and 80. We train all models, including the baseline, with standard data augmentation of horizontal flips and crop.

4. Results

We show that models trained with Patch Gaussian can overcome the trade-off observed in Fig. 1 and learn models that are robust while maintaining their generalization accuracy (Section 4.1). In doing so, we establish a new state of the art in CIFAR-C and ImageNet-C Common Corruptions benchmark (Hendrycks & Dietterich, 2018) (Section 4.2).
4.1. **Patch Gaussian overcomes this trade-off and improves both accuracy and robustness**

We train models on various hyper-parameters of Patch Gaussian and find that the model selected by the method in Sec. 3.2 yields improved robustness to Gaussian noise, like Gaussian, while also improving clean accuracy, like Cutout. In Figure 3, we visualize these results in an ablation study, varying patch sizes and $\sigma$ of the model selected.

4.2. **Training with Patch Gaussian leads to improved Common Corruption robustness**

Thus far, we’ve been analyzing robustness to Gaussian noise. In this section, rather than focusing on worst-case perturbations bounded to be close to the original image in pixel space (e.g.: in $L_2$ or $L_\infty$ metric), we look at a more general set of corruptions (Gilmer et al., 2018) that machine learning models are likely to encounter: the Common Corruptions benchmark (Hendrycks & Dietterich, 2018). This benchmark, also referred to as CIFAR-C and ImageNet-C, is composed of images transformed with 15 corruptions, at 5 severities each. The corruptions are designed to model those commonly found in real-world settings, such as brightness, weather conditions, and different kinds of noise.

Table 1 shows that Patch Gaussian achieves state of the art on both of these benchmarks. In them, we report mean Corruption Error (mCE) of each model. However, ImageNet-C was released in compressed JPEG format (ECMA International, 2009), which alters the corruptions applied to the raw pixels. We report results on the benchmark as-released (“Original mCE”) as well as on 12 corruptions without the extra compression (“mCE (-noise)”).

To compute mCE, one must normalize each Corruption Error with respect to a baseline. For each model and dataset, we normalize with respect to Baseline. The one exception is Original mCE ImageNet, where we use the AlexNet baseline to be directly comparable with previous work (Hendrycks & Dietterich, 2018; Geirhos et al., 2018a).

Because Patch Gaussian is a noise-based augmentation, we wanted to verify whether its gains on this benchmark were solely due to improved performance on noise-based corruptions (Gaussian Noise, Shot Noise, and Impulse Noise). To do this, we also measure the models’ average performance on all other corruptions, reported as “Original mCE (-noise)”, and “mCE (-noise)”. We observe that Patch Gaussian outperforms all other models, even on corruptions like fog, which have performance hurt by Gaussian (Ford et al., 2019). This indicates that Patch Gaussian augmentation is generally useful.

5. **Conclusion**

In this work, we introduced a single data augmentation operation, Patch Gaussian, which improves robustness to common corruptions without incurring a drop in clean accuracy. For models that are large relative to the dataset size (like ResNet-200 on ImageNet and all models on CIFAR-10, Patch Gaussian improves clean accuracy and robustness concurrently. It achieves this by interpolating between two standard data augmentation operations Cutout and Gaussian. Finally, we showed that models trained with Patch Gaussian achieve state of the art on the ImageNet and CIFAR Common Corruptions benchmark.

---

**Table 1.** Patch Gaussian achieves state of the art in the CIFAR-10-C (left) and Imagenet-C (right) benchmark (Hendrycks & Dietterich, 2018) while improving clean accuracy. *Cutout 16 is presented for direct comparison with (DeVries & Taylor, 2017; Gastaldi, 2017). “SIN+IN ftIN” is the shape-biased model from (Geirhos et al., 2018a). “Original mCE” refers to the jpeg-compressed benchmark, as used in (Geirhos et al., 2018a; Hendrycks & Dietterich, 2018). “mCE” is a version of it without the extra jpeg compression. Note that Patch Gaussian improves robustness even in corruptions that aren’t noise-based. Augmentation hyper-parameters were selected based on the method in Sec. 3.2 and can be found in Appendix. For Resnet-200, we also present Gaussian at a higher $\sigma$ to highlight the accuracy-robustness trade-off.


### References


Improving Robustness Without Sacrificing Accuracy with Patch Gaussian Augmentation


Appendix
Table 2. Augmentation hyper-parameters selected with the method in Section 3.2 for each model/dataset. *Indicates manually-chosen stronger hyper-parameters, used to highlight the effect of the augmentation on the models. “≤” indicates that the value is uniformly sampled up to the given maximum value.

<table>
<thead>
<tr>
<th>Model/Dataset</th>
<th>Z</th>
<th>Augmentation</th>
<th>Patch Size</th>
<th>σ</th>
<th>Other</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIFAR-10</td>
<td>96.5%</td>
<td>Cutout</td>
<td>= 12</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gaussian</td>
<td>-</td>
<td>≤ 0.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Patch Gaussian</td>
<td>= 25</td>
<td>≤ 1.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cutout*</td>
<td>= 22</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gaussian*</td>
<td>-</td>
<td>≤ 1.0</td>
<td></td>
</tr>
<tr>
<td>ShakeNet-112</td>
<td>97.0%</td>
<td>Cutout</td>
<td>= 7</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gaussian</td>
<td>-</td>
<td>≤ 0.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Patch Gaussian</td>
<td>= 26</td>
<td>≤ 1.0</td>
<td></td>
</tr>
<tr>
<td>ImageNet</td>
<td>76.0%</td>
<td>Baseline</td>
<td>-</td>
<td>-</td>
<td>includes weight decay = 0.0001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cutout</td>
<td>= 60</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gaussian</td>
<td>-</td>
<td>≤ 0.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Patch Gaussian</td>
<td>≤ 250</td>
<td>≤ 1.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cutout*</td>
<td>= 200</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gaussian*</td>
<td>-</td>
<td>≤ 1.0</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Larger Weight Decay</td>
<td>-</td>
<td>-</td>
<td>0.001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dropblock</td>
<td>-</td>
<td>-</td>
<td>groups = 3,4; keep prob = 0.9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Label Smoothing</td>
<td>-</td>
<td>-</td>
<td>0.1</td>
</tr>
<tr>
<td>Resnet-50</td>
<td>78.5%</td>
<td>Baseline</td>
<td>-</td>
<td>-</td>
<td>includes weight decay = 0.0001</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cutout</td>
<td>= 30</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Gaussian</td>
<td>-</td>
<td>≤ 0.1</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Patch Gaussian</td>
<td>≤ 350</td>
<td>≤ 1.0</td>
<td></td>
</tr>
</tbody>
</table>
Improving Robustness Without Sacrificing Accuracy with Patch Gaussian Augmentation

Figure 4. Accuracy/robustness trade-off observed for Cutout and Gaussian on Resnet-50 models. See figure 1 for details.

Figure 5. Overcoming the accuracy/robustness trade-off with Patch Gaussian for models trained on CIFAR-10 (top row) and Resnet-50 (bottom row). See figure 3 for details.
Figure 6. Visualizing images modified with Patch Gaussian, with centered patch, at various patch sizes and $\sigma$. 