Barrage of Random Transforms for Adversarially Robust Defense

Edward Raff[1,2,4] Jared Sylvester[1,2,4] Steven Forsyth[3] Mark McLean[1]

[1]Laboratory for Physical Sciences [2]Booz Allen Hamilton [3]NVIDIA [4]U.M.B.C.

Abstract

Defenses against adversarial examples, when using the ImageNet dataset, are historically easy to defeat. The common understanding is that a combination of simple image transformations and other various defenses are insufficient to provide the necessary protection when the obfuscated gradient is taken into account. In this paper, we explore the idea of stochastically combining a large number of individually weak defenses into a single barrage of randomized transformations to build a strong defense against adversarial attacks. We show that, even after accounting for obfuscated gradients, the Barrage of Random Transforms (BaRT) is a resilient defense against even the most difficult attacks, such as PGD. BaRT achieves up to a 24× improvement in accuracy compared to previous work, and has even extended effectiveness out to a previously untested maximum adversarial perturbation of $\epsilon = 32$.

1. Introduction

Adversarial machine learning has been a research area for over a decade [1], but it has recently received increased focus and attention from the larger community. This is largely due to the success of modern deep learning techniques within the realm of computer vision tasks and the surprising ease with which such systems are fooled into giving incorrect decisions [2]. In particular, there are concerns about the safety of self-driving cars, as they could be fooled into misreading stop signs as speed limits, and other possible nefarious actions [3].

Consider an adversary A whom, given some victim model $f(\cdot)$, wants to alter $\widetilde{x} = A(x)$ such that $f(x) \neq f(A(x))$. Many works have attempted to find a transform $t(\cdot)$ that can be applied to an image x to yield a new image $\hat{x} = t(x)$ such that $f(x) = f(t(A(x)))$. If it were possible to find such a defensive transform $t(\cdot)$ it would allow us a simple and convenient way to circumvent the adversarial problem. This is particularly alluring for computer vision, since there exists a rich literature of computer vision transformations to pull from. Athalye, Carlini, and Wagner [4] has shown that the many attempts to find such a defensive transformation $t(\cdot)$ that defeats adversarial attacks have all failed, due to a problem they term obfuscated gradients. More broadly, every defense we are aware of that has undergone thorough evaluation has failed to produce any level of protection for ImageNet[5], as exemplified in the RobustML catalog where all ImageNet results are reduced to $\leq 0.1\%$ accuracy.[1] In contrast, we present a new, state-of-the-art defense for ImageNet that pays some cost to accuracy when not under attack, but achieves a Top-5 accuracy of up to 57.1% when under attack. These attacks are carried out by the strongest adversary we could construct, which is significantly stronger than those used in similar work in key respects.

In our work, we instead look not for a single transformation $t(\cdot)$, but propose to build a collection of many different transforms $t_{1,\ldots,n}$ from which we will randomly select a subset to apply to each image at both training and testing time. The individual transforms will be randomly parameterized as will the particular subset chosen and the order in which they are applied. By creating a barrage of random transformations, we show that such an ensemble defense can provide tangible benefits against attack, even after taking into account all of the methods by which obfuscated gradients can mislead us into using a broken defense [4].

Overall we provide the following contributions:

- A new, state-of-the-art defense on ImageNet, that fully accounts for the obfuscated gradients issue.
- Results that show ensembling weak defenses can create a strong defense, provided they are combined in a randomized fashion and the population of defenses is large. Prior work had conjectured that this was not the case [6].

BaRT is inspired by and builds upon a number of prior works that have used singular transformations to try to defend against attacks. We will review work related to our approach in section 2 and detail both the

[1] https://www.robust-ml.org/

BaRT strategy and it's constituent transformations in section 3, as well as the threat model of our adversary in section 4. While heuristic in nature, we find that after accounting for our strongest adversary we obtain state-of-the-art robustness against attack on the ImageNet dataset, which we show in section 5.

2. Related Work

While work on adversarial attacks against machine learning models has existed for over a decade, recent work that showed their success against neural networks [7, 8] has spawned increased motivation and attention to this problem. There were some who thought this concern was over stated, and that the number of variations in position, lighting, angle, and other factors that would occur in the real world would render adversarial attacks a non-issue for physical systems [9]. However, it was later shown that these difficulties could be circumvented making it possible for adversarial examples to be constructed [10, 11].

Still, the intuition that adjustments in angle, position, or other kinds of visual transformations of some object could defeat an adversary by somehow filtering or removing the adversary's perturbations was strong and alluring. As such, many papers have been presented that attempt to defeat adversaries using some kind of image pre-processing before classification (e.g., [12–14]). As far as we are aware, these types of defenses have all been defeated in the white-box threat model, either by correctly incorporating the defense into the adversary's search procedure [2], or by properly accounting for obfuscated gradients [4]. Obfuscated gradients occur when the defense has, intentionally or not, masked information about the gradient making it unreliable (or non-existent) for the adversary to use. These can occur in a number of ways, but all of which have proposed workarounds to obtain a suitable approximate gradient for the adversary to use [4]. In this work, we use only techniques which have already been defeated to build our defense. This way we can leverage known solutions to the obfuscated gradient and thus fully account for the problem and ensure our adversary's attack has full knowledge of the defense.

Few approaches have been able to scale up to ImageNet's size, and we find most works that have attempted to defend it against attack have been based on transformations or denoising. Prakash, Moran, Garber, *et al.* [15] claimed 81% accuracy under attack and Liao, Liang, Dong, *et al.* [16] 75%, but both were reduced to 0% under just $\epsilon = 4$ when obfuscated gradients were accounted for [17]. Xie, Zhang, Yuille, *et al.* [18] claimed 86% accuracy and Guo, Rana, Cissé, *et al.* [13] 75%, but these were later also reduced to 0% accuracy[4]. Even different approaches with more modest claims were later shown to be deficient, such as Kannan, Kurakin, and Goodfellow [19] who initially reported 27.9% accuracy but which was later demonstrated to be 0.1% [20].

Others before us have looked at building a multi-component defense, but prior work has reached the conclusion that a combined defense is no stronger than any of its constituent members [6]. In this paper we demonstrate that this is not necessarily true. Prior attempts at ensembling defenses have all combined their constituents in a fixed strategy, which has failed to be useful. In contrast, we demonstrate that a stochastic combination of weak defenses is effective.

A number of recent works have looked at developing provably secure training procedures for deep learning [21–23]. We believe that in the long term this is the most encouraging and desirable path toward defending against adversarial attacks. However, these methods are not yet usable for large datasets. The most recent work in this area has been "scaling up" to CIFAR-10 [24], which is orders of magnitude smaller than ImageNet.

The state-of-the-art defense that has been repeatedly found to be effective is Adversarial Training, which involves augmenting the training data with adversarially crafted examples generated as the training progresses [8]. Madry, Makelov, Schmidt, *et al.* [25] used adversarial training on the CIFAR dataset, which still has the best empirical robustness to attack [24] and has been repeatedly validated as effective and capable of fully defending against the best known adversaries under the white-box threat model [4]. Kurakin, Goodfellow, and Bengio [26] attempted to scale adversarial training up to the ImageNet dataset, which they found especially difficult. As far as we are aware, their work provides the best uncontested defense against adversarial attack on ImageNet. Against an adversary operating in the L_∞ distance, they obtain Top-1 and Top-5 accuracy of 1.5% and 5.5% for Top-1 and Top-5 respectively for a max perturbation of $\epsilon = 16$. We will show that our defense outperforms adversarial training across all $\epsilon \in [2, 16]$, and even continues to provide a robust defense up to $\epsilon = 32$. We are not aware of any prior work which has considered an L_∞ adversary given this wide of a range.

3. A Barrage of Random Transforms

Given the research that has been performed over the past year, it is clear that a single transformation of the input image is not sufficient to produce a reliable defense. We take the perspective that given an omnipotent adversary, randomness is one way to construct a decision process that the adversary can not trivially circumvent. The question then becomes: *is there a way to randomly pre-process images before they are classified*

by a CNN, such that accuracy is not obliterated and the adversary is unable to effectively operate?

Since we are working on images, we can make use of a plethora of pre-existing image transformation and preprocessing steps that have been developed by the computer vision community over the past several decades. We leverage these to create 10 groups $G_{1,\ldots,10}$ of transformations. Each group G_j will have some number of transforms $t(\cdot)$ contained within that group. We used a total of $n = 25$ different transforms $t_{1,\ldots,25}$, and denote the set of all transforms $T = \bigcup_{i=1}^{25} t_i$, and $\forall j, G_j \subset T$, with each group of transforms having no overlap ($G_j \cap G_k = \emptyset$ for $j \neq k$).

Each transform $t_i(\cdot)$ will have some parameters p_i that alter the behavior of the transform, and so by randomly selecting the values of p we can can have $t_i(x|p_i)$ produce many different outputs, introducing a stochastic component. This alone is not new, but we also have a collection of n different transforms to choose from. To further maximize the randomness, we select an ordering, or "permutation," π of k transforms to apply. The ordering π will change every time we attempt to use a model $f(\cdot)$, with the goal being that $f(x) = f(t_{\pi(1)}(t_{\pi(2)}(\ldots(t_{\pi(k)}(A(x))))))$.

The intuition is that by randomly selecting k out of n transforms, where each transform is itself randomized, and applying them in a random order, we create a defense that the adversary A can not easily defeat. We focused on this randomness on top of randomness because it provides a mechanism that the adversary can not easily deal with, even if they have perfect knowledge of all transformations t_i and the parameters p_i that alter their behavior. The space of possible actions is too large to find a single alteration $\widetilde{x} = A(x)$ such that the attacker will successfully induce an error by the model for all permutations π and parameterizations $p_{\pi(\ldots)}$.

The transforms we use are listed below. There are five singleton-groups (a group that has only one transform member, $|G_i| = 1$). When a group has more than one constituent transform, we randomly select a transform from the group to act as the group's representative, selecting a new representative on every application. This is to prevent the choice of multiple transformations which all have very similar effects from being applied at the same time, thereby increasing the diversity of changes made to each input.

We emphasize that for every individual transform we evaluate in this work, we have independently tested the transform and achieved 100% evasion success against it using the attack methodology outlined in subsection 4.1. As such, we know that all of these defenses are insufficient in isolation. Thus it is their stochastic combination that makes them significantly stronger than any constituent member. This is counter to previous conclusions that ensembling defenses are not effective and only as strong as the strongest individual defense in the ensemble [6]. The critical difference between our own and prior ensembling defense work is is the number of defenses (25 weak defenses, compared to ≤ 3 for most prior work), and the use of randomness to select subsets of defenses in random orderings.

We employ 25 transforms in total, and so only briefly describe the larger groups here. Further explanation, and Python code, are provided in the appendix.

Color Precision Reduction Reducing bit-resolution of color was originally proposed as a defense by Xu, Evans, and Qi [27] and later reduced to 0% effectiveness [4]. It works by simply reducing the number of bits used to represent the color space of an image, and was tested down to using just 1 bit of color. We incorporate this approach, and make the transform random in two ways. First, the number of colors will be reduced to a value selected from $\mathcal{U}[8, 200]$. Second, with 50% probability we choose between: 1) using the same number of colors for each channel, or 2) selecting a different random number of colors to be used by each channel.

JPEG Noise Using lossy JPEG compression to introduce artifacts was introduced by Kurakin, Goodfellow, and Bengio [10]. Their work looked at how different values of the JPEG compression level (a range from 1 to 100) reduced the impact of adversarial attacks for different values of $\epsilon \leq 16$. However, it was subsequently defeated, having 0% effectiveness [4]. When using this approach, we randomize it by selecting the compression level from $\mathcal{U}[55, 95]$.

Swirl We introduce a simple defense which is to apply a weak swirl to the image, rotating the pixels around a randomly selected point in the image. The radius of intensity is randomly selected from $\mathcal{U}[10, 200]$, and strength from $\mathcal{U}[0.1, 2.0]$.

Noise Injection In early work Tabacof and Valle [28] looked at the impact of addition Gaussian noise on adversarial attacks. We extend this by randomly selecting from Gaussian, Poisson, Salt, Pepper, Salt & Pepper, and Speckle noise to be inserted. With a 50% chance we will either: 1) apply the same noise to every channel, or 2) apply a randomly selected noise type to each channel independently.

FFT Perturbation We introduce a defense built around perturbing the 2D FFT of each channel of the

input image separately. In the frequency domain of the image, we scale all coefficients by a value sampled from $\mathcal{U}[0.98, 1.02]$ (used for all channels). Then for each channel, we randomly choose between 1) zeroing out random coefficients of the FFT, or 2) zeroing out the lowest frequency coefficients of the FFT. The proportion of coefficients that will be set to zero is a random value selected from $\mathcal{U}[0.0, 0.95]$. After altering the coefficients in the frequency domain we return a new, modified image in the spatial domain.

Zoom Group We consider two transforms that have the effect of zooming into the image. To prevent "over zooming" into the image, they are grouped and only one is selected from the group at each step. A simple zoom into a random portion of the image is done, similar to prior work [13], as well as a content-aware zoom based on seam carving [29].

Color Space Group We include four transforms that operate by altering the channels of the image by adding a random constant value, but provide larger impact by first converting the image from RGB to a different color space, and then converting back to RGB after modification. While a more difficult approach would be to allow every pixel in every color coordinate to receive a different value, we intentionally choose the simpler constant value to aid our adversary. This approach is applied to the HSV, XYZ, LAB, and YUV color spaces as the four transform members of this group.

Contrast Group We consider three different types of histogram equalization. Because each one attempts to re-scale and redistribute the values of the histogram of an image to broaden the covered range, they do not make sense to apply in a sequential manner. We use a simple version of Histogram Equalization, an adaptive variant called CLAHE [30], and an approach known as "contrast-stretching."

Grey Scale Group We as humans are usually able to recognize most objects from grey scale imagery, and as such, include conversion to grey scale as one of our defense techniques. For this reason we perform grey-scale transformation four different ways which can be applied selectively to different color channels.

Denoising Group The final group we consider is a number of classical denoising operations and transformations. We group them to avoid over-zealous application that can result in images which appear overly blurry and become difficult to interpret. This includes a Gaussian blur, median, mean, and mean-bilateral [31] filtering, Chambolle and wavelet [32] denoising, and non-local mean denoisng. Prior works have used the median filter [12], wavelet [15, 17], and non-local mean [27] as defenses, but all have since been defeated.

4. Methodology

Given the set of transformations outlined in section 3, we will use a ResNet50 model as our base architecture for experimentation. In particular, we will start with a pre-trained ResNet50 model, and perform an additional 100 epochs of training on ImageNet using Adam[33]. For each dataset in the batch, we randomly pick $k \sim \mathcal{U}[0, 5]$ transformations to apply to each image, so that the model is familiar with the transformations we apply at test time. Following Biggio, Fumera, and Roli [34], we will now fully state the threat model that we will operate in.

Once we have a trained model, our adversary will attack it in three ways: 1), reduce the Top-1 accuracy (any output besides the correct class is a success for the attacker); 2) reduce the Top-5 accuracy (any output is a success for the attacker provided the correct class is ranked sixth or lower), and; 3) increase the targeted success rate. In the first two conditions the attacker can trick the model into any incorrect classification. In the final condition, the attacker has a specific, randomly selected class that it must induce the model into outputting. All of these attacks will be performed on the standard ImageNet validation set.

Our adversary's capability will include making modifications to any input feature of the test data under the L_∞ metric, for which the adversary will attempt to modify the input x to a new input $\hat{x}$ such that $\|x - \hat{x}\|_\infty < \epsilon$. In our experiments, we will test a range of $\epsilon \in [2, 32]$.

We will operate in the white-box scenario, and assume that our adversary has full and complete knowledge of our training data, architecture, weights, and defensive transforms. To perform the attacks, we will use the Fast Gradient Sign Method (FGSM) [8] because it is a common baseline. More importantly, we will also use Projected Gradient Descent (PGD) [26], which is also targeted toward the L_∞ metric and is currently the strongest known attack for this metric. PGD has been conjectured to be a near-optimal first-order attack [25]. We use the FoolBox library for the implementation of these attacks [35].

To further ensure the attacker's strength, we follow recommendations from Demontis, Melis, Pintor, *et al.* [36], and attempt to optimize for the adversary in the L_∞ ball with *maximum confidence*, rather than minimum distance. This includes making sure that the PGD attack runs through all optimization steps, even

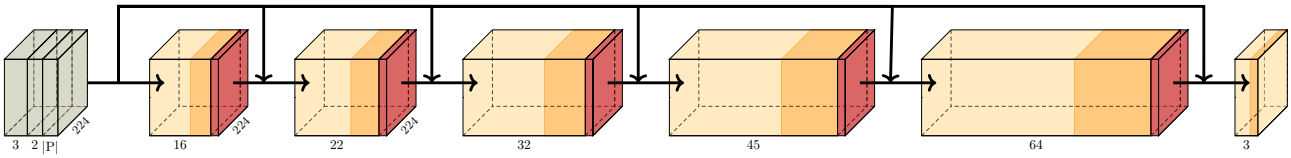

Figure 1: Diagram of BPDA network architecture. Input is of dimension $3+2+|P(t)|$, first three dimensions are the RGB channels, second two are the CoordConv channels, and the last set corresponds to the random parameters that affect the transform $t(\cdot)$'s output. Arrows that connect in the diagram indicate concatenation, yellow is convolution (number of filters below) followed by batch-normalization, and red is the ReLU activation

if the attack appears to have been successful at an earlier iteration. By default all experiments will perform PGD with 40 attack iterations, with stronger attacks in the appendix. Below we will further detail all the steps we take to implement the adversary's attack, so that we fully account for the gradient obfuscation and other issues that have thwarted previous defenses [2, 4].

4.1. Making A Strong Adversary

To maximize the strength of our attacker, we must first resolve two issues. The first is that our transformation process is randomized, which means we can not take the gradient from a single instance of the attack, as the next realization of a transformed image will have a differently parameterized transform. To remedy this situation, we use the Expectation over Transformation (EoT) [11]. The idea of EoT is to perform the transformation multiple times, and take the average gradient over several runs. When we use iterative attacks like PGD, this means for every iteration of the attack we will take the average of several transforms at that step in the attack.

The second issue we have is that not all of our transformations are differentiable. The solution to this problem was proposed by Athalye, Carlini, and Wagner [4], and is called Backward Pass Differentiable Approximation (BPDA). The idea is simple: when a transform $t(\cdot)$ is not itself differentiable, use a neural network to learn a function $f_t(\cdot)$ that approximates the transform. Since it is implemented with a neural network, $f_t(\cdot)$ is differentiable, and so we can use $\nabla f_t(\cdot)$ to obtain a gradient that is useful for the adversary as an approximation to $\nabla t(\cdot)$. This approach is effective, and using a naive identity function $f_t(x) = x$ is often sufficient to defeat many attacks. Indeed, it is enough to defeat most of our transforms individually. However, we learn a small CNN to approximate this gradient to maximize the adversary's advantage.

We also recognize that while we have a repertoire of transforms that are selected from at random, each transform t itself is randomized as well. Denoting the set of parameters of a transform t as $P(t)$, the transform is deterministic given a specific realization $p \sim P(t)$ of these transforms. To learn our model $f_t(\cdot)$, we will create an input that has $5 + |P(t)|$ channels, and is the same size as the image to be learned. The first three channels will be the RGB channels of the original image. The next two channels will be the CoordConv values proposed by Liu, Lehman, Molino, *et al.* [37], so that our networks can deal with location specific transformations. We found CoordConv necessary in our BPDA model to effectively approximate the Swirl transform. The remaining $|P(t)|$ channels will each have a constant value, which is the value of the realized parameters p. Placing the random values p each into their own distinct channel provides a mechanism for us to allow the network $f_t(\cdot)$ to learn the fully specified deterministic mapping. Each CNN $f_t(\cdot)$ has 6 convolutional layers, followed by batch-normalization and then a ReLU activation. For each layer we include a skip connection from the input, following the DenseNet approach. (See Figure 1). We train the network as a denoising auto-encoder, where the target is the parameterized transform of the image (i.e., the loss is $\|f_t(x,p) - t(x|p)\|_2^2$), with 100 epochs of training for all 25 BPDA networks. Once $f_t(\cdot)$ is trained, we perform BPDA by back-propagating through $f_t(\cdot)$ to the first 3 channels that correspond to the original image RGB values.

Combining BPDA and EoT as we have described above, we can defeat any of our transforms individually 100% of the time for both targeted and un-targeted attacks. This confirms that we have implemented these approaches appropriately, and have maximized the strength of our adversary.

As part of our evaluation, we also wish to address a concern raised by Madry, Makelov, Schmidt, *et al.* [25], which is the computational cost of a threat model. They argue that the strength of an adversary should be in some way computationally constrained, in the same manner that cryptographic problems are secure because we assume the adversary does not have the dramatic compute resources necessary to attack a given encryption scheme. Using 10 iterations of EoT combined with the iterative nature of PGD (40 optimization steps)

means we must perform 400 gradient calculations per attack, combined with the time to compute the image transformations and back-propagate through the additional BPDA networks. This takes about 48 hours per experiment given a workstation with 10 CPU cores and a Titan X GPU. We will also consider results with 40 EoT iterations — the highest we have observed in the literature — to evaluate if an even stronger adversary would be significantly more successfully, but these experiments required 240 hours *each* on a DGX-1. In total, the results presented in this paper consumed approximately 320 days on our DGX-1. We include tests at both of these EoT scales to help confirm our attack is robust, and simply increasing the number of iterations of the adversary does not dramatically change results. We also feel we are approaching a limit of reasonable compute for an adversary to have, and would be the largest barrier to replication if we pushed to even more attack iterations.

4.1.1. Medoid over Transformations

We take a moment to define a new type of attack to help ensure that we are not inadvertently engaging in accidental obfuscation of gradients. In particular, we note that the expectation over transformation approach uses the mean gradient over some transformation, shown more formally in Equation 1 where z_{EoT} is the number of iterations of the EoT sampling and $t^{(i)}$ is a deterministic realization of the transform t (i.e., the pseudo random number generator has been seeded with the value i to provide a deterministic result).

$$\nabla \mathbb{E}_{t^{(i)} \sim t} f(t^{(i)}(x)) \approx \frac{1}{z_{\text{EoT}}} \sum_{i=1}^{z_{\text{EoT}}} \nabla f(t^{(i)}(x)) \qquad (1)$$

One possible source of gradient obfuscation might be that the mean of the distribution does not exist or is not well defined. This notion comes from the recognition that in our larger framework t in Equation 1 corresponds to our entire randomized pipeline of selecting k transforms from $G_{1,\ldots,10}$. Our concern by analogy is that we could have a situation similar to the Cauchy distribution: The mean of the Cauchy distribution does not exist; every empirical mean is equally likely. However, one can successfully estimate the Cauchy distribution's position by instead using the median.

With this notion in mind, we include a new *Medoid over Transformations* (MoT) estimate, in which we use the medoid of the gradients of a sample of z_{MoT} transformations as our gradient estimate to perform attacks with.

$$\underset{i}{\text{argmin}} \sum_{j=1}^{z_{\text{MoT}}} \|\nabla f(t^{(i)}(x)) - \nabla f(t^{(j)}(x))\|_2^2 \qquad (2)$$

If we are accidentally performing gradient obfuscation by instead pushing information from the mean to the medoid — similar to the behavior of a Cauchy distribution — we would expect to see an increase in performance with the MoT attack compared to the EoT. Our results will confirm that this is not the case, as the MoT attack performs worse than the EoT attack. However, we include the results and attack description here to build further confidence that we have attempted to make the strongest attack possible.

5. Results

Few people have had their defense stand up to further testing due to a variety of issues related to obfuscated gradients and failing to fully account for all components of the defense when designing the white-box adversary. Fewer still have been able to scale their defensive techniques up to the ImageNet dataset. Now that we have defined our methodology to make sure we have accounted for both obfuscated gradients and ensure our adversary has fully captured the defense in their attack, we will show how we obtain new state-of-the-art results on the ImageNet dataset, as well as the associated costs in achieving such performance.

Kurakin, Goodfellow, and Bengio [26] provide the strongest results on the full ImageNet dataset that we are aware of. They do this with adversarial training, which they noted had great difficulty scaling up to the ImageNet corpus. At a maximum perturbation of $\epsilon = 16$, they achieved only a Top-1 accuracy of 1.5% and a Top-5 accuracy of 5.5% when under attack by PGD.

For BaRT, we will by default assume $\epsilon = 16$, the number of transformations $k = 5$, and the number of EoT runs will be 10. In our experiments we will investigate changing all of the values to observe their impact on our effectiveness against attack. We begin by showing the accuracy of our methods in Table 1. Here we can see the first immediate down side to BaRT, which is a significant reduction in accuracy if our model is not under attack. The off-setting benefit is the first significant improvement in accuracy when the model is under attack. At a cost of increased runtime, it is possible to create multiple inferences of an input by applying the transform $t(\cdot)$ multiple times, and then classifying each differently transformed version of the image. This creates an ensemble effect, and removes any loss in accuracy due to BaRT's application. Due to space, details on ensembling BaRT are left to Appendix E.

5.1. Experiments

The immediate product of our work is that the BaRT strategy provides a 9.3–24 times improvement in accuracy on ImageNet compared to prior state of the art.

Table 1: Accuracy (%) of baseline prior work on adversarial training [26] and BaRT. 'Clean Images' is the results of classifying non-attacked images without any transforms; 'Attacked' shows results when using PGD with $\epsilon = 16$.

	Clean Images		Attacked	
Model	Top-1	Top-5	Top-1	Top-5
Inception v3	78	94	0.7	4.4
Inception v3 w/Adv. Train	78	94	1.5	5.5
ResNet50	76	93	0.0	0.0
ResNet50-BaRT, $k = 5$	65	85	16	51
ResNet50-BaRT, $k = 10$	65	85	36	57

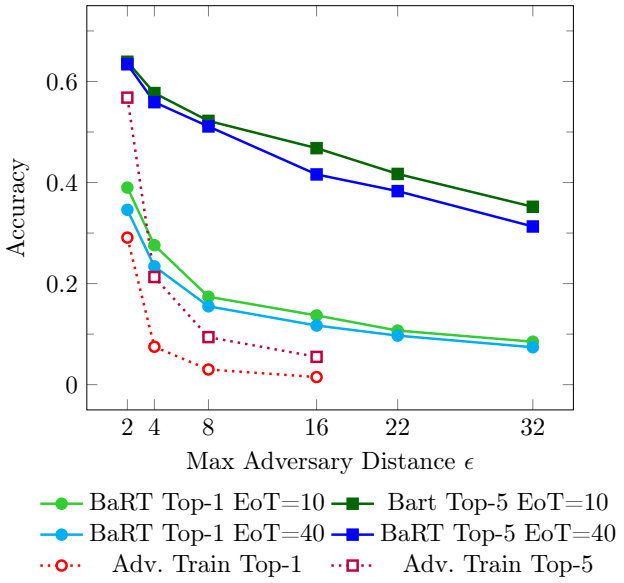

Figure 2: Accuracy of model under attack by PGD for varying adversarial distances ϵ with EoT steps={10,40}.

We now further investigate the differing parameters of our defense. First we look at the larger range of ϵ, the bound on the adversary's freedom to alter the input. In Figure 2 we show accuracy under PGD attack as ϵ varies from 2 to 32. We note that $\epsilon = 16$ is the largest we have observed in any prior work, and is considered a powerful adversary. We are the first to test $\epsilon = 32$, and still show non-trivial robustness to attack.

In these results we see that BaRT dominates adversarial training across all values of ϵ. We also see that adversarial training degrades quickly as ϵ moves from just 2 to 4. In contrast BaRT Top-1 and Top-5 accuracy when attacked with $\epsilon = 32$ is still better than the results with adversarial training and $\epsilon = 4$. For $\epsilon > 2$, BaRT also shows Top-1 accuracy higher than adversarial training's Top-5 accuracy.

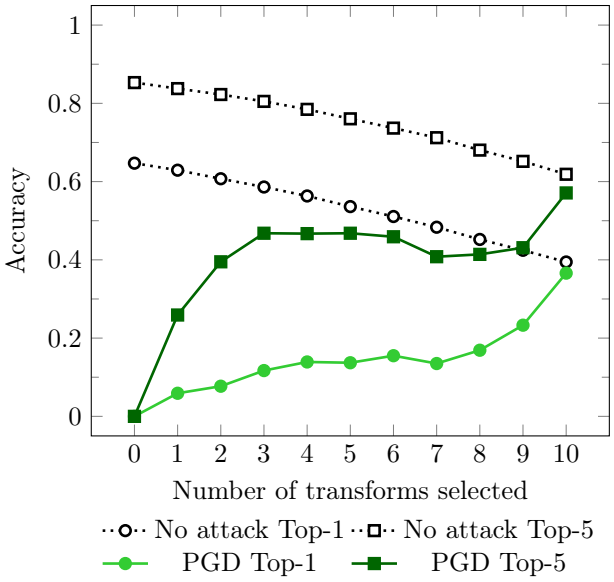

Figure 3: Accuracy of model when varying the number of transforms used, both when not under attack and when being attacked by PGD.

These results also demonstrate that while increasing the number of EoT steps does increase the adversary's success rate, the difference is not large. Using 40 steps already requires a level of compute not reasonable for most institutions, and gives us confidence that other attempts to simply throw even more compute to the adversary will be nonviable. This is before we consider that it is relatively easy to write these transformations, and we could add even more transformations to the pipeline to further impede the adversary's compute requirements and reduce their success rate. While we do not have the resources to test exhaustively, we show in Appendix F that using even 520 PGD steps shows no significant change in the attacker's success rate.

Next we investigate the number of transforms applied. For these results, we remind the reader that BaRT was only trained with up to $k = 5$ transforms applied to the training data. In Figure 3 we plot the Top-1 and Top-5 accuracy of BaRT (under attack and on clean images) as a function of the number of transforms k selected at test time. Our initial expectation was that we would see the best performance (i.e., greatest accuracy under attack) when $k = n/2$, as this would maximize the number of combinatorial paths $\binom{n}{k}$. However, this was not the case.

Instead we see that every transformation $t_i(\cdot)$ we apply produces some associated costs and benefits. The benefit is that increasing $k \to n$ improves our performance when under attack. A slight dip occurs after

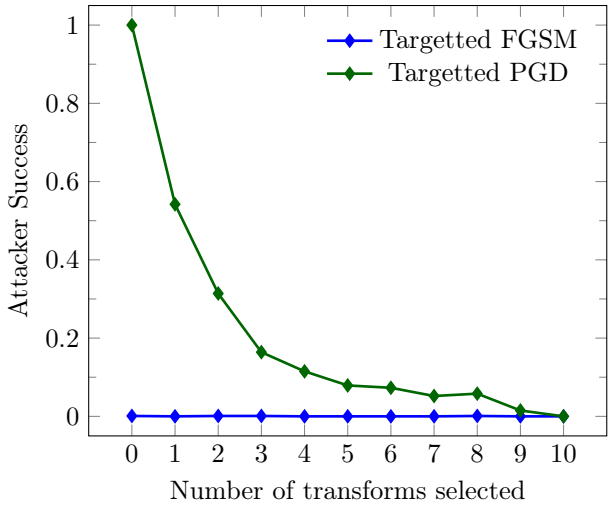

Figure 4: Attacker success rate against BaRT model when varying the number of transforms used, for both FGSM and PGD attacks with $\epsilon = 16$.

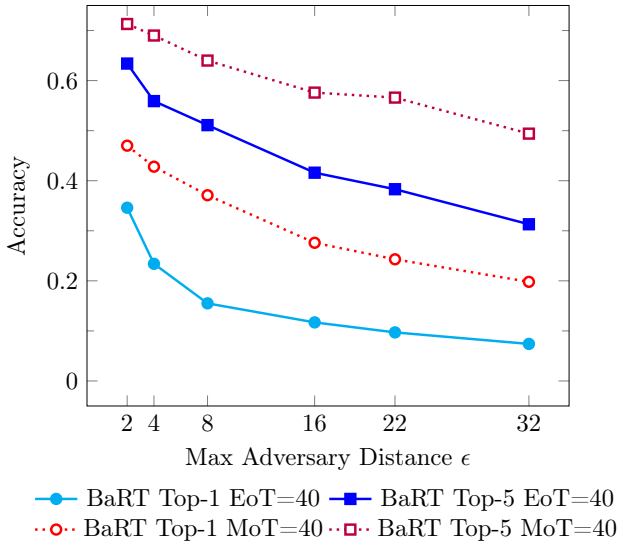

Figure 5: Accuracy of model under attack by EoT and MoT versions of PGD for varying adversarial distances ϵ and EoT steps=40.

$k = 5$ transformations are applied, which we expect is related to using $k \leq 5$ during training. One can also observe a steady decrease in the accuracy on non-attacked, clean images as k increases, which happen to almost intersect at $k = 10$. Adding more transformations also has an impact on run-time, but calculating the transformations is fast relative to the cost of needing a GPU for CNN inference, and is approximately three orders of magnitude faster than running the attacks.

Overall this validates that an ensemble of weak defenses can form a single strong defense, provided that the ensemble is applied in a random fashion. We also see that maximizing the combinatorial search space is not a dominating strategy, since we see maximal adversarial robustness at $k = 10$ instead of $k = 5$. This tells us that the amount of transformation applied to the image is also an important component of defeating the adversary, as this is maximized at $k = 10$. Cumulatively, we could argue that selecting the value of k to use in practice should be a function of the likelihood of being under attack. If a model is continuously under attack or needs maximal worst-case performance, one should choose $k = 10$ because the non-attacked accuracy is not meaningful when under attack.

We also explore the impact on targeted adversarial attacks in Figure 4, where we look at the attacker's success rate as a function of k. Here we can see that when no transformations are present, PGD attack achieves 100% success rate against the model, but quickly degrades as transformations are added — reaching 0.0% success at $k = 10$ transformations. We note that additional runs may produce values near zero instead of at zero, but it suffices to show that the ability for the adversary to perform targeted attacks can be almost completely impeded by our BaRT defense.

Lastly, in subsubsection 4.1.1 we considered the possibility that we might be engaging in obfuscated gradients by moving information to the medoid of the distribution. We developed a new Medoid over Transformation attack to test this hypothesis. The results are shown in Figure 5. While MoT does produce adversarial examples, it has uniformly worse performance compared to using the mean gradient. As such we further conclude that we have not relied on obfuscated gradients, and that our defense is effective.

6. Conclusion

We have introduced BaRT, a strategy for defending image classifiers against attack by randomly selecting a few transforms from a large pool of stochastic transformations, and apply each in a random order before processing the image. This scales to datasets like ImageNet, and provides state-of-the-art results when under attack even after accounting for all known obfuscated gradients. While heuristic in nature, our results provide evidence that a strong defense can be made from many weaker ones, and indicates strategic applications of randomness may benefit future work.

Acknowledgments We would like to thank Battista Biggio for reviewing a draft of this work and providing insightful comments and feedback.

References

[1] B. Biggio and F. Roli, "Wild patterns: Ten years after the rise of adversarial machine learning," *Pattern Recognition*, vol. 84, pp. 317–331, Dec. 2018. DOI: 10.1016/j.patcog.2018.07.023.

[2] N. Carlini and D. Wagner, "Adversarial Examples Are Not Easily Detected: Bypassing Ten Detection Methods," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, ser. AISec '17, New York, NY, USA: ACM, 2017, pp. 3–14. DOI: 10.1145/3128572.3140444.

[3] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust Physical-World Attacks on Deep Learning Models," in *Computer Vision and Pattern Recognition (CVPR)*, 2018.

[4] A. Athalye, N. Carlini, and D. Wagner, "Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples," in *International Conference on Machine Learning (ICML)*, 2018.

[5] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision (IJCV)*, vol. 115, no. 3, pp. 211–252, 2015. DOI: 10.1007/s11263-015-0816-y.

[6] W. He, J. Wei, X. Chen, N. Carlini, and D. Song, "Adversarial Example Defenses: Ensembles of Weak Defenses Are Not Strong," in *Proceedings of the 11th USENIX Conference on Offensive Technologies*, ser. WOOT'17, Berkeley, CA, USA: USENIX Association, 2017.

[7] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," in *ICLR*, 2014. DOI: 10.1021/ct2009208.

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," in *International Conference on Learning Representations (ICLR)*, 2015.

[9] J. Lu, H. Sibai, E. Fabry, and D. Forsyth, "No Need to Worry about Adversarial Examples in Object Detection in Autonomous Vehicles," in *The First Workshop on Negative Results in Computer Vision. CVPR 2017.*, 2017.

[10] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," in *ICLR Workshop*, 2017.

[11] A. Athalye, L. Engstrom, A. Ilyas, and K. Kwok, "Synthesizing Robust Adversarial Examples," 2017.

[12] X. Li and F. Li, "Adversarial Examples Detection in Deep Networks with Convolutional Filter Statistics," in *2017 IEEE International Conference on Computer Vision (ICCV)*, IEEE, Oct. 2017, pp. 5775–5783. DOI: 10.1109/ICCV.2017.615.

[13] C. Guo, M. Rana, M. Cissé, and L. Van Der Maaten, "Countering Adversarial Images Using Input Transformations," in *International Conference on Learning Representations (ICLR)*, 2018.

[14] P. Samangouei, M. Kabkab, and R. Chellappa, "Defense-GAN: Protecting Classifiers Against Adversarial Attacks Using Generative Models," in *International Conference on Learning Representations (ICLR)*, 2018.

[15] A. Prakash, N. Moran, S. Garber, A. DiLillo, and J. Storer, "Deflecting Adversarial Attacks with Pixel Deflection," in *CVPR*, 2018. DOI: 10.1109/CVPR.2018.00894.

[16] F. Liao, M. Liang, Y. Dong, T. Pang, X. Hu, and J. Zhu, "Defense against Adversarial Attacks Using High-Level Representation Guided Denoiser," in *CVPR*, 2018.

[17] A. Athalye and N. Carlini, "On the Robustness of the CVPR 2018 White-Box Adversarial Example Defenses," *arXiv*, 2018.

[18] C. Xie, Z. Zhang, A. L. Yuille, J. Wang, and Z. Ren, "Mitigating Adversarial Effects Through Randomization," in *International Conference on Learning Representations (ICLR)*, 2018.

[19] H. Kannan, A. Kurakin, and I. Goodfellow, "Adversarial Logit Pairing," *arXiv*, 2018. DOI: 10.4103/0972-124X.94617.

[20] L. Engstrom, A. Ilyas, and A. Athalye, "Evaluating and Understanding the Robustness of Adversarial Logit Pairing," *arXiv*, 2018.

[21] E. Wong and Z. Kolter, "Provable Defenses against Adversarial Examples via the Convex Outer Adversarial Polytope," in *Proceedings of the 35th International Conference on Machine Learning*, J. Dy and A. Krause, Eds., ser. Proceedings of Machine Learning Research, vol. 80, Stockholmsmässan, Stockholm Sweden: PMLR, 2018, pp. 5283–5292.

[22] M. Abbasi and C. Vision, "Certified Defenses Against Adversarial Examples," in *International Conference on Learning Representations (ICLR)*, 2018.

[23] K. Dvijotham, R. Stanforth, S. Gowal, T. Mann, and P. Kohli, "A Dual Approach to Scalable Verification of Deep Networks," in *Conference on Uncertainty in Artificial Intelligence (UAI)*, 2018.

[24] E. Wong, F. Schmidt, J. H. Metzen, and J. Z. Kolter, "Scaling provable adversarial defenses," *ArXiv e-prints*, 2018.

[25] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards Deep Learning Models Resistant to Adversarial Attacks," in *ICLR*, 2018.

[26] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial Machine Learning at Scale," in *International Conference on Learning Representations (ICLR)*, 2017.

[27] W. Xu, D. Evans, and Y. Qi, "Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks," in *Proceedings 2018 Network and Distributed System Security Symposium*, Reston, VA: Internet Society, 2018. DOI: 10.14722/ndss.2018.23198.

[28] P. Tabacof and E. Valle, "Exploring the space of adversarial images," in *2016 International Joint Conference on Neural Networks (IJCNN)*, IEEE, Jul. 2016, pp. 426–433. DOI: 10.1109/IJCNN.2016.7727230.

[29] S. Avidan and A. Shamir, "Seam Carving for Content-aware Image Resizing," in *ACM SIGGRAPH 2007 Papers*, ser. SIGGRAPH '07, New York, NY, USA: ACM, 2007. DOI: 10.1145/1275808.1276390.

[30] K. Zuiderveld, "Contrast Limited Adaptive Histogram Equalization," in *Graphics Gems IV*, P. S. Heckbert, Ed., San Diego, CA, USA: Academic Press Professional, Inc., 1994, ch. Contrast L, pp. 474–485.

[31] B. Weiss, "Fast Median and Bilateral Filtering," in *ACM SIGGRAPH 2006 Papers*, ser. SIGGRAPH '06, New York, NY, USA: ACM, 2006, pp. 519–526. DOI: 10.1145/1179352.1141918.

[32] S. Chang, Bin Yu, and M. Vetterli, "Adaptive wavelet thresholding for image denoising and compression," *IEEE Transactions on Image Processing*, vol. 9, no. 9, pp. 1532–1546, 2000. DOI: 10.1109/83.862633.

[33] D. P. Kingma and J. L. Ba, "Adam: A Method for Stochastic Optimization," in *International Conference On Learning Representations*, 2015.

[34] B. Biggio, G. Fumera, and F. Roli, "Security evaluation of pattern classifiers under attack," *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 4, pp. 984–996, 2014. DOI: 10.1109/TKDE.2013.57.

[35] J. Rauber, W. Brendel, and M. Bethge, "Foolbox: A Python toolbox to benchmark the robustness of machine learning models," *arXiv preprint arXiv:1707.04131*, 2017.

[36] A. Demontis, M. Melis, M. Pintor, M. Jagielski, B. Biggio, A. Oprea, C. Nita-Rotaru, and F. Roli, "Why Do Adversarial Attacks Transfer? Explaining Transferability of Evasion and Poisoning Attacks," *ArXiv e-prints*, pp. 26–28, 2018.

[37] R. Liu, J. Lehman, P. Molino, F. P. Such, E. Frank, A. Sergeev, and J. Yosinski, "An Intriguing Failing of Convolutional Neural Networks and the CoordConv Solution," pp. 1–24, 2018.

[38] A. Chambolle, "An Algorithm for Total Variation Minimization and Applications," *J. Math. Imaging Vis.*, vol. 20, no. 1-2, pp. 89–97, Jan. 2004. DOI: 10.1023/B:JMIV.0000011325.36760.1e.

[39] J. Darbon, A. Cunha, T. F. Chan, S. Osher, and G. J. Jensen, "Fast nonlocal filtering applied to electron cryomicroscopy," in *2008 5th IEEE International Symposium on Biomedical Imaging: From Nano to Macro*, IEEE, May 2008, pp. 1331–1334. DOI: 10.1109/ISBI.2008.4541250.

[40] L. I. Kuncheva and C. J. Whitaker, "Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy," *Machine learning*, vol. 51, no. 2, pp. 181–207, 2003.

[41] C. Ju, A. Bibaut, and M. van der Laan, "The relative performance of ensemble methods with deep convolutional neural networks for image classification," *Journal of Applied Statistics*, vol. 45, no. 15, pp. 2800–2818, 2018.