BAMBOO: BALL-SHAPE DATA AUGMENTATION AGAINST ADVERSARIAL ATTACKS FROM ALL DIRECTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural networks (DNNs) are widely adopted in real-world cognitive applications because of their high accuracy. The robustness of DNN models, however, has been recently challenged by adversarial attacks where small disturbance on input samples may result in misclassification. State-of-the-art defending algorithms, such as adversarial training or robust optimization, improve DNNs' resilience to adversarial attacks by paying high computational costs. Moreover, these approaches are usually designed to defend one or a few known attacking techniques only. The effectiveness to defend other types of attacking methods, especially those that have not yet been discovered or explored, cannot be guaranteed. This work aims for a general approach of enhancing the robustness of DNN models under adversarial attacks. In particular, we propose *Bamboo* – the first data augmentation method designed for improving the general robustness of DNN without any hypothesis on the attacking algorithms. *Bamboo* augments the training data set with a small amount of data uniformly sampled on a fixed radius ball around each training data and hence, effectively increase the distance between natural data points and decision boundary. Our experiments show that *Bamboo* substantially improve the general robustness against arbitrary types of attacks and noises, achieving better results comparing to previous adversarial training methods, robust optimization methods and other data augmentation methods with the same amount of data points.

1 INTRODUCTION

In recent years, thanks to the availability of large amounts of training data, deep neural network (DNN) models (e.g., convolutional neural networks (CNNs)) have been widely used in many real-world applications such as handwritten digit recognition (LeCun et al. (1998)), large-scale object classification (Simonyan & Zisserman (2014)), human face identification (Chen et al. (2016)) and complex control problems (Mnih et al. (2013)). Although DNN models have achieved close to or even beyond human performance in many applications, they exposed a high sensitivity to input data samples and therefore are vulnerable to the relevant attacks. For example, *adversarial attacks* apply a "small" perturbation on input samples, which is visually indistinguishable by humans but can result in the misclassification of DNN models. Several attacking algorithms have been also proposed, including FGSM (Szegedy et al. (2013)), DeepFool (Moosavi Dezfooli et al. (2016)), CW (Carlini & Wagner (2017)) and PGD (Madry et al. (2018)) etc., indicating a serious threat against the systems using DNN models.

Many approaches have also been proposed to defend against adversarial attacks. *adversarial training*, for example, adds the classification loss of certain known adversarial examples into the total training loss function: Goodfellow et al. (2014) use the FGSM noise for adversarial training and Madry et al. (2018) use the PGD noise as the adversaries. These approaches can effectively improve the model's robustness against a particular attacking algorithm, but won't guarantee the performance against other kinds of attacks (Carlini & Wagner (2017)). *Optimization based methods* take the training process as a *min-max* problem and minimize the loss of the worst possible adversarial examples, such as what were done by Sinha et al. (2017) and Yan et al. (2018). The approach can increase the margin between training data points and the decision boundary along some directions. However, solving the *min-max* problem on-the-fly generates a high demand for the computational load. For large models like VGG (Simonyan & Zisserman (2014)) and ResNet (He et al. (2016)), optimizing the *min-max* problem could be extremely difficult. A large gap exists between previously proposed algorithms aiming for defending against adversarial attacks and the goal of efficiently improving the overall robustness of DNN models without any hypothesis on the attacking algorithms.

Generally speaking, defending against adversarial attacks can be considered as a special case of increasing the generalizability of machine learning models to unseen data points. *Data augmentation method*, which is originally proposed for improving the model generalizability, may also be effective to improve the DNN robustness against adversarial attacks. Previous studies show that augmenting the original training set with shifted or rotated version of the original data can make the trained classifier robust to shift and rotate transformations of the input (Simard et al. (1998)). Training with additional data sampled from a Gaussian distribution centered at the original training data can also effectively enahnce the model robustness against natural noise (Chapelle et al. (2001)). The recently proposed *Mixup* method (Zhang et al. (2017)) augmented the training set with linear combinations of the original training data and surprisingly improved the DNN robustness against adversarial attacks. Although these data augmentation methods inspired our work, they may not offer the most efficient way to enhance the adversarial robustness of DNN as they are not designated to defend adversarial attacks.

In this work, we propose *Bamboo*—a ball shape data augmentation technique aiming for improving the general robustness of DNN against adversarial attacks from *all directions*. *Bamboo* augments the training set with data uniformly sampled on a fixed radius ball around each training data point. Our theoretical analysis shows that without requiring any prior knowledge of the attacking algorithm, training the DNN classifier with our augmented data set can effectively enhance the general robustness of the DNN models against the adversarial noise. Our experiments show that *Bamboo* offers a significantly enhanced model robustness comparing to previous robust optimization methods, without suffering from the high computational complexity of these prior works. Comparing to other data augmentation method, *Bamboo* can also achieve further improvement of the model robustness using the same amount of augmented data. Most importantly, as our method makes no prior assumption on the distribution of adversarial examples, it is able to work against all kinds of adversarial and natural noise. To authors' best knowledge, *Bamboo* is the first data augmentation method specially designed for improving the general robustness of DNN against all directions of adversarial attacks and noise.

The remaining of the paper is organized as follows. Section 2 explains how to measure model robustness and summaries previous research on DNN robustness improvement; In Section 3, we elaborate *Bamboo*'s design principle and the corresponding theoretical analysis. Section 4 empirically discusses the parameter selection and performance of our method and compares it with some related works; At the end, we conclude the paper and discuss the future work in Section 5.

2 Background

2.1 Measurement of DNN robustness

2.1.1 Robustness under gradient based attack

A metric for measuring the robustness of the DNN is necessary. Szegedy et al. (2013) propose the fast gradient sign method (FGSM) noise, which is one of the most efficient and most commonly applied attacking method. FGSM generates an adversarial example x' using the sign of the local gradient of the loss function J at a data point x with label y as shown in Equation (1):

$$x' = x + \epsilon \, \text{sign}(\nabla_x J(\theta, x, y)), \tag{1}$$

where ϵ controls the strength of FGSM attack. For its high efficiency in noise generation, the classification accuracy under the FGSM attack with certain ϵ has been taken as a metric of the model robustness.

As FGSM attack leverages only the local gradient for perturbing the input, *gradient masking* (Papernot et al. (2016)) that messes up the local gradient can effectively improve the accuracy under FGSM attack. However, gradient masking has little effect on the decision boundary, so it may not increase the actual robustness of the DNN. In other words, even a DNN model achieves high accuracy under FGSM attack, it may still be vulnerable to other attacking methods. Madry et al. (2018) propose projected gradient descent (PGD), which attacks the input with multi-step variant FGSM that is projected into certain space $x + \mathcal{S}$ at the vicinity of data point x for each step. Equation (2) demonstrates a single step of the PGD noise generation process.

$$x^{t+1} = \Pi_{x+\mathcal{S}}(x^t + \epsilon\,\text{sign}(\nabla_x J(\theta, x, y))). \tag{2}$$

Madry et al. (2018)'s work shows that comparing to FGSM, adversarial training using PGD adversarial is more likely to lead to a universally robust model. Therefore the classification accuracy under the PGD attack would also be an effective metric of the model robustness.

2.1.2 Robustness under optimization based attack

Besides these gradient based methods, the generation of adversarial examples can also be viewed as an optimization process. In this work, we mainly focus on *untargeted* optimization-based attacks. Szegedy et al. (2013) describe the general objective of such attacks as Equation (3):

$$\begin{aligned} &minimize_\delta \; D(x, x+\delta),\\ &\quad s.t. \; C(x+\delta) \neq C(x), \; x+\delta \in [0,1]^n. \end{aligned} \tag{3}$$

Where D is the distance measurement, most commonly the Euclidean distance; and C is the classification result of the DNN. The optimization objective is to find an adversarial example $x' = x + \delta$ that results in misclassification by paying the minimum distance to the original data point x.

Note that the objective in Equation (3) includes the classification function of DNN as a constraint. Due to the nonlinearity and the nonconvexity of the DNN classifier, the objective in Equation (3) can not be easily optimized. In order to generate strong optimization-based attacks more efficiently, CW attack (Carlini & Wagner (2017)) was proposed which defines a objective function f such that $C(x+\delta) \neq C(x)$ if and only if $f(x+\delta) \leq 0$. Several possible choices of function f are provided in Carlini & Wagner (2017)'s work, which are better suited for optimization comparing to the original constraint with DNN classification function C. With the use of f, the optimization problem in Equation (3) can be modified to:

$$\begin{aligned} &minimize_\delta \; D(x, x+\delta),\\ &\quad s.t. \; f(x+\delta) \leq 0, \; x+\delta \in [0,1]^n. \end{aligned} \tag{4}$$

It can be equivalently formulated as:

$$\begin{aligned} &minimize_\delta \; D(x, x+\delta) + c \cdot f(x+\delta),\\ &\quad s.t. \; x+\delta \in [0,1]^n. \end{aligned} \tag{5}$$

where c is a positive constant. The objective in Equation (5) can be optimized more easily than that in Equation (3), leading to a higher chance of finding the optimal δ efficiently (Carlini & Wagner (2017)). Carlini & Wagner (2017) successfully demonstrate that most of the previous works with high performance under FGSM attack would not be robust under their CW attack.

Since the objective of CW attack is to find the minimal possible perturbation strength of a successful attack, the resulted δ will point to the direction of the nearest decision boundary around x, and its strength can be considered as an estimation of the distance between the testing data point and the decision boundary. Therefore the average strength required for a successful CW attack can be considered as a reasonable measurement of the model robustness.

In this work, we will use the average strength of untargeted CW noise across all the data in testing set as the metric of robustness when demonstrating the effect of parameter tuning on our proposed method. Both the average CW strength and the testing accuracy under different strengths of FGSM and PGD attacks are taken as the metrics when comparing our method to previous works.

2.2 Previous works towards increasing network robustness

One of the most straightforward ways of analyzing and improving the robustness of a DNN is to formulate the robustness with key factors, such as the shape of the decision boundary or parameters and weights of the DNN model. Fawzi et al. (2016)'s work empirically visualizes the shape of the decision boundary, the observation of which shows that the curvature of the boundary tends to be lower when close to the training data points. This technique isn't very helpful in practice, mainly due to the difficulty in drawing a theoretical relationship between the decision boundary curvature and the DNN robustness that can effectively guide the DNN training. Some other works try to derive a bound of the DNN robustness from the network weights (Peck et al. (2017), Hein & Andriushchenko (2017)). These obtained bounds are often too loose to be used as a guideline for robust training, or too complicated to be considered as a factor in the training objective.

A more practical approach is *adversarial training*. For example, we can generate adversarial examples from the training data and then include their classification loss to the total loss function of the training process. As the generation of the adversarial examples usually relies on existing attacking techniques like FGSM (Goodfellow et al. (2014)), DeepFool (Yan et al. (2018)) or PGD (Madry et al. (2018)), the method can be efficiently optimized for the limited types of *known* adversarial attacks. However, it may not promise the robustness against other attacking methods, especially those newly proposed ones. Alternatively, the defender may online generate the worst-case adversarial examples of the training data and minimize the loss of such adversarial examples by solving a *min-max* optimization problem during the training process. For instance, the distributional robustness method (Sinha et al. (2017)) trains the weight θ of a DNN model so as to minimize the loss L of adversarial example x' which is near to original data point x but has supremum loss, such as

$$minimize_{\theta}\ F(\theta) := \mathbb{E}[sup_{x'}\{L(\theta; x') - \gamma D(x', x)\}], \tag{6}$$

where γ is a positive constant that tradeoffs between the strength and effectiveness of the generated x'. This method can achieve some robustness improvement, but suffers from high computational cost for optimizing both the network weight and the potential adversarial example. Also, this work only focuses on small perturbation attacks, so the robustness guarantee may not hold on the improvement of robustness under large attacking strength (Sinha et al. (2017)).

3 Proposed Approach

3.1 Vicinity risk minimization against adversarial attacks

Most of the supervised machine learning algorithms, including the ordinary training process of DNNs, follow the principle of *empirical risk minimization* (ERM). ERM tends to minimize the total risk R on the training set data, as stated in Equation (7):

$$minimize_{\theta}\ R(\theta) := \mathbb{E}_{(x,y)\sim P(x,y)} L(f(x, \theta), y), \tag{7}$$

where $f(\cdot, \theta)$ is the machine learning model with parameter θ, L is the loss function and $P(x, y)$ is the joint distribution of data and label in the training set.

Such an objective is based on the hypothesis that the testing data has a similar distribution as the training data, so minimizing the loss on the training data would naturally lead to the minimum testing loss. DNN, as a sufficiently flexible machine learning model, can be well optimized towards this objective and *memorize* the training set distribution (Zhang et al. (2016)). However, the distribution of adversarial examples generated by attacking algorithms may be different from the original training data. Thus the memorization of DNN models would lead to unsatisfactory performance on adversarial examples (Goodfellow et al. (2014)).

As our work aims to improve the model robustness against adversarial attacks, we propose to follow the principle of *vicinity risk minimization* (VRM) instead of ERM during the training process. Firstly proposed by Chapelle et al. (2001), the VRM principle targets to minimize the *vicinity risk* $\hat{R}$ on the *virtual data pair* $(\hat{x}, \hat{y})$ sampled from a *vicinity distribution* $\hat{P}(\hat{x}, \hat{y}|x, y)$ generated from the original training set distribution $P(x, y)$. Consequently, the optimization objective of the VRM-based training can be described as:

$$minimize_{\theta}\ \hat{R}(\theta) := \mathbb{E}_{(\hat{x},\hat{y})\sim \hat{P}(\hat{x},\hat{y}|x,y),(x,y)\sim P(x,y)} L(f(\hat{x}, \theta), \hat{y}). \tag{8}$$

For the choice of vicinity distribution, they use Gaussian distribution centered at original training data, which makes the model more robust to natural noise (Chapelle et al. (2001)).

Now we consider improving the robustness against adversarial attacks. It would be easier to detect and defense the adversarial attacks if the strength of the perturbation is large, therefore most of the

Algorithm 1: *Bamboo*: Ball-shape data augmentation

```
Input  : Augmentation ratio $N$, Ball radius $r$, Original training set $(X, Y)$
Output: Augmented training set $(\hat{X}, \hat{Y})$
$n$ := length($X$);
$\hat{X} := X, \hat{Y} := Y$;                    ▷ Initializing augmented dataset with original training set
$count := n$;
for $i = 1 : n$ do
    $x := X[i], y := Y[i]$;
    for $j = 1 : N$ do
        $count := count + 1$;
        Sample $\delta \sim \mathcal{N}(0, \mathcal{I})$;     ▷ $\mathcal{N}$ refers to normal distribution and $\mathcal{I}$ is identity matrix
        $\delta_r := \frac{\delta}{||\delta||_2} \cdot r$;                 ▷ Normalizing the length of $\delta$
        $\hat{X}[count] := x + \delta_r, \hat{Y}[count] := y$;   ▷ Augmenting the data into training set
    end
end
return $(\hat{X}, \hat{Y})$
```

attacking algorithms will apply a constraint on the strength of the perturbation. So the adversarial example $\hat{x}$ can be considered as a point within a r-radius ball around the original data x. Without any prior knowledge of the attacking algorithm, we can consider the adversarial examples as uniformly distributed within the r-radius ball: $\hat{x} \sim Uniform(||\hat{x} - x||_2 \leq r)$. Following the VRM principle, we can improve the robustness against adversarial attacks by optimizing the objective in Equation (8) with vicinity distribution:

$$\hat{P}(\hat{x}, \hat{y}|x, y) = Uniform(||\hat{x} - x||_2 \leq r) \cdot \delta(\hat{y}, y). \tag{9}$$

However, the input space of DNN model is usually high dimensional. Directly sampling the virtual data point $\hat{x}$ within the r-radius ball may be data inefficient. Here we propose to further improve the data efficiency by utilizing the geometry analysis of DNN model. Previous research shows that the curvature of DNN's decision boundary near a training data point would most likely be very small (Fawzi et al. (2016)), and the DNN model tends to behave linearly, especially at the vicinity of training data points (Goodfellow et al. (2014), Fawzi et al. (2016)). These observations indicate that the objective of minimizing the loss of data points sampled *within* the ball can be approximated by minimizing the loss of data points sampled *on* the edge of the ball. Formally, the vicinity distribution in Equation (9) can be modified to:

$$\hat{P}(\hat{x}, \hat{y}|x, y) = Uniform(||\hat{x} - x||_2 = r) \cdot \delta(\hat{y}, y). \tag{10}$$

By optimizing the VRM objective in Equation (8) with this vicinity distribution, we can improve the robustness of DNN against adversarial attacks with higher data efficiency in sampling the virtual data points for augmentation.

3.2 *Bamboo* AND ITS INTUITIVE EXPLANATION

As explained in the previous section, minimizing the loss of data points uniformly sampled on the edge of a r-radius ball around each point in the training set likely leads to a more robust DNN model against adversarial attacks. So we propose *Bamboo*, a ball-shape data augmentation scheme that augments the training set with N virtual data points uniformly sampled from a r-radius ball centered at each original training data point. In practice, for each data point in the training data, we first sample N perturbations from a Gaussian distribution with zero mean and identity matrix as variance matrix. Then we normalize the l_2 norm of each perturbation to r. Following the symmetric property of Gaussian distributions, the normalized perturbations should be uniformly distributed on a r-radius ball. Finally we augment the resulted data points into the training set by adding these normalized perturbations to the original training data. Algorithm 1 provides a formal description of the process of the proposed data augmentation method.

Figure 1 intuitively demonstrates the effect of *Bamboo* data augmentation. During the training process, the decision boundary will be formed to surround all the training data points of a certain class.

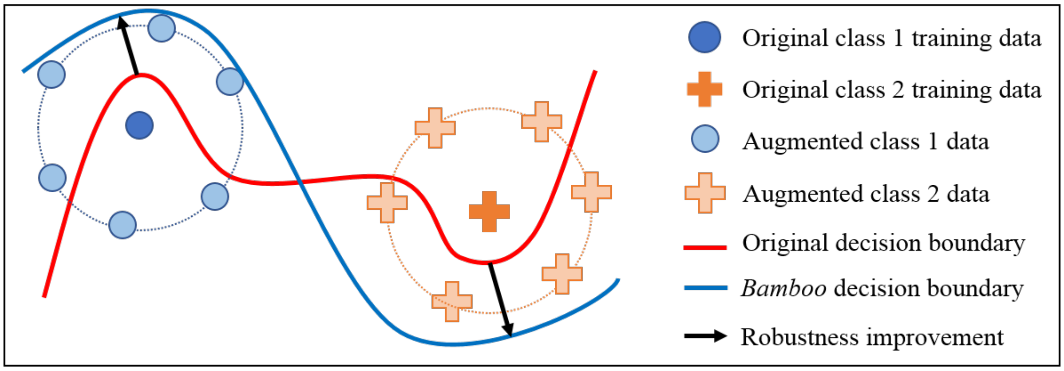
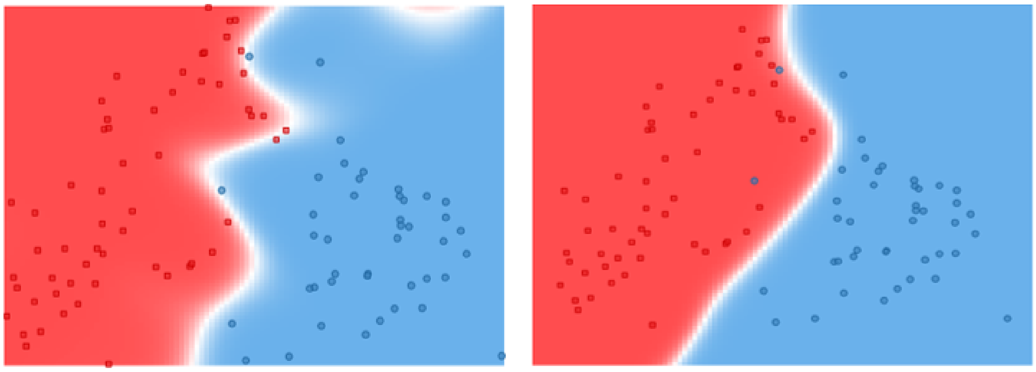

Original class 1 training data
Original class 2 training data
Augmented class 1 data
Augmented class 2 data
Original decision boundary
Bamboo decision boundary
Robustness improvement

Figure 1: Intuition of *Bamboo*'s effect on the DNN decision boundary



(a) Without data augmentation

(b) *Bamboo* data augmentation

Figure 2: Visualization of *Bamboo*'s effect on the DNN decision boundary

Since the decision boundary of the DNN model tends to have small curvature around training data points (Fawzi et al. (2016)), including the augmented data on the ball naturally pushes the decision boundary further away from the original training data points, therefore increases the robustness of the learned model. In such sense, if the DNN model can perfectly fit to the augmented training set, increasing the ball radius will increase the margin between the decision boundary and the original training data, and more points sampled on each ball will make it less likely for the margin to get smaller than r.

Figure 2 shows the effect of *Bamboo* with a simple classification problem. Here we classify 100 data points sampled from the MNIST class of the digit "3" from another 100 data points in the class of the digit "7" using a multi-layer perceptron with one hidden layer. The dimension of all data points are reduced to 2-D using PCA for visualization. Figure 2a shows the decision boundary without data augmentation, where the decision boundary is more curvy and is overfitting to the training data. In Figure 2b, the decision boundary after applying our data augmentation becomes smoother and is further away from original training points, implying a more robust model with the training set augmented with our proposed *Bamboo* method.

4 Experiment

4.1 Experiment setup

To analyze the performance of our proposed method, we test it with *Cleverhans* (Nicolas Papernot (2017)), a python library based on Tensorflow that provides reliable implementation for most of the previously proposed adversarial attack algorithms. As mentioned in Section 2.1, for evaluating the effect of parameter r and N on the performance of our model, we use the average strength of successful CW attack (Carlini & Wagner (2017)) as the metric of robustness. When comparing with previous work, we use both CW attack strength (marked as *CW rob* in Table 1) and the testing accuracy under FGSM attack (Szegedy et al. (2013)) with $\epsilon = 0.1, 0.3, 0.5$ respectively (marked as *FGSM1*, *FGSM3* and *FGSM5* in Table 1). The accuracy under 50 iterations of PGD attack (Madry et al. (2018)) with $\epsilon = 0.3$ is also evaluated here (marked as *PGD3* in Table 1). Moreover, to show the robustness of the trained DNN model to unknown attack, we test the accuracy under Gaussian noise with variance $\sigma = 0.5$ (marked as *GAU5* in Table 1), which demonstrates the robustness against attacks from all directions.

For parameter tuning, we train and test the DNN model on MNIST data set (LeCun et al. (1998)).Both MNIST and CIFAR-10 data set (Krizhevsky & Hinton (2009)) are used for comparing with previous work. The MNIST test adopts the network structure provided in the tutorial of Cleverhans, which consists of three convolutional layers with 64 8×8 kernels, 128 6×6 kernels and 128 5×5 kernels respectively followed by a linear layer with 10 output neurons. For the CIFAR experiment, we choose VGG-16 (Simonyan & Zisserman (2014)) with 10 output neurons as the DNN model. The selection of models is made to demonstrate the scalability of these defending methods. These DNN models, without applying any further training trick, obtain the accuracy of 98.18% on original MNIST testing set after 10 epochs of training. The accuracy on CIFAR-10 testing set would be 83.95% after 100 epochs of training. ImageNet (Deng et al. (2009)) is also used to compare between *Bamboo* and *Mixup* (Zhang et al. (2017)), where we train a ResNet-18 (He et al. (2016)) network for 90 epochs with each method.

In order to analyze the effect of the defending methods on the decision boundaries of DNNs around the testing data points, He et al. (2018) propose a linear search method to find the distance to the nearest boundary in different directions, where they gradually perturb the input data point along random orthogonal directions. When the prediction of the perturbed input becomes different to that of the original input, the perturbation distance is used as an estimate of the decision boundary distance. In our experiments, we follow the setting used in He et al. (2018)'s work, where we use 784 random orthogonal directions for testing MNIST and 1000 random orthogonal directions for testing CIFAR-10. For each testing data point, we find the top 20 directions with the smallest decision boundary distance for each training method, showing how the decision boundary change with different defending methods. We also compute the average of the top 20 smallest distance across all the testing data points, implying the overall effectiveness of different methods on increasing the robustness.

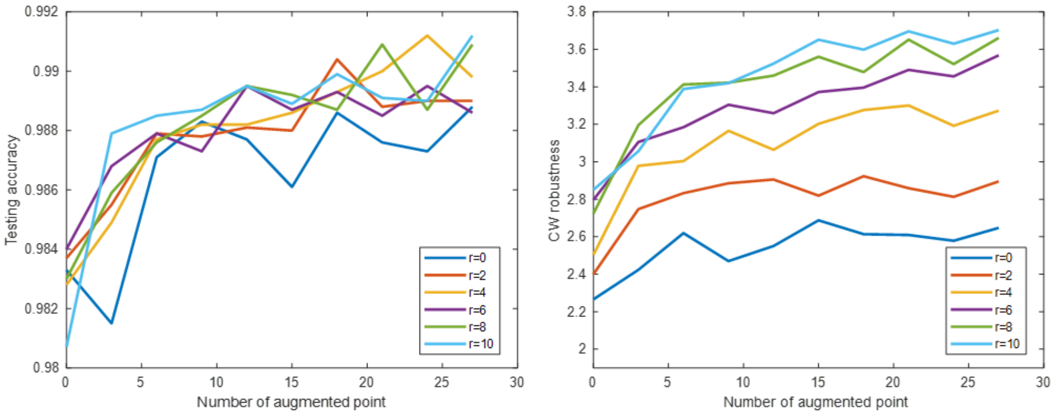
To show the effectiveness of our method, we compare it with FGSM adversarial training (Goodfellow et al. (2014)) with $\epsilon = 0.1, 0.3, 0.5$, the state-of-the-art optimization based defending method *distributional robust optimization (DIST)* (Sinha et al. (2017)) and the adversarial training with PGD attack (Madry et al. (2018)). Newly proposed data augmentation method *Mixup* (Zhang et al. (2017)) is also used for comparison. For the implementation of these algorithms, we adopt the original implementation of adversarial training in Cleverhans, and the open-sourced Tensorflow implementations that replicate the functionality of the distributional robust optimization method[1] and the Mixup method[2]. The hyper-parameters of these algorithms are carefully selected to produce the best performance in our experiments.

4.2 Parameter tuning

As mentioned in Section 3.2, the *Bamboo* augmentation has two hyper-parameters: the ball radius r and the ratio of the augmented data N. We first analyze how the testing accuracy and model robustness change when tuning these parameters. Figure 3a shows the influence of r and N to the testing accuracy. When we fix the radius r, the testing accuracy increases as the number of augmented points grows up. Adjusting the radius, however, has little impact on the testing accuracy. Figure 3b presents the relationship between the number of augmented points and CW robustness under different ball radius. When r is fixed, the robustness improves as data augmentation ratio N increases. The effectiveness of further increasing N becomes less significant as N gets larger. Under the same data amount, increasing the radius r can also enhance the robustness, while the effectiveness of increasing r saturates as r gets larger. According to these observations, in the following experiments, we manually tuned r and N in each experiment setting for the best tradeoff between the robustness and the training cost.

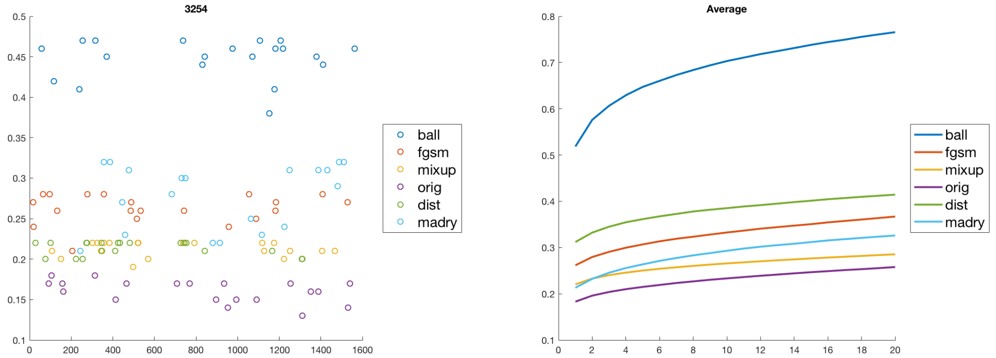
[1] Distributional: `https://github.com/ricvolpi/certified-distributional-robustness`

[2] Mixup: `https://github.com/tensorpack/tensorpack/tree/master/examples/ResNet`



(a) Testing accuracy under different parameters (b) CW robustness under different parameters

Figure 3: Performance result on MNIST dataset



(a) Top 20 smallest distance on MNIST testing data point No.3254

(b) Average of the top 20 smallest distance over all MNIST testing data points

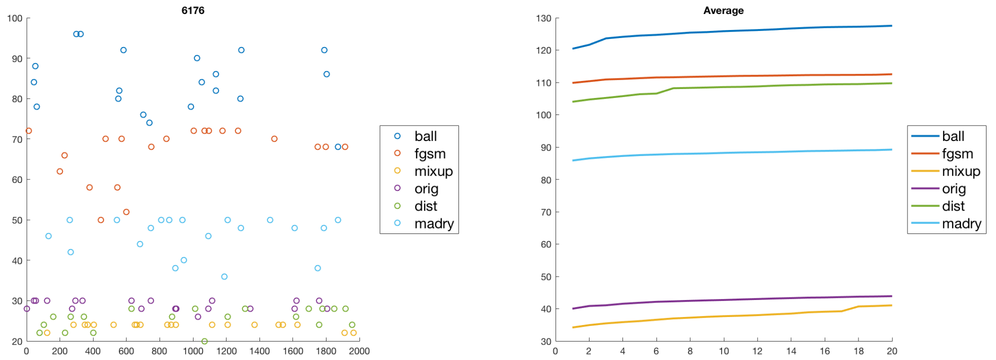
Figure 4: Decision boundary comparison on MNIST dataset

4.3 Boundary visualization

Figure 4 and Figure 5 shows the top 20 smallest decision boundary on random orthogonal directions for MNIST and CIFAR-10 testing points respectively. These results provides a visualization of the effect of different training methods on the decision boundary. From Figure 4a and 5a we can see that adversarial training methods like *FGSM* (Goodfellow et al. (2014)) and *Madry* (Madry et al. (2018)) can improve the decision boundary distance on original vulnerable directions, while may cause other directions to be more vulnerable after training. Comparing to these previous adversarial training methods, optimization based methods and data augmentation methods, our *Bamboo* data augmentation can provide largest gain on robustness for the most vulnerable directions, without introducing new vulnerable directions. The average results over the whole testing set shown in Figure 4b and 5b also proof that *Bamboo* can further improve the overall robustness of the trained model comparing to previous methods.

4.4 Performance comparison

Table 1 summarizes the performance of the DNN model trained with *Bamboo* comparing to other methods. These results are consisted to our prior observations on the advantages and shortcomings of previous works. The adversarial training methods can improve the robustness against the attacking methods they are trained on, especially the one with the same strength as used in training. However, it cannot guarantee the robustness against other kinds of attacks. Distributional robust op-



(a) Top 20 smallest distance on CIFAR-10 testing data point No.6176

(b) Average of the top 20 smallest distance over all CIFAR-10 testing data points

Figure 5: Decision boundary comparison on CIFAR-10 dataset

Table 1: **Performance comparison**: bold type marks the best performance, and italics type marks the second from the best performance

MNIST	Original	FGSM	DIST	PGD	Mixup	**Ours**
				$\epsilon = 0.3$,	$\alpha = 0.12$,	$r = 8$,
		$\epsilon = 0.5$	$c = 0.01$	50 iterations	10×data	10×data
CW rob	2.442	2.390	2.5010	2.343	*2.803*	**3.554**
Test acc	0.9818	0.9817	0.9873	0.9869	**0.9904**	**0.9904**
FGSM1 acc	0.5382	0.6375	*0.8542*	0.7511	0.8323	**0.9292**
FGSM3 acc	0.2606	**0.8963**	0.1169	*0.5840*	0.2623	0.5558
FGSM5 acc	0.1423	**0.9390**	0.0244	0.1340	0.1344	*0.2878*
PGD 3 acc	0.0126	0.0258	0.0065	**0.2534**	0.0180	*0.1281*
GAU 5 acc	*0.6358*	0.6316	0.5735	0.5886	0.5813	**0.9556**
CIFAR-10	Original	FGSM	DIST	PGD	Mixup	**Ours**
				$\epsilon = 0.3$,	$\alpha = 0.12$,	$r = 10$,
		$\epsilon = 0.5$	$c = 0.01$	50 iterations	16×data	16×data
CW rob	38.010	38.210	*38.503*	38.108	37.648	**38.746**
Test acc	*0.8395*	0.7995	0.7935	0.7791	**0.8521**	0.8249
FGSM1 acc	0.4922	0.4927	0.3825	0.4588	**0.7483**	*0.6853*
FGSM3 acc	0.4463	0.6517	0.2241	0.3848	**0.7287**	*0.6806*
FGSM5 acc	0.4093	**0.7572**	0.1998	0.3405	*0.7192*	0.6721
PGD 3 acc	0.2987	0.2233	0.1871	**0.5291**	*0.5018*	0.4111
GAU 5 acc	0.3701	*0.6356*	0.6169	0.5390	0.3371	**0.6961**

timization Sinha et al. (2017) can improve the CW robustness of the DNN model and demonstrate a big improvement against adversarial attacks with small strength (e.g. against FGSM1 on MNIST in the table), but its performance drops dramatically when facing an attack with a larger strength.

Table 2: **Performance comparison on ImageNet**: Tested on ResNet-18 (He et al. (2016)) model after 90 epochs training. Comparison only done between *Bamboo* and Mixup (Zhang et al. (2017)) against FGSM, due to the lack of support of the effectiveness and the lack of open-sourced implementations of other defending and attacking methods on ImageNet

	Original	Mixup	**Ours**
Top-1 acc	57.336	58.213	**60.520**
Top-5 acc	80.647	81.452	**83.216**
Top-1 FGSM	11.342	12.947	**14.062**
Top-5 FGSM	22.860	26.400	**26.562**

We also note that the overall performance of this method on CIFAR-10 dataset is not as good as that on MNIST, possibly due to the scalability issue of the *min-max* optimization as elaborated in Equation (6). A large-scale CNN and larger input space for the CIFAR-10 experiment may be too complicated to efficiently find an optimal solution. Although not specially designed against adversarial attack, the performance of Mixup Zhang et al. (2017) is promising on robustness gain and the accuracy against adversarial attack with small strength. However, the overall robustness achieved by Mixup, indicated by the CW robustness, is not as good as what is achieved by *Bamboo*. The ImageNet experiment results showed in Table 2 show the same trend as well.

Comparing to previous methods, *Bamboo* achieves the highest robustness under CW attack on both MNIST and CIFAR-10 experiments, and the lowest accuracy drop when facing Gaussian noise. Comparing to adversarial training methods with FGSM and PGD that only work best against the attacks they are trained on, *Bamboo* demonstrates a higher robustness against a wide range of attacking methods. Comparing to Distributional robust optimization whose robustness drops quickly as the strength of adversarial attacks goes up, the performance of our method is less sensitive to the change of the attacking strength. Therefore *Bamboo* can also be effectively applied against large-strength attacks. Also, the overall performance of *Bamboo* is better than Mixup with the same amount of data augmented, implying that *Bamboo* is more data efficient in improving DNN robustness against adversarial attack. All these observations lead to the conclusion that our proposed *Bamboo* method can effectively improve the overall robustness of DNN models, no matter which kind of attack is applied or which direction of noise is added.

5 Conclusion and future work

In this work we propose *Bamboo*, the first data augmentation method that is specially designed for improving the overall robustness of DNNs. Without making any assumption on the distribution of adversarial examples, *Bamboo* is able to improve the DNN robustness against attacks from all directions. Previous analysis and experiment results have proven that by augmenting the training set with data points uniformly sampled on a r-radius ball around original training data, *Bamboo* is able to effectively improve the robustness of DNN models against different kinds of attacks comparing to previous adversarial training or robust optimization methods, and can achieve stable performance on large DNN models or facing strong adversarial attacks. With the same amount of augmented data, *Bamboo* is able to achieve better performance against adversarial attacks comparing to other data augmentation methods.

We have shown that the resulted network robustness improves as we increase the radius of the ball or the number of augmented data points. In future work we will discuss the theoretical relationship between the resulted DNN robustness and the parameters in our method, and how will the change in the scale of the classification problem affect such relationship. We will also propose new training tricks better suited for training with augmented dataset. As we explore these theoretical relationships and training tricks in the future, we will be able to apply our method more effectively on any new DNN models to improve their robustness against any kinds of adversarial attacks.

REFERENCES

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *Security and Privacy (SP), 2017 IEEE Symposium on*, pp. 39–57. IEEE, 2017.

Olivier Chapelle, Jason Weston, Léon Bottou, and Vladimir Vapnik. Vicinal risk minimization. In *Advances in neural information processing systems*, pp. 416–422, 2001.

Dong Chen, Gang Hua, Fang Wen, and Jian Sun. Supervised transformer network for efficient face detection. In *European Conference on Computer Vision*, pp. 122–138. Springer, 2016.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pp. 248–255. Ieee, 2009.

Alhussein Fawzi, Seyed-Mohsen Moosavi-Dezfooli, and Pascal Frossard. Robustness of classifiers: from adversarial to random noise. In *Advances in Neural Information Processing Systems*, pp. 1632–1640, 2016.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Warren He, Bo Li, and Dawn Song. Decision boundary analysis of adversarial examples. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=BkpiPMbA-`.

Matthias Hein and Maksym Andriushchenko. Formal guarantees on the robustness of a classifier against adversarial manipulation. In *Advances in Neural Information Processing Systems*, pp. 2263–2273, 2017.

Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. URL `https://openreview.net/forum?id=rJzIBfZAb`.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Seyed Mohsen Moosavi Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

Ian Goodfellow Reuben Feinman Fartash Faghri Alexander Matyasko Karen Hambardzumyan Yi-Lin Juang Alexey Kurakin Ryan Sheatsley Abhibhav Garg Yen-Chen Lin Nicolas Papernot, Nicholas Carlini. cleverhans v2.0.0: an adversarial machine learning library. *arXiv preprint arXiv:1610.00768*, 2017.

Nicolas Papernot, Patrick D. McDaniel, Ian J. Goodfellow, Somesh Jha, Z. Berkay Celik, and Ananthram Swami. Practical black-box attacks against deep learning systems using adversarial examples. *CoRR*, abs/1602.02697, 2016. URL `http://arxiv.org/abs/1602.02697`.

Jonathan Peck, Joris Roels, Bart Goossens, and Yvan Saeys. Lower bounds on the robustness to adversarial perturbations. In *Advances in Neural Information Processing Systems*, pp. 804–813, 2017.

Patrice Y Simard, Yann A LeCun, John S Denker, and Bernard Victorri. Transformation invariance in pattern recognitiontangent distance and tangent propagation. In *Neural networks: tricks of the trade*, pp. 239–274. Springer, 1998.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Aman Sinha, Hongseok Namkoong, and John Duchi. Certifiable distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*, 2017.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *CoRR*, abs/1312.6199, 2013. URL `http://arxiv.org/abs/1312.6199`.

Ziang Yan, Yiwen Guo, and Changshui Zhang. Deepdefense: Training deep neural networks with improved robustness. *arXiv preprint arXiv:1803.00404*, 2018.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.