IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

*Digital Object Identifier 10.1109/ACCESS.2017.DOI*

# Besting the Black-Box: Barrier Zones for Adversarial Example Defense

**KALEEL MAHMOOD[1], PHUONG HA NGUYEN[2], LAM M. NGUYEN[3], THANH NGUYEN[4] , MARTEN VAN DIJK[5] .**
[1]Department of Electrical and Computer Engineering, University of Connecticut, CT, USA (e-mail: kaleel.mahmood@uconn.edu)
[2]eBay Inc., CA, USA (e-mail: phuongha.ntu@gmail.com)
[3]IBM Research, Thomas J. Watson Research Center, Yorktown Heights, NY, USA (e-mail: LamNguyen.MLTD@ibm.com)
[4]Amazon Inc., WA, USA (e-mail: thanhng.cs@gmail.com)
[5]CWI Amsterdam, The Netherlands (e-mail: Marten.van.Dijk@cwi.nl)

Corresponding author: Kaleel Mahmood (e-mail: kaleel.mahmood@uconn.edu).

**ABSTRACT** Adversarial machine learning defenses have primarily been focused on mitigating static, white-box attacks. However, it remains an open question whether such defenses are robust under an adaptive black-box adversary. In this paper, we specifically focus on the black-box threat model and make the following contributions: First we develop an enhanced adaptive black-box attack which is experimentally shown to be $\geq 30\%$ more effective than the original adaptive black-box attack proposed by Papernot et al. For our second contribution, we test 10 recent defenses using our new attack and propose our own black-box defense (barrier zones). We show that our defense based on barrier zones offers significant improvements in security over state-of-the-art defenses. This improvement includes greater than $85\%$ robust accuracy against black-box boundary attacks, transfer attacks and our new adaptive black-box attack, for the datasets we study. For completeness, we verify our claims through extensive experimentation with 10 other defenses using three adversarial models (14 different black-box attacks) on two datasets (CIFAR-10 and Fashion-MNIST).

**INDEX TERMS** Adversarial machine learning, adversarial examples, adversarial defense, black-box attack, security, deep learning.

## I. INTRODUCTION

There are many applications based on Convolution Neural Networks (CNNs) such as image classification [1], [2], object detection [3], [4], semantic segmentation [5] and visual concept discovery [6]. However, it is well-known that CNNs are highly susceptible to small perturbations $\eta$ which are added to *benign* input images $x$. As shown in [7], [8], by adding visually imperceptible perturbations to the original image, adversarial examples $x'$ can be created, i.e., $x' = x + \eta$. These adversarial examples are misclassified by the CNN with high confidence. Hence, making CNNs secure against this type of attack is a significantly important task.

In general, adversarial machine learning attacks can be categorized as either white-box or black-box. This categorization depends on how much information about the classifier is necessary to run the attack. The majority of the literature has focused on white-box attacks [9]–[11] where the classifier/defense parameters are known. Likewise, the majority of defenses have been designed with the goal of thwarting white-box attacks [12]–[24]. In this paper, we focus on black-box attacks, where the classifier parameters are hidden or assumed to be secret. This type of adversary

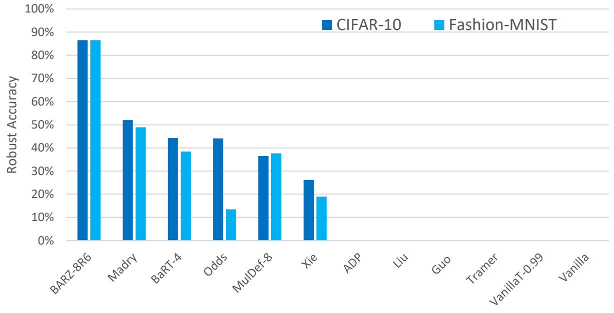


FIGURE 1: The robust accuracy (1-$\alpha$) of each of the 11 defenses analyzed in this paper. For a given defense the robust accuracy is computed as the minimum robust accuracy achieved over all 14 types of black-box attacks. Notice that if no bar is present, then this means $0\%$ robust accuracy.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3138966, IEEE Access

represents a more practical threat model than the white-box attacker [25]. This is in part due to the fact the adversary cannot access the classifier parameters, but is still able to successfully create adversarial examples [25], [26]. Despite not having the defense parameters, the black-box adversary may still query the defense, be able to access $\mathcal{X}$ (the training dataset for the defense), or build a synthetic model to assist them in creating adversarial examples. By analyzing defenses through a black-box adversarial lens, we help complete the security picture by offering both new attack and defense perspectives to the community. Specifically we make the following contributions:

1) **Mixed Black-Box Attack**: We develop an enhanced version of Papernot's black-box attack [26] by expanding the amount of data available to the attacker and changing the final attack generation method $\phi$. These changes significantly improves the attack success rate, i.e. $> 30\%$ improvement on CIFAR-10 and Fashion-MNIST.
2) **Barrier Zone Defense**: We develop a novel defense based on barrier zones – coined BARZ. We show barrier zone based defenses can outperform all 10 other recent defenses studied in this paper. These defenses includes Madry's Adversarial Training [27], Barrage of Random Transforms [22] and Ensemble Diversity [24] just to name a few. A synopsis of our results is displayed in Figure 1 where we show the minimum robust accuracy of each defense under all 14 types of black-box attacks.
3) **The $\delta$ Metric** (minor contribution): In adversarial machine learning, every defense comes with two distinct values to consider. These values are the cost of the defense (drop in clean accuracy) and the robustness/security (performance on adversarial data). We propose an intuitive way to help gauge this trade-off between robustness and cost in the form of the $\delta$ metric.

### A. COMPARING DEFENSES

Figure 1 shows how the robust accuracy of the BARZ defense (defined as $1-\alpha$, where $\alpha$ is the attack success rate of the best out of 14 types of black-box attacks) compares to 10 other recent defenses from literature. The literature defines the attack success rate $\alpha$ as the fraction of adversarial examples that are misclassified by the defense. Here it is also important to precisely define the term *adversarial example*. In short, adversarial examples are clean images that are correctly identified by the classifier in their untampered form, and to which adversarial noise has further been added by the attacker.

For this reason, using only the attack success rate $\alpha$ does not give the complete picture (i.e. only $\alpha$ is shown in Figure 1). The attack success rate $\alpha$ only corresponds to the fraction of original images which the defense classifier can correctly label. In essence, for any given defense $d$, $\alpha$ depends on the clean accuracy of the defense $p_d$ and not the state-of-the-art or best achievable clean accuracy $p$. Here $p$ specifically refers to the accuracy measured on the clean images, without any defense i.e., the clean accuracy. When a defense is present, we denote the corresponding clean accuracy of the defense as $p_d$. So, to complete the story of Figure 1, we need to understand to what extent, the defense itself leads to a lowering of the clean accuracy of the vanilla scheme from $p$ down to $p_d$.

TABLE 1: Accuracy for non-malicious and malicious environments.

| | Vanilla | Defense |
|---|---|---|
| non-malicious | $p$ | $p_d$ |
| malicious | $\approx 0$ | $p_d \cdot (1-\alpha)$ |

Comparing defenses along these two separate metrics of (a) robust accuracy $1-\alpha$ (how well the attacker is able to defeat the defense) and (b) clean accuracy $p_d$ of the defense itself (without adversarial presence) leads to fuzziness. It is not clear which metric is considered more important or what combination is 'best'. The first row in Table 1 depicts the non-malicious environment (i.e., no adversary) and shows the accuracy $p$ of the vanilla scheme, which is the best we can achieve to-date, and the accuracy $p_d$ of the defense, which is less than $p$ as explained above. For the malicious environment, the vanilla scheme cannot achieve any accuracy because $\alpha = 0$ (see the black-box boundary attack in Figure 2). This type of attack can always successfully transform a correctly classified image into an adversarial example that is misclassifed by the vanilla scheme. The probability of proper/accurate classification by the defense in the presence of adversaries is equal to $p_d \cdot (1-\alpha)$ in the lower right corner of Table 1, since the defense properly labels a fraction $p_d$ if no adversary is present, and out of these images a fraction $\alpha$ is successfully attacked, if an adversary is present.

To avoid any fuzziness, we combine both metrics $p_d$ and $1-\alpha$ into a single '$\delta$-metric': We define $\delta$ as the drop in accuracy from the clean accuracy $p$ of the vanilla scheme in the non-malicious environment (top left corner) to the accuracy of the defense in the malicious environment $p_d \cdot (1-\alpha)$ (bottom right corner):

$$\delta = p - p_d \cdot (1-\alpha).$$

When we analyze the non-malicious environment we are only interested in the clean accuracy of the defense – because we do not assume any attack. This gives Figure 2 where the y-axis corresponds to the accuracy $p_d$ for the defense in the non-malicious environment and the x-axis corresponds to the accuracy for the defense in the malicious environment – that is, the x-axis represents the drop $\delta$ from clean accuracy of the vanilla scheme in the non-malicious environment to the accuracy of the defense in the malicious environment (the price for resistance against adversarial examples). We notice that the x-axis and y-axis can map in a straightforward way to the clean defense accuracy $p_d$ and robust accuracy $1-\alpha$ themselves, which we could have reported as the x-axis and y-axis in our plots instead. But this would not visually make clear what combination $(p_d, 1-\alpha)$ is the best in a

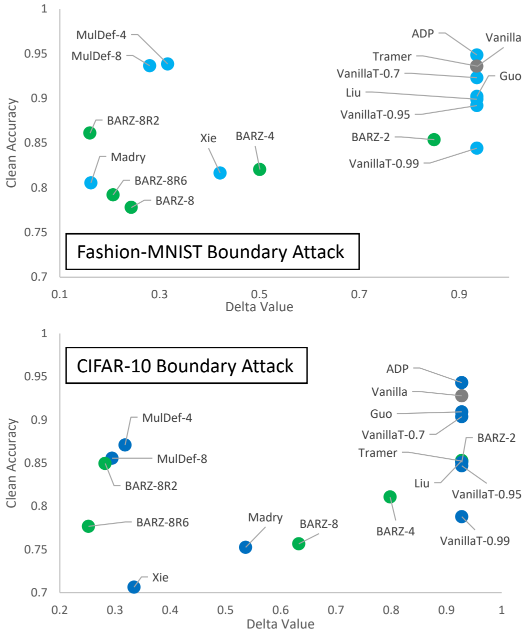


FIGURE 2: The $\delta$ metric vs clean accuracy $p_d$ for the boundary attack. The BARZ results are shown in green and the vanilla result is shown in gray.

malicious environment. We prefer to plot the $\delta$-metric as this corresponds directly to the (drop in) accuracy of the defense classifier in the malicious environment.

In practice, when evaluating a defense, we not only take into consideration the accuracy $p - \delta$ of the defense in the malicious environment but also the accuracy of the defense in the non-malicious environment given by $p_d$ in the top right corner of Table 1. From a pure machine learning perspective, we want a defense which does not affect $p$ 'too much' – in other words the drop $\gamma = p - p_d$ should be small and limited to a couple of percentage points. However, security often does not come for free and in order to minimize $\delta$ we may need to sacrifice much more than a couple of percentage points. This means that we need to study a trade-off between minimizing $\delta$ and an acceptable $p_d$. This paper presents such a study and our defense BARZ is aimed at minimizing $\delta$ despite a possibly significant drop $\gamma$ from $p$ to $p_d = p - \gamma$ in the non-malicious environment. It turns out that this leads to a robust accuracy for BARZ which outperforms those of other defenses as depicted in Figure 1 and Figure 2.

### B. OUTLINE

The rest of the paper is organized as follows: In Section II we discuss black-box adversaries, why we focus on certain attacks and our new mixed black-box attack. In Section III we discuss the defenses we study, the security principles behind them and why we selected these defenses for analysis. In Section IV we introduce the mathematical intuition behind the security principles in the barrier zone defense. We discuss how barrier zone are realized in practice and show empirical proof of them as well, in Section IV. In Section V we explain how to concisely analyze the efficiency of a defense. We give experimental results for all 11 defenses and 14 attacks in Section VI. Lastly we offer concluding remarks in Section VII.

## II. ATTACKS

The general setup in adversarial machine learning for both white-box and black-box attacks is as follows [28]: We assume a trained classifier $f$ with a correctly identified sample $x$ with class label $y$. The goal of the adversary is to modify $x$ by some amount $\eta$ such that $f(x + \eta)$ produces class label $\hat{y}$. In the case of untargeted attacks, the attack is considered successful as long as $\hat{y} \neq y$. In the case of targeted attacks, the attack is only successful if $\hat{y} \neq y$ and $\hat{y} = t$ where $t$ is a target class label specified by the adversary. For both untargeted and targeted attacks, typically the magnitude of $\eta$ is limited [8] so that humans can still visually recognize the image.

The difference between white-box and black-box attacks lies in how $\eta$ is obtained. In white-box attacks, $\eta$ may be computed through backpropagation on the classifier or by formulating the attack as an optimization problem [7], [11], [29] which takes into account the classifier's trained parameters. The white-box adversary has access to the trained parameters which can be used to compute gradients – in essence, the white-box adversary has access to a gradient oracle (which when queried spits out gradient information).

Black-box attacks on the other hand do not have access to the classifier's parameters when generating $\eta$ and must rely on other information. The black-box adversary may have access to the classifier itself which upon querying returns a score vector or the label for which the score is maximized – we call this a black-box oracle. Besides a black-box oracle, the black-box adversary may also have information about the training data that was used to train the classifier.

From a crypto perspective, a white-box adversary is strictly stronger than a black-box adversary and also has access to the black-box oracle. However, we often forget that the classifier parameters known to the white-box adversary can not only be used to compute a gradient oracle but also a black-box oracle. This is because we often think that gradient information leads to more powerful attacks, hence, we may not need to consider black-box attacks. A defense that demonstrates robustness to white-box attacks that only make use of a gradient oracle does not always imply robustness to black-box attacks. Gradient masking makes it possible for a defense to give a false sense of security [10] against a fully-equipped white-box adversary as it only thwarts white-box attacks based on the gradient oracle. This shows that there is a need to also separately test gradient free attacks, such as black-box attacks.

In this paper, we focus on black-box adversaries which utilize adaptive attacks [26]. A natural question is why do we focus on adaptive black-box type attacks? We do so for the following reasons:

1) State-of-the-art white-box attacks on published defenses have been extensively studied in the literature [9]–[11]. The level of attention given to black-box attacks in defense papers is significantly less. By focusing on black-box attacks, we seek to complete the security picture. This full security picture means that the current defenses we analyze have not only white-box attacks (from their own publication), but also adaptive black-box results (as reported in this paper). Future defenses can build upon the security concepts developed in this paper and our experiments, when making their own analyses. This completed security spectrum brings us to our next point.
2) By completing the security picture (with black-box attacks) we allow the readers to compare defense results. This comparison can be done because the same adversarial model, dataset and attack is used for each defense. This is completely different from adaptive white-box attacks which may require different adversarial models and different security assumptions for each attack. For example, in [9] to break a detector defense (The Odds are Odd), a custom objective function must be employed to achieve a high attack success rate in the adaptive white-box attack. Alternatively, creating an adaptive white-box attack on an ensemble model defense (ADP [24]) is much different. The only requirement is to increase the number of iteration used in a simple gradient based white-box attack, to make the attack adaptive and effective. Although both adaptive attacks in our example are white-box, the latter (the adaptive white-box attack on ADP) technically only requires being able to backpropagate on the model. As noted in [30] it is improper to compare the robustness of two defenses under different adversarial models.

### A. BLACK-BOX ATTACK VARIATIONS

**A. Pure black-box attack** [10], [31]–[33]: The adversary is *only* given knowledge of a training data set $\mathcal{X}_0$.

**B. Oracle based black-box attack** [26]: The attacker does not have access to the original training dataset, but may generate a synthetic dataset $S_0$ similar to the training data. The adversary can adaptively generate synthetic data and query the defense $\mathcal{O}$ to obtain class labels for this data. The synthetic dataset $S_0$ is then used to train the synthetic model. It is important to note the adversary does not have access to the entire original training dataset $\mathcal{X}_0$.

In this paper, we propose a new version of this attack which we call the **Mixed Black-Box Attack**. In this attack, the adversary is given the *entire* original training dataset, the ability to generate synthetic data and query access to the defense to label the data. The adversary in our attack also has multiple different adversarial generation methods $\phi$ to choose from to create adversarial examples. In this way, the adversary can train a synthetic model whose behavior mirrors that of the defense more precisely. In short, the attacker adapts the synthetic model to the defense. It is important to note the earlier version of this attack [26] did not allow full access to the training dataset $\mathcal{X}_0$ and the adversarial generation method $\phi$ was fixed to be the Fast Gradient Sign Method (FGSM).

Experimentally, we show that the mixed black-box attack outperforms the original attack proposed by Papernot. Our experiments also show the mixed black-box attack works better on certain types of randomized defenses when compared to both boundary and pure black-box attacks [10], [25], [31]–[34]. The pseudo-code for the mixed black-box attack is given in Algorithm 1 and explained in section II-B.

**Algorithm 1 - Mixed Black-Box Attack**. Oracle $\mathcal{O}$ (i.e., the classifier with defense) is modeled using synthetic model $M$ which is trained using method $T$ for $E$ epochs with starting dataset $X_0 \subseteq \mathcal{X}_0$ and data augmentation parameter $\lambda$. The final adversarial samples are generated from input set $X_{clean}$ using attack method $\phi$ within perturbation $\epsilon$.

```
Input: $\mathcal{O}$, $X_0$, $\phi$, $\lambda$, $E$ and $X_{clean}$
$S_0 \leftarrow \{(x, \mathcal{O}(x)) \; : \; x \in X_0\}$
//Train model based on initial random parameters $\theta$
$M(\theta_0) \leftarrow T(M(\theta), S_0)$
for $e \in \{1, .., E\}$:
    //Augment the dataset with Jacobian technique $J_F$
    $X_e = \{x + \lambda \cdot sgn(J_F(x)) : x \in X_{e-1}\}$
    $S_e \leftarrow \{(x, \mathcal{O}(x)) \; : \; x \in X_e\} \cup S_{e-1}$
    //Train $M$ on the new dataset
    $M(\theta_e) \leftarrow T(M(\theta_{e-1}), S_e)$
//Generate adversarial examples with $M(\theta_E)$ and attack $\phi$
Output: $X_{adv} \leftarrow \{(x, \phi(M(\theta_E), \epsilon; x, y)) \; : \; (x, y) \in X_{clean}\}$
```

**C. Boundary black-box attack** [35]: In this type of attack the adversary has query access to the classifier and only generates a single sample at a time. The main idea of the attack is to try and find the boundaries between the class regions using a binary search methodology and a gradient approximation for the points located on the boundaries.

**D. Score based black-box attacks** In the literature, these attacks are also called Zeroth Order Optimization based black-box attacks [36]. The adversary adaptively queries the defense to approximate the gradient for a given input based on a derivative-free optimization approach. This approximated gradient allows the adversary to directly work with the classifier of the defense. Another attack in this line is called SimBA (Simple Black Box Attack) [37]. Unlike all the previously mentioned attacks, this attack requires the score vector $f(x)$ to mount the attack, instead of merely using the hard label.

The only type of black-box attack we do not consider in our analysis from the ones enumerated above, is the score based black-box attack. Just like white-box attacks are susceptible to gradient masking, score based black-box attacks can be neutralized by a type of masking [30]. This means defenses can appear to be secure to score based black-box attacks, while actually not offering true black-box security. Furthermore, it has been noted that a decision (hard label) based black-box attack represents a more practical adversarial model [25]. Therefore, we slightly focus our scope on the three other black-box variants.

We implement the pure black-box attack and mixed black-box attacks. In both these types of attacks adversarial samples are generated from the synthetic model using six different methods, FGSM [8], BIM [38], MIM [39], PGD [27], C&W [11] and EAD [40]. We also consider boundary black-box attacks. Here we implement the original boundary attack, the Hop Skip Jump Attack (HSJA) [25], as well as the newly proposed Ray Searching Attack (RayS) [34]. In total these attacks represent fourteen different ways to generate black-box adversarial examples.

### B. ATTACK SUCCESS RATE

For classifier $C$ we define $\mathcal{X}(C)$ as the set consisting of image label pairs $(x_i, y_i)$ from the training data set $\mathcal{X}_0$ that are correctly classified by $C$, i.e.,

$$\mathcal{X}(C) = \{(x_i, y_i) \in \mathcal{X}_0 \ : \ C(x_i) = y_i\}.$$

We say $\mathcal{X}(C)$ represents the set of clean images with respect to classifier $C$.

We broaden our description of a classifier $C$ by allowing it to output a 'do not know' symbol $\perp$. This may happen if $C$ computes a score vector $f(x)$ on input $x$ where the scores do not clearly favor any label. Later we will also interpret $\perp$ as the 'adversarial' symbol indicating that it may be an adversarial example.

We define the attack success rate $\alpha$ for classifier $C$ with respect to a particular adversarial sample generation technique $\phi$ as

$$\alpha(C,\phi) = 1 - \frac{1}{|\mathcal{X}(C)|} \sum_{(x_i,y_i)\in\mathcal{X}(C)} \Pr[C(\phi(x_i,y_i)) \in \{y_i, \perp\}].$$

Here, the probabilities are over the coin tosses used in $\phi$ and $C$. The attack success rate reflects when an adversarial example is successful meaning that $C$ will predict a legitimate label, that is $\neq \perp$, which is not equal to the correct class label, that is $\neq y_i$.

We note that $\phi$ is separately trained/modeled/generated using the information available to the black-box adversary. This information may consist of sets $\mathcal{X}_0$ and set $\mathcal{X}(C)$, and based on these sets a self-generated synthetic model $M(\theta)$, where $\theta$ denotes the parameters of the synthetic model. Implicitly, $\phi$ incorporates a perturbation parameter $\epsilon$ indicating into what extent an adversarial example $\phi(x_i, y_i)$ may differ from the original image $x_i$.

The attack success rate estimates the fraction of clean images of $C$ for which successful adversarial examples can be generated. Successful means $C(\phi(x_i, y_i)) \neq y_i$, i.e., the adversarial example $\phi(x_i, y_i)$ is misclassified to an incorrect label even though it is close to the original image $x_i$ (with respect to perturbation parameter $\epsilon$). Here we consider so-called untargeted attacks where the adversary is only interested in misclassification to some other legitimate but wrong label. (An adversarial example for a targeted attack are defined to be successful if the classifier labels it with a target class label specified by the adversary.) In practice we estimate $\alpha(C, \phi)$ by taking a subset $X_{clean} \subseteq \mathcal{X}(C)$ and compute the fraction of adversarial examples $\phi(x, y)$, $(x, y) \in X_{clean}$, that are successful.

The above applies to the mixed-box black attack, see Algorithm 1, as follows. By oracle $\mathcal{O}$ we denote the classifier with defense to which the adversary has access. The attacker starts with some starting data $X_0 \subseteq \mathcal{X}_0$, generally, we assume the worst-case for the defender, i.e., the adversary uses all the training data $X_0 = \mathcal{X}_0$ as a starting point. Data augmentation is used to recursively generate an augmented dataset $S_e$ where queries to oracle $\mathcal{O}$ are used to find labels. Some training method $T$ (based on mathematical optimization for machine learning) learns new parameters $\theta_e$ for model $M$ based on $S_e$ with initial parameters $\theta_{e-1}$. The final synthetic model $M(\theta_E)$ can be attacked by using a white-box attack method $\phi$ (this is possible because the black-box adversary knows parameters $\theta_E$, hence, a gradient oracle for its synthetic model $M(\theta_E)$ is available). At the final step adversarial examples are generated for $X_{clean}$ and we can compute the fraction for which these are successful – and this estimates $\alpha(\mathcal{O}, \phi(M(\theta_E), \epsilon; \cdot))$.

## III. DEFENSES

The field of adversarial defenses is rapidly expanding, with multiple defense papers released almost every month[1]. To examine every proposed defense is beyond the scope of this paper. Instead, we focus our analysis on ten recent, related and/or popular defenses. In this section we describe the related defenses, their common security elements and why we selected them for comparison. The related defenses we consider are Barrage of Random Transforms (BaRT) [22], The Odds are Odd (Odds) [23], Ensemble Diversity (ADP) [24], Madry's Adversarial Training (Madry) [27], Multi-model-based Defense (Mul-Def) [21], Countering Adversarial Images using Input Transformations (Guo) [20], Ensemble Adversarial Training: Attacks and Defenses (Tramer) [14], Mixed Architectures (Liu) [33], Mitigating adversarial effects through randomization (Xie) [18], Thresholding Networks (a basic proof of concept defense developed in this paper) and Barrier Zones (BARZ), the main technique proposed in this paper. In general, adversarial defenses can be divided based on several underlying defense mechanisms. We note this type of division is common in other defense papers as well [41]. While the definitions for categorization we provide

[1] https://nicholas.carlini.com/writing/2019/all-adversarial-example-papers.html

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3138966, IEEE Access

here are by no means absolute, they give us a way to better understand and analyze the field.

1) **Multiple models** - The defense uses multiple classifiers for prediction. The classifier outputs may be combined through averaging (i.e. ADP), randomly picking one classifier from a selection (Mul-Def) or through majority voting (Mixed Architecture).
2) **Image transformations** -The defense applies image transformations before classification. In some cases, the transformation may be randomized (Xie and BaRT) or fixed (Guo).
3) **Adversarial training** - The classifier is trained to correctly recognize adversarial examples with their correct label. Madry, Mul-Def and Tramer all use adversarial training.
4) **Adversarial detection** - The defense outputs a null label if the sample is considered to be adversarially manipulated. Odds employs an adversarial detection mechanism, as does the vanilla thresholding network we consider as a proof of concept defense in this paper.
5) **Randomization** - The defense employs some form of randomization during prediction that is not known a priori to the attacker. BaRT and Xie both apply random image transformations at run time to the input.

### A. BARRAGE OF RANDOM TRANSFORMS (BART)

Barrage of Random Transforms (BaRT) by [22] is a defense that applies a set of image transformations $i_1, ..i_r$ to the input $x$ before classification. There are ten types of image transformations that BaRT employs: JPEG compression, image swirling, noise injection, Fourier transform perturbations, zooming, color space changes, histogram equalization, grayscale transformations and denoising operations. For each input $x$, the number of transformations, the order of the transformations and the parameters in the transformations are randomly selected at run time.

**Why we selected it:** As the defense we propose (BARZ) also uses image transformations, BaRT is a natural candidate to compare to. In building the defense, BaRT trains a single network on multiple image transformations. In contrast, our defense trains multiple networks, each on its own smaller set of image transformations. Comparing these two different ways of building image transformation based defenses is of interest.

### B. THE ODDS ARE ODD (ODDS)

The Odds are Odd was first introduced in [23] as a statistical test for detecting adversarial samples. The concept behind the test is based on a simple observation: clean samples and adversarial samples have different values in the logits layer $l(\cdot)$. Here we define the logits layer as the layer before the soft-max layer. When given an input $x$, the test works by creating multiple copies of the input each with random noise added $\hat{x}_1, ..., \hat{x}_p$. The statistical test uses $l(\hat{x}_1), .., l(\hat{x}_p))$ as input to distinguish between adversarial and clean examples.

**Why we selected it:** In the black-box setting adversarial detection is one possible way to make the defense stronger as the attacker has to produce a wrong class label and avoid the defense marking the input as adversarial ($\perp$). In the defense proposed in this paper (BARZ) we also employ detection by using a threshold voting method with multiple classifiers. As security through detection is precisely what Odds attempt to achieve, it makes sense to compare statistical detection methods to voting based detection defenses such as BARZ.

### C. IMPROVING ADVERSARIAL ROBUSTNESS VIA PROMOTING ENSEMBLE DIVERSITY (ADP)

Using multiple classifier in a defense is a straight-forward concept based on the notion that it is more difficult to break an ensemble of classifiers as opposed to a single one. In [24] they further this notion by specifically training an ensemble of classifiers to avoid the case where the majority of classifiers simultaneously misclassify an adversarial example. In this defense, security is achieved during training in which an adaptive diversity promoting (ADP) regularizer is used. The ADP regularizer pushes the non-maximal predictions of each ensemble classifier to be mutually orthogonal.

**Why we selected it:** ADP uses an ensemble of classifiers without image transformations or adversarial training. BARZ on the other hand, uses multiple classifiers *with* image transformations. If it were possible to achieve black-box robustness in an ensemble without image transformations (e.g. with only special training like in ADP) this would negate the need for special image transformations in a black-box defense. Therefore, testing ADP and comparing it to BARZ has important black-box security implications.

### D. MADRY'S ADVERSARIAL TRAINING (MADRY)

Madry's adversarial training [27] is a widely used defense with clear security objectives. As CNNs misclassify adversarial examples, the authors in [27] proposed generating the adversarial examples and subsequently learning to classify them correctly during training. In general adversarial training can be broken down into two steps. In the first step, for a given clean dataset and classifier, the defender uses a white-box adversarial attack $\phi$ to derive an adversarial dataset. In the second step, the classifier is trained with the adversarial examples and the original clean labels. These two steps are repeated during training multiple times to create a robust adversarial trained classifier.

**Why we selected it:** Madry's adversarial training is one of the most commonly accepted adversarial machine learning defenses due to its intuitive design and robust results. While the security principles that Madry's adversarial training are based on do not directly overlap with BARZ, it nevertheless is a defense standard to compare to.

### E. MULTI-MODEL-BASED DEFENSE (MUL-DEF)

In [21] they proposed a defense against white-box attacks based on multiple networks, each with the *same* architecture.

The authors in [21] developed their defense based on a specialized training technique. They first start with a classifier $C_1$ that has been trained on the clean dataset $\mathcal{X}$. A white-box attack $\phi_{C_1}$ is done on $C_1$ to generate a set of adversarial examples $\mathcal{S}_1$. A new training set is formed from the original dataset and adversarial examples: $\mathcal{X} \cup \mathcal{S}_1$. This new set is used to train the next classifier $C_2$. This process is repeated such that classifier $C_j$ is trained on $\mathcal{X} \cup \mathcal{S}_1 \cup ... \cup \mathcal{S}_{j-1}$. During prediction the final output is randomly selected from classifiers $C_2, .., C_m$ where $m$ is the number of specially trained classifiers in the Mul-Def.

**Why we selected it:** Mul-Def has overlapping security concepts with BARZ. Both use multiple models in the defense and both try to create distinct classifiers (Mul-Def through special training and BARZ through training on transformed data). In the randomized form of BARZ, a random subset of model outputs is used similar to Mul-Def. The main difference between the two defenses is that Mul-Def does not employ any voting among the models and does not implement any adversarial detection. If an ensemble defense could avoid having to implement detection, this would clearly boost the clean accuracy of the defense. This is due to the fact imperfect detection methods mark some clean samples as adversarial (false positives). Due to their similar security concepts, it is logical to compare Mul-Def to BARZ.

### F. COUNTERING ADVERSARIAL IMAGES USING INPUT TRANSFORMATIONS (GUO)

In [20], the designer selects a set of possible image transformations for a single classifier and keeps the selection of the chosen image transformations secret. The main security idea in this defense (Guo) is that the image transformations will distort the adversarial noise enough such that it is no longer causes the classifier to misclassify the adversarial example.

**Why we selected it:** While we do not directly test the original Guo image transformations, the security concepts behind the Guo defense are the same as a single network in BARZ. Essentially, the security principles in the Guo defense (single network and image transformations) are a special case of BARZ when the number of classifiers $m = 1$. Since Guo defense has already been proposed, it would be redundant to propose BARZ, if BARZ-1 (i.e. the Guo defense) already offered substantial security. Therefore, it is necessary to experiment with the Guo defense.

### G. ENSEMBLE ADVERSARIAL TRAINING: ATTACKS AND DEFENSES (TRAMER)

The authors in [14] proposes another type of adversarial training method. In this defense, adversarial examples are generated by attacking multiple networks with multiple different attack methods. After this the designer trains a new network with the generated adversarial examples. The authors in [14] argued that this adversarial training can make the adversarially trained network more robust against (pure) black-box attacks because it is trained with adversarial examples from different sources (i.e., pre-trained networks).

**Why we selected it:** The Tramer defense has natural security concepts parallel to BARZ. Both defenses rely on multiple models. In BARZ these models are used for consensus voting, in the Tramer defense they are indirectly relied on (for generating new adversarial examples). Both defenses are also designed with black-box adversaries in mind. Hence, the Tramer defense is a natural choice to test when considering black-box threat models.

### H. MIXED ARCHITECTURE (LIU)

In [33], the authors studied the transferability between CNNs with different architectures for the ImageNet dataset. They found that adversarial samples do not always transfer between different architectures, i.e. adversarial samples misclassified by $C_1$ are not always misclassified by $C_2$. Based on this study one could propose a defense made up of different CNNS $C_1, .., C_m$ each with a different structure.

**Why we selected it:** While not directly proposed in [33], the question of the viability of a mixed architecture defense arises from the results of [33]. As BARZ uses multiple models, would it make a significant difference in robustness if the architectures of the models are mixed? By testing the mixed architecture defense (Liu) we try and empirically answer this question.

### I. MITIGATING ADVERSARIAL EFFECTS THROUGH RANDOMIZATION (XIE)

In [18] a defense is developed using a single classifier where a random image transformation $i_r$ is applied to the input $x$ at run time. Unlike BaRT or BARZ, this method does not require retraining the classifier on the different image transformations $i_1, ...i_p$.

**Why we selected it:** The Xie defense uses image transformations just like BARZ. Hence this defense presents a unique competing concept: achieve security through randomization without costly retraining. Whether gaining this robustness without retraining is possible under a black-box adversary is why we study the Xie defense in this paper.

### J. THRESHOLDING NETWORK (VANILLAT)

The thresholding network is a simple defense demonstrated in this paper to highlight the challenging nature of creating robust barrier zones. The threshold network is a detection type of defense that uses a vanilla classifier $C$ and threshold $t$. If the highest probability $p$ from classifier $C$ falls below threshold $t$, the sample is marked as adversarial: $\perp$.

**Why we selected it:** When considering barrier zones defenses, the first intuition might be that simply thresholding a vanilla classifier could work. That would mean robustness could be achieved without multiple classifiers or image transformations. We develop the thersholding network defense to empirical demonstrate that a single classifier barrier zone is not sufficient.

## IV. BARRIER ZONE DEFENSE (BARZ)

With so many different kinds of defenses, a natural question is why do we propose another? In short, the answer is because no current defense we analyze performs well against ALL types of black-box attacks and offers a flexible trade-off between security and clean accuracy. For example, adversarially trained networks like Madry perform poorly against pure black-box attacks (less than 65% robust accuracy on CIFAR-10 [27]). Randomized defenses like Xie and Mul-Def work well against boundary attacks but fail against mixed black-box attacks which can adapt to the randomization (we show results for this in section VI). If we want to increase their security, it is not immediately clear how much clean accuracy will be impacted. Likewise, if we want greater clean accuracy, without completely abandoning the defense, it is not obvious how this can be accomplished. In BARZ by adding more networks this trade-off between security and clean accuracy is transparent. BARZ is also one of the only defenses that performs well across all types of black-box attacks (pure, mixed and boundary).

We present full experimental results in section VI to support these claims and give an individual analysis of every defense with respect to black-box attacks in the appendix. Our main focus is to create a defense where the other proposed methods fall short. We strive to create a high fidelity defense (BARZ) that provides flexibility between security and clean accuracy.

### A. SECURITY PRINCIPLES OF BARRIER ZONES

The BARZ defense is based on the concept of barrier zones. Barrier zones are the regions in between classes where if an input falls in this region, it is marked as adversarial. For any new defense the first question is why is it effective, or in this case why do barrier zones provide security? Here we give the mathematical intuition behind this concept.

Suppose we have $m$ classifiers $C_j$ with corresponding attack success rates $\alpha_j = \alpha(C_j, \theta_j)$, where adversarial sample generation technique $\theta_j$ is specific to classifier $C_j$. Let us construct a new classifier $C$ which uses each $C_j$ to predict a label and outputs the majority decision. If more than one label has the same majority vote, then $C$ outputs $\perp$ representing that it does not know how to assign a label. To output a legitimate label, $C$ needs to have a clear majority vote which is not shared by multiple labels.

Consider an adversarial sample generation technique $\phi$ tuned to $C$. Let vote $V_k$ be defined as

$$V_k(x_i, y_i) = |\{1 \le j \le m \ : \ C_j(\phi(x_i, y_i)) = k\}|$$

(assuming deterministic algorithms $C_j$ and $\phi$ for simplicity). Only if $V_{y_i} > V_k$ for all labels $k \neq y_i$, classifier $C$ will output the correct label $y_i$. The adversarial example $\phi(x_i, y_i)$ is successful if a label different from $y_i$ and $\perp$ is output. That is, there exists a label $\hat{y} \notin \{y_i, \perp\}$ such that $V_{\hat{y}} > V_k$ for all legitimate labels $k \neq \hat{y}$.

This shows that the difference

$$A(y_i, k) = V_{y_i} - V_k$$

represents the 'advantage' of choosing $y_i$ over $k$ in classifier $C$. By using notation $A(.,.)$ and translating our characterization of successful adversarial examples, we have attack success rate $\alpha = \alpha(C, \phi)$ equal to

$$\alpha = \frac{\left| \left\{ \begin{array}{c} (x_i, y_i) \in \mathcal{X}(C) : \\ \exists_{\hat{y} \in K \setminus \{y_i, \perp\}} \ \forall_{k \in K \setminus \{\hat{y}\}} \ A(\hat{y}, k) > 0 \end{array} \right\} \right|}{|\mathcal{X}(C)|}, \qquad (1)$$

where $K$ is the set of all legitimate class labels together with $\perp$.

This establishes the conditions for a successful attack on multiple standard classifiers when the output is determined by the majority. We now demonstrate how two security principles in BARZ increase the difficulty of the attack conditions.

**Absolute Consensus Majority Voting:** Instead of using simple majority voting, in BARZ we use absolute consensus majority voting. This means if all classifiers do not agree on the same label, the sample is interpreted as adversarial/suspicious, labeled $\perp$, and the attack fails. We can see that this specifically changes the threshold $> 0$ in (1) to $\geq m$ for a successful attack. Note that while the threshold is now higher, the base conditions for a successful attack, advantages $A(\hat{y}, k)$, did not change in value. Our next security principle deals with the base conditions.

**Input Transformations:** In BARZ each classifier $C_j$ implements its own unique secret input linear transformation $\psi_j$. It is important to note that in this subsection we discuss the secret transformations $\phi_j$ abstractly without designating the specific type of transformation. Theoretically, this allows us to develop the mathematical formulation of the attack success rate of the adversary without assuming the type of transformation. However, for experimentation and defense implementation the image transformation is important and we discuss its choice further in Section IV-B. Once the secret input linear transformation $\psi_j$ is applied, a classifier $C'_j$ is executed:

$$C_j = C'_j \circ \psi_j.$$

The reason for individual transformations is to further increase the difficulty in crafting adversarial example $\phi(x_i, y_i)$. It has already been shown in the literature that vanilla classifiers have high transferability [33]. Therefore, using standard vanilla classifiers without transformations (for all $k$, $\psi_k$ is the identity function), does not significantly improve the security for the following reason: If

$$C'_1(\phi(x_i, y_i)) = \hat{y} \neq y_i,$$

then due to transferability there is a high probability that all standard vanilla classifiers $C'_k$ output the same wrong label $\hat{y}$. This implies that the absolute consensus majority voting with vanilla classifiers yields a high attack success rate $\alpha$. See necessary condition in (1) with absolute consensus majority vote $\geq m$.

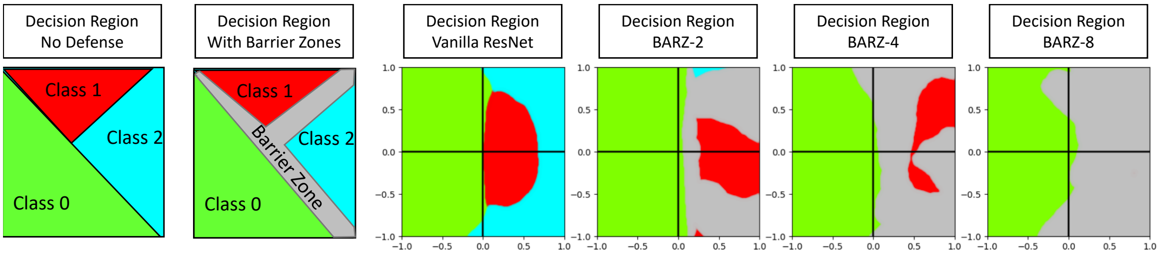


FIGURE 3: Decision regions with and without barrier zones.

We can rewrite $\phi(x_i, y_i)$ as the corresponding clean image and noise: $\phi(x_i, y_i) = x_i + \eta_i$. Under this formulation we can reformulate (by using linearity of $\psi_j$) the base condition $A(\hat{y}, k)$ to

$$\begin{aligned} &|\{1 \leq j \leq m \;:\; C'_j(\phi(x_i, y_i)) = \hat{y}\}| \\ &\quad -|\{1 \leq j \leq m \;:\; C'_j(\phi(x_i, y_i)) = k\}| \\ &= |\{1 \leq j \leq m \;:\; C'_j(\psi_j(x_i) + \psi_j(\eta_i)) = \hat{y}\}| \\ &\quad -|\{1 \leq j \leq m \;:\; C'_j(\psi_j(x_i) + \psi_j(\eta_i)) = k\}| \end{aligned} \tag{2}$$

There are several important takeaways from (2). While the transformation $\psi_j$ changes between classifiers, the noise the adversary crafts $\eta_i$ does not change. In essence for a single sample $x_i$ the adversary must generate noise $\eta_i$ that is invariant to the set of transformations $\psi_1, .., \psi_m$. Specifically the condition for a successful attack is now: $C'_1(\psi_1(x_i) + \psi_1(\eta_i)) = \hat{y}, \ldots, C'_m(\psi_m(x_i) + \psi_m(\eta_i)) = \hat{y}$ for some $\hat{y} \notin \{y_i, \perp\}$. That is, noise $\psi_j(\eta_i)$ must fool classifier $C_j$, for all $j$ simultaneously, while the adversary can only construct a single noise value $\eta_i$.

When we combine (2) with absolute consensus majority voting our final attack success rate for the adversary can be concisely written as:

$$\frac{\left|\left\{ \begin{array}{c} (x_i, y_i) \in \mathcal{X}(C) : \\ \exists_{\hat{y} \in K \setminus \{y_i, \perp\}} \forall_{j=1}^{m} C'_j(\psi_j(x_i) + \psi_j(\eta_i)) = \hat{y} \end{array} \right\}\right|}{|\mathcal{X}(C)|}.$$

In the original multi-classifier attack formulation (1) only a majority of the classifiers had to miss classify the adversarial example $\phi(x_i, y_i)$ to a label $\hat{y}$ such that $A(\hat{y}, k) > 0$ for any $k \neq \hat{y}$. Under the BARZ defense it is clear the new conditions requires ALL classifiers and each transformation to be bypassed.

### B. REALIZING BARRIER ZONES

In practice barrier zones forces the adversary to add noise $\eta$ greater than a certain magnitude in order to overcome the barrier zone. Because an attack fails if the noise becomes visual perceptible to humans, the adversary is limited in terms of the magnitude of $\eta$. In many cases this means the adversary may not be able to overcome the barrier zone and therefore cannot fool the classifier. Barrier zones are shown both in a theoretical diagram and with actual experimental results in Figure 3. The natural question is how can barrier zones be implemented in classifiers? In this subsection we discuss different techniques that can be used to create barrier zones.

**Multiple Classifiers**: Barrier zones can be created through the use of multiple classifiers. A naïve approach to this method would be to simply use CNNs with different architectures. However, we show that merely using different architectures does not yield security. Specifically, we test such a defense in our results by using one VGG16 and one ResNet56 with majority voting (we denote this as the Liu defense). This has also been shown in the literature in [33]. Other examples of architectural defenses not yielding security include ADP and Mul-Def (which we test in this paper). Instead to break transferability between networks we introduce secret image transformations for each classifier. Our defense composed of multiple classifiers (each with their own transformations) is depicted in Figure 4. Each CNN has two simple unique secret image transformations as shown in Figure 4. The first is a fixed linear transformation $c(x) = Ax + b$, where $A$ is a matrix and $b$ is a vector.

After the linear transformation a resizing operation $i$ is applied to the image before it is fed into the CNN. The CNN corresponding to $c$ and $i$ is trained on clean data $\{i(c(x))\}$. Multiple CNNs are used, each with their own resizing operation and $A$ and $b$ components as shown in Figure 4.

From [22] we know adversarial examples are sensitive to image transformations which either distort the value of the pixels in the image or change the original spatial location of the pixels. It is important to note that in this paper we experimentally established that image resizing and linear transformations can reduce transferability. However, there may be other image transformations that can also accomplish this goal.

**Image Transformation Defenses:** A few simple questions arise when dealing with image transformations in security. For example, can only one network with image transformations be used without retraining? We test this concept using the defense by Xie (and we show it performs worse than BARZ under the mixed black-box attack). Can only a single network with image transformations and retraining work? In essence we test a single network, with one set of transformations (Guo) and a single network retrained on multiple random transformations (BaRT). Both of these defenses perform worse than BARZ for the mixed black-box attack.

Another valid question is can only detection of adversarial samples be employed? We test this hypothesis in the following way, we use a vanilla network and a confidence threshold, i.e. any sample below a certain confidence score is marked as adversarial. We also test the Odds defense which employs its own adversarial detection method. In section VI we show that neither thresholding nor the Odds defense are able to outperform BARZ.

It is important to note that it may be possible to further combine other defense techniques such as adversarial training, randomizing some of the image transformations or any number of other techniques. However, the goal of this paper is not to exhaustively test every possible defense combination. The goal is not to test every defense in the literature either. The objective of this work is to provide a defense framework against black-box adversaries that offers clear trade-offs between clean accuracy and security.

### C. BARRIER ZONE GRAPHS

In Figure 3 we show barrier zone graphs for various defenses for a single image from CIFAR-10. These graphs are based on the decision region graphs originally presented in [33]. In our graphs, each point on the 2D grid corresponds to the class label of an image $I'$. Green represents that $I'$ has been classified correctly, while red and blue regions represent incorrect class labels. Gray represents that the null (adversarial) class label has been assigned. The image $I'$ is generated from the original image $I$:

$$I' = I + x \cdot g + y \cdot r. \tag{3}$$

Here $g$ represents the gradient of the loss function with respect to $I$. In (3) $r$ represents a normalized random matrix that is orthogonal to $I$ (note $g$ is also normalized). Variables, $x$ and $y$ represent the magnitude of each matrix which is determined based on the coordinates in the 2D graph.

In essence the graph can be interpreted in the following sense: The origin is classification of the original image without adversarial perturbations or random noise added. As we move along the x-axis in the positive direction, the magnitude of the gradient matrix $x$ increases. Moving positively along only the x-axis is equivalent to the FGSM attack, where the image is modified by adding the gradient of the loss function (with respect to the input). If we move along the y-axis only, the magnitude of the random noise matrix $y$ increases. This is equivalent to adding random noise to the image. Moving along the positive x-axis and any direction in the y-axis means we are adding an adversarial perturbation and a random noise to the original image $I$. The further from the origin, the greater the magnitude of $x$ and $y$ and hence the larger the distortion that is applied to create $I'$.

In the case where a defense uses multiple networks $m$, each network $i$ will have a different gradient matrix $g_i$. To compensate for this, we average the individual gradient matrices together before normalizing to get $g$. It is important to note that while the graphs shown in Figure 3 give experimental proof of the concept of barrier zones, they cannot be used to attack BARZ defenses in practice. When creating the graphs, we have knowledge of the individual gradient matrices $g_i$ for each individual network $i$. With a black-box adversary only the final output of the defense, $\mathcal{O}(x)$ is known. Individual network outputs are not obtainable. Hence it is not possible to precisely estimate the individual gradients $g_i$ to construct a barrier zone graph under a black-box adversarial model to the best of our knowledge.

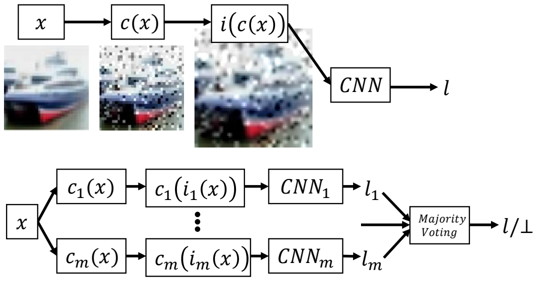


FIGURE 4: Top picture: design of a single network with transformations in BARZ. Bottom picture: the complete BARZ defense comprised of multiple networks. Each network has its own set of transformation. The final output is decided through absolute consensus majority voting. If an absolute consensus is not reached, then the sample is marked as adversarial.

## V. MEASURING DEFENSE PERFORMANCE

In general, when building a defense, there are two primary aspects to consider. The first aspect is security. In the field of adversarial machine learning, security is represented by robust accuracy. When building a defense, the second aspect to consider is the cost. In adversarial machine learning, this cost usually comes in the form of a drop in clean accuracy, $\gamma$. In the ideal case, security would be free, i.e., $\gamma = 0$. In adversarial machine learning, it is well documented that robustness (security) is not free. There is an inherent trade-off between clean accuracy and robustness [42], [43]. Under these circumstances the natural question is, if a cost is always incurred how do we judge a defense?

In this paper, we answer this question by using a metric that measures this trade-off by taking into account both the robustness and clean accuracy. We introduce the $\delta$-metric to properly understand the combined effect of:

1) A drop $\gamma$ in clean accuracy from an original clean accuracy $p$ to clean accuracy

$$p_d = p - \gamma \tag{4}$$

for the defense. Here, clean accuracy $p$ corresponds to a vanilla scheme without defense strategy in a non-malicious environment. Similarly, clean accuracy $p_d$ represents the accuracy for the defense measured in the non-malicious environment without adversaries. (We take "clean" to have the additional meaning of being in a non-malicious environment.)

2) The attacker's success rate $\alpha$ against the defense. If the defense recognizes an adversarial manipulated image as an adversarial example, then it outputs the adversarial label $\perp$ and the attack is not considered successful. When defining $\alpha$, we restrict ourselves to adversarial examples for those images which the defense (in their original non-attacked form) properly classifies by their correct labels. The attacker's success rate is then defined as the fraction of adversarial examples that manipulate these images in such a way that the defense produces labels different from the correct labels and different from the adversarial label $\perp$. For completeness, literature defines the robust accuracy or defense success rate as $1-\alpha$. (We notice that most defenses cannot recognize an adversarial manipulated image as an adversarial example and do not have an adversarial label as possible output.)

Proper classification by the defense in the presence of adversaries is one of the following: An image (possibly after adversarial manipulation) is recognized by its correct label (implying the attack did not work). Or, an adversarial manipulated image is given the adversarial label $\perp$ (if the defense offers this possibility).

The probability of proper/accurate classification by the defense in the presence of adversaries is equal to $(p-\gamma)(1-\alpha)$ (since the defense properly labels a fraction $p-\gamma$ if no adversary is present and out of these images a fraction $\alpha$ is successfully attacked if an adversary is present). In other words $(p-\gamma)(1-\alpha)$ is the accuracy of the defense in the presence of adversaries (malicious environment). Going from a non-malicious environment without defense to a malicious environment with defense gives a drop in accuracy of

$$\delta = p - (p-\gamma)(1-\alpha) = \gamma + (p-\gamma)\alpha. \qquad (5)$$

$\delta$ can be used to measure the effectiveness of different defenses, the smaller the better. If two defenses offer roughly the same $\delta$, then it makes sense to consider their $(\gamma, \alpha)$ pairs and choose the defense that either has the smaller $\alpha$ or the smaller $\gamma$.

From a pure ML perspective, in order for a defense to perform well in a non-malicious environment, we want $\gamma$ very small or, equivalently, $p_d$ close to $p$. From a pure security perspective, in order for a defense to perform well in a malicious environment, we want $\delta$ to be small. Therefore, for properly comparing defenses we focus on tuples $(\delta = \gamma + (p-\gamma)\alpha, p_d = p - \gamma)$, where $\alpha$ corresponds to the best attacker's success rate across the best known attacks from literature. Notice that the vanilla scheme can be considered in a malicious environment as well and this will correspond to some $(\delta_{van}, p_d = p)$. Clearly defenses that result in $\delta \geq \delta_{van}$ do not improve over implementing no defense at all (which is the plain vanilla scheme).

In the ideal case $\delta = 0$ when the attack always fails ($\alpha = 0$) and there is no cost in using the defense ($\gamma = 0$). Due to adversarial attacks, $\alpha > 0$ and, hence, this condition does not occur. Therefore, we look for a defense with the smallest $\delta$, e.g. a defense that has both a low $\alpha$ and low $\gamma$. If two defenses have similar $\delta$ values, we may simply consider the one with the better clean accuracy, which is precisely what we do in this paper. It is important to note the $\delta$ metric is simply one way to understand the trade-off between robustness and clean accuracy. It is by no means the definitive or only way to do so. In this paper, we focus on measuring defenses using the $\delta$ metric due to its concise ability to capture both sets of information, $\alpha$ (security) and $\gamma$ (cost). For those interested in other metrics, we provide all the accuracy measurements separately in graphs and tables in the appendix for all attacks and defenses covered in this paper.

## VI. EXPERIMENTAL RESULTS

In this section we provide experimental results to show the effectiveness of the BARZ defense. We also show the improvement our mixed black-box attack gives. We experiment with two popular datasets, Fashion-MNIST [44] and CIFAR-10 [45]. Unlike other reported results in the literature, for every defense, we construct it using the same network architecture whenever possible. We apply the defense to the same dataset and we run every defense under the same set of attacks. This allows us to provide an unprecedented comparison of adaptive black-box attack results.

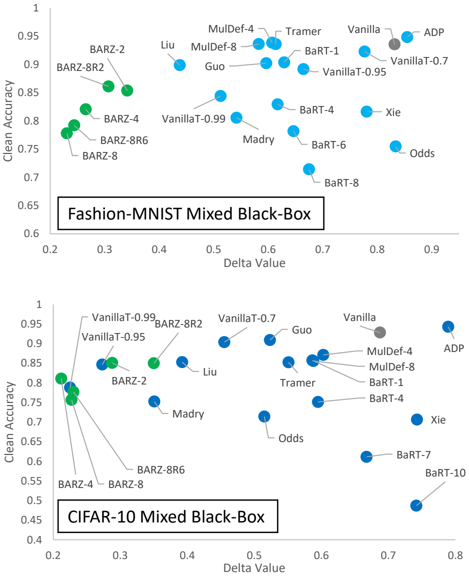


FIGURE 5: The $\delta$ metric vs clean accuracy $p_d = p - \gamma$ for the mixed black-box. The BARZ results are shown in green and the vanilla result is shown in gray.

### A. THE MIXED BLACK-BOX ATTACK

As stated in Section III, our mixed black-box attack is an expansion of the Papernot attack. The original paper [26] experimented with only a single method for generating adversarial samples, the fast gradient sign method (FGSM). We compare

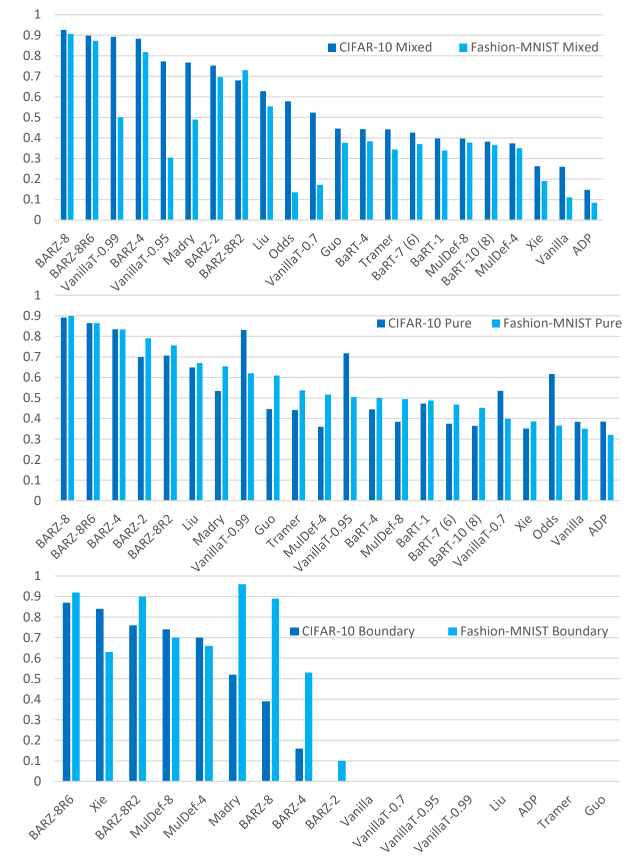


FIGURE 6: Robust accuracies for the untargeted mixed black-box (top), untargeted pure black-box (middle) and untargeted boundary attack (bottom). Note if the defense is listed but no bar is present it means the defense has a 0% robust accuracy against the attack. That is the attack works 100% of the time on the defense.

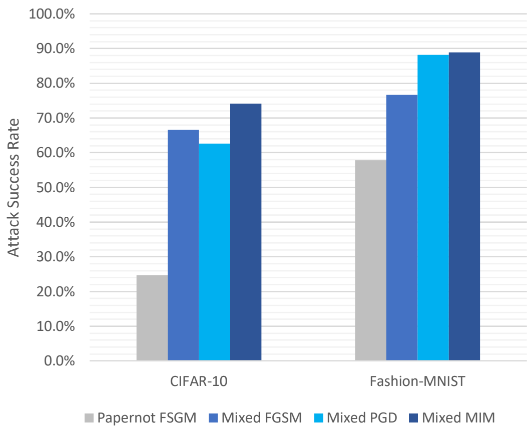


FIGURE 7: The attack success rate of the original Papernot attack and the new mixed black-box attack proposed in this paper. Further comparison and full descriptions to reproduce the experiments are given in the appendix.

results for the Papernot attack and mixed black-box attack in Figure 7 for the $||l||_\infty$ norm with maximum perturbation $\epsilon = 0.05$ for CIFAR-10 and $\epsilon = 0.1$ for Fashion-MNIST. The attack success rate is measured using 1000 samples from the test set. Overall, by providing the adversary with more data, the untargeted attack success rate on a vanilla network can increase by 49.4% for CIFAR-10 and by 31.1% for Fashion-MNIST. More experimental details for these results are given in appendix. Some may argue against the practicality of an adversary that has training data access. However, as a defense designer we want to consider the strongest possible hard label black-box adversary. Hence, the mixed black-box attack is clearly necessary for defense validation.

### B. PURE BLACK-BOX AND BOUNDARY ATTACKS

In addition to the mixed black-box attack, we also consider the pure black-box and boundary attack. Each of these attacks can be further categorized based on how the adversarial samples are generated. For both the pure and mixed black-box attack (proposed in this paper) we use six different adversarial generation methods (FGSM, IFGSM, PGD, MIM, C&W and EAD). For pure black-box attacks we use the same set of generations methods (but the model used in conjunction with the attack is not adaptively trained). For the boundary attacks, we consider HSJA and RayS. In total this represents four types of black-box attacks and 14 different ways adversarial samples can be generated. For CIFAR-10, the maximum perturbation we allow is $\epsilon = 0.05$ and for Fashion-MNIST the maximum perturbation is $\epsilon = 0.1$. For RayS we allow 10,000 queries per sample and for HSJA we use a variable query style attack (which we explain in detail in the appendix). Note in Table 2 some attacks are not applicable to certain defenses. This occurs only for boundary attacks for 2 defenses (BaRT and Odds). This is due to computational complexity issues of non-parallelizable prediction for the run time of the boundary attacks. We fully explain this in the appendix along with precise attack details for all the attacks.

### C. DEFENSES

We experiment with 11 defenses (BARZ, vanilla thresholding, Guo, Liu, ADP, Xie, Madry, Tramer, Mul-Def, BaRT and Odds). In terms of network architecture, we use ResNet56 [46] for the networks in the CIFAR-10 defenses and VGG16 [47] for the networks in the Fashion-MNIST defenses. It is important to note that the results reported here do not always match the literature results identically. This is due to difference in architectures and datasets. For example, the authors of BaRT never published a CIFAR-10 version of their defense, so our BaRT implementation will have different accuracy than what they report for ImageNet. Likewise, Madry's original CIFAR-10 defense was trained using a Wide ResNet where as we use ResNet56V2. We use the same base architecture for every defense (whenever possible) and the same dataset to make our comparisons as valid as possible. Due to the limited space, we cannot

TABLE 2: $\delta$ values and clean accuracies for all the defenses under different attacks. The best $\delta$ for every category is shown in bold. Note robust accuracy for every type of attack (e.g. HSJA, RayS, mixed black-box MIM, pure black-box PGD etc. are given in the appendix.

| | CIFAR-10 | | | | | Fashion-MNIST | | | |
|---|---|---|---|---|---|---|---|---|---|
| | $\delta$ Pure | $\delta$ Mixed | $\delta$ Boundary | Clean Acc | | $\delta$ Pure | $\delta$ Mixed | $\delta$ Boundary | Clean Acc |
| Vanilla | 0.5168 | 0.6875 | 0.9278 | 0.9278 | Vanilla | 0.5960 | 0.8317 | 0.9356 | 0.9356 |
| VanillaT-0.7 | 0.3973 | 0.4551 | 0.9278 | 0.9038 | VanillaT-0.7 | 0.5525 | 0.7768 | 0.9356 | 0.9232 |
| VanillaT-0.95 | 0.2741 | 0.2732 | 0.9278 | 0.8468 | VanillaT-0.95 | 0.4789 | 0.6644 | 0.9356 | 0.8920 |
| VanillaT-0.99 | 0.2534 | 0.2250 | 0.9278 | 0.7879 | VanillaT-0.99 | 0.4105 | 0.5127 | 0.9356 | 0.8442 |
| Liu | 0.3385 | 0.3922 | 0.9278 | 0.8528 | Liu | 0.3333 | 0.4376 | 0.9356 | 0.8990 |
| ADP | 0.4799 | 0.7892 | 0.9278 | 0.9430 | ADP | 0.6036 | 0.8550 | 0.9356 | 0.9486 |
| Xie | 0.6396 | 0.7427 | 0.3344 | 0.7064 | Xie | 0.6205 | 0.7805 | 0.4213 | 0.8164 |
| Madry | 0.5140 | 0.3507 | 0.5366 | 0.7524 | Madry | 0.4080 | 0.5417 | 0.1623 | 0.8055 |
| Tramer | 0.4667 | 0.5510 | 0.9278 | 0.8524 | Tramer | 0.4320 | 0.6136 | 0.9356 | 0.9361 |
| MulDef-4 | 0.5080 | 0.6030 | 0.3182 | 0.8709 | MulDef-4 | 0.4372 | 0.6071 | 0.3161 | 0.9386 |
| MulDef-8 | 0.5009 | 0.5881 | 0.2947 | 0.8556 | MulDef-8 | 0.4720 | 0.5825 | 0.2801 | 0.9365 |
| Guo | 0.4805 | 0.5232 | 0.9278 | 0.9092 | Guo | 0.3852 | 0.5963 | 0.9356 | 0.9023 |
| BARZ-2 | 0.2821 | 0.2881 | 0.9278 | 0.8507 | BARZ-2 | 0.2603 | 0.3414 | 0.8502 | 0.8537 |
| BARZ-4 | **0.2153** | **0.2120** | 0.7981 | 0.8106 | BARZ-4 | 0.2514 | 0.2653 | 0.5008 | 0.8204 |
| BARZ-8 | 0.2258 | 0.2273 | 0.6328 | 0.7565 | BARZ-8 | **0.2363** | **0.2308** | 0.2433 | 0.7779 |
| BARZ-8R2 | 0.2771 | 0.3501 | 0.2822 | 0.8495 | BARZ-8R2 | 0.2846 | 0.3070 | **0.1606** | 0.8611 |
| BARZ-8R6 | 0.2226 | 0.2303 | **0.2521** | 0.7767 | BARZ-8R6 | 0.2505 | 0.2442 | 0.2070 | 0.7920 |
| Odds | 0.4565 | 0.5151 | NA | 0.7141 | Odds | 0.6405 | 0.8337 | NA | 0.7547 |
| BaRT-1 | 0.4238 | 0.5867 | NA | 0.8571 | BaRT-1 | 0.4538 | 0.6292 | NA | 0.9039 |
| BaRT-4 | 0.5213 | 0.5950 | NA | 0.7513 | BaRT-4 | 0.5010 | 0.6171 | NA | 0.8294 |
| BaRT-7 | 0.6368 | 0.6673 | NA | 0.6114 | BaRT-6 | 0.5432 | 0.6464 | NA | 0.7817 |
| BaRT-10 | 0.7491 | 0.7418 | NA | 0.4869 | BaRT-8 | 0.6120 | 0.6748 | NA | 0.7144 |

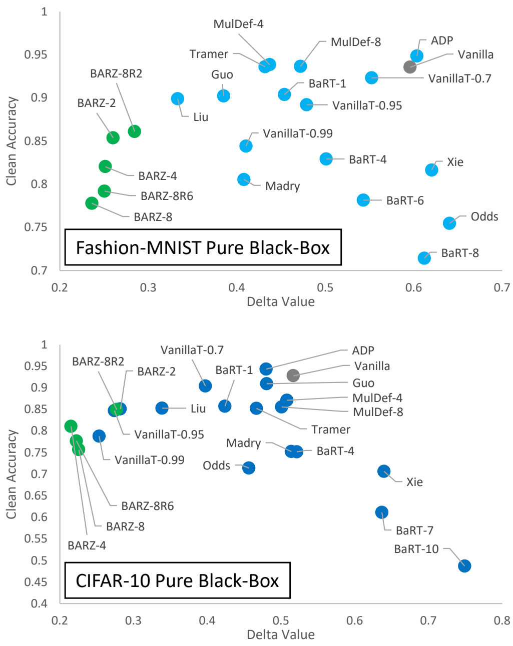


FIGURE 8: The $\delta$ metric vs clean accuracy $p_d = p - \gamma$ for the pure black-box. The BARZ results are shown in green and the vanilla result is shown in gray.

describe the full implementation details of every defense here. We encourage the reader to examine the appendix for further details if interested.

**BARZ and Thresholding Defenses.** In this paper we experiment with BARZ and also a naive defense which we call vanilla thresholding. A common misconception is that by merely thresholding the output of a vanilla classifier (i.e. marking a sample as adversarial if the network is not confident in its prediction) then all black-box attacks can be mitigated. We provide results for the 70%, 95% and 99% thresholding network to show this is simply not the case.

For BARZ, we realize the barrier zones through image transformations. Specifically, each network has an image transformation selected from mappings $c(x) = Ax + b$. We explain how we chose the randomized $A$ and $b$ based on the dataset in the appendix. We can consider an image transformation $c_j(x)$ as an extra randomly fixed layer added to the layers which form the $j$-th CNN. We tested three of these designs: One with 8 networks (BARZ-8) each using a different image resizing operation from 32 to 32, 40, 48, 64, 72, 80, 96, 104. The second with 4 networks (BARZ-4) being the subset of the 8 networks that use image resizing operations from 32 to 32, 48, 72, 96. The third with 2 networks (BARZ-2) being a subset of the 8 networks that use image resizing operations from 32 to 32 and 104.

We also consider a randomized version of BARZ which we denote as BARZ-$x$R$y$. In this version, a subset of $y$ networks (selected from $x$ networks) are used to do the absolute majority vote on a sample. For instances, in BARZ-8R2 every time a sample is submitted, two of the eight networks are randomly selected to classify the sample.

### D. EXPERIMENTAL ANALYSIS

The main results for our paper are given in Table 2 for CIFAR-10 and Fashion-MNIST and the robust accuracy is visually shown in Figure 6. We compute the $\delta$ metric for every defense based on the attack that the defense is weakest to (i.e. has the lowest robust accuracy). For example, if the BARZ-8 defense has a robust accuracy of 60% against RayS (60% of the adversarial samples do not fool the defense) and a robust accuracy of 39% against HSJA, then HSJA is used to compute the BARZ-8 boundary $\delta$ metric. Visually the results for the worst case $\delta$ metric for the pure black-box attack, mixed black-box attack and boundary attack adversaries are given in Figures 5, 8 and  2.

In terms of performance, our proposed defense (BARZ) outperforms every other defense for both CIFAR-10 and Fashion-MNIST. On CIFAR-10, BARZ-4 gives the best tradeoff between security and accuracy for $\delta$ mixed and $\delta$ pure, and BARZ-8 has the best robust accuracy (92.6% for mixed and 92.8% for pure). For boundary attacks BARZ-8R6 gives the best trade-off for CIFAR-10 as well as the best robust accuracy (87%). Likewise, for Fashion-MNIST BARZ-8 has the lowest $\delta$ for the mixed and pure black-box attacks. For Fashion-MNIST BARZ-8 also has the best pure and mixed robust accuracy with 90.6% and 89.9% respectively. For the boundary attacks for Fashion-MNIST, we can see BARZ-8R2 gives the best trade-off but Madry gives slightly better robust accuracy (96% for Madry versus 92% for BARZ-8R2). For those interested in the conventional robust accuracy measurement, we give the overall result in Figure 1. This figure shows the minimum robust accuracy across all black-box attacks for each defense. We can only summarize the main results within this section. In the appendix, we go in depth further comparing results for the 11 defenses.

## VII. CONCLUSION

In this paper, we advance the field of adversarial machine learning by providing a new black-box attack and a novel black-box defense based on barrier zones. Our new attack is experimentally shown to be stronger than the original Papernot attack. It also outperforms boundary and pure black-box attacks on defenses like Xie and Mul-Def. Second, and most importantly, we develop a new barrier zone based defense. Our defense outperforms all 10 other defense methods we tested under pure, mixed and boundary based black-box attacks. When comparing across all black-box attacks and datasets tested in this paper, our best defense configuration gives over 85% robust accuracy for CIFAR-10 and Fashion-MNIST, an improvement of over 30% compared to the next best defense. Overall we develop the first barrier zone defense (BARZ), experimentally shown to be robust against 14 different types of black-box attacks.

## REFERENCES

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," in *NIPS*, F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, Eds., 2012, pp. 1097–1105.

[2] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *ICLR*, 2015.

[3] R. B. Girshick, "Fast R-CNN," in *IEEE-ICCV*, 2015, pp. 1440–1448.

[4] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," in *NIPS*, 2015, pp. 91–99.

[5] E. Shelhamer, J. Long, and T. Darrell, "Fully Convolutional Networks for Semantic Segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 640–651, Apr. 2017.

[6] J. Wang, Z. Zhang, C. Xie, Y. Zhou, V. Premachandran, J. Zhu, L. Xie, and A. L. Yuille, "Visual Concepts and Compositional Voting," *CoRR*, 2017.

[7] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. J. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," *CoRR*, vol. abs/1312.6199, 2013. [Online]. Available: http://arxiv.org/abs/1312.6199

[8] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and Harnessing Adversarial Examples," *CoRR*, vol. abs/1412.6572, 2014. [Online]. Available: http://arxiv.org/abs/1412.6572

[9] F. Tramer, N. Carlini, W. Brendel, and A. Madry, "On adaptive attacks to adversarial example defenses," *arXiv preprint arXiv:2002.08347*, 2020.

[10] A. Athalye, N. Carlini, and D. A. Wagner, "Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples," in *ICML 2018*, 2018, pp. 274–283.

[11] N. Carlini and D. Wagner, "Adversarial examples are not easily detected: Bypassing ten detection methods," in *AISec@CCS*, 2017.

[12] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *2016 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2016, pp. 582–597.

[13] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial Machine Learning at Scale," *CoRR*, vol. abs/1611.01236, 2016. [Online]. Available: http://arxiv.org/abs/1611.01236

[14] F. Tramèr, A. Kurakin, N. Papernot, D. Boneh, and P. D. McDaniel, "Ensemble Adversarial Training: Attacks and Defenses," *CoRR*, vol. abs/1705.07204, 2017. [Online]. Available: http://arxiv.org/abs/1705.07204

[15] X. Cao and N. Z. Gong, "Mitigating Evasion Attacks to Deep Neural Networks via Region-based Classification," *CoRR*, vol. abs/1709.05583, 2017. [Online]. Available: http://arxiv.org/abs/1709.05583

[16] J. H. Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On Detecting Adversarial Perturbations," *CoRR*, vol. abs/1702.04267, 2017. [Online]. Available: http://arxiv.org/abs/1702.04267

[17] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting Adversarial Samples from Artifacts," *CoRR*, vol. abs/1703.00410, 2017.

[18] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating adversarial effects through randomization," *ICLR (Poster)*, 2018.

[19] D. Meng and H. Chen, "Magnet: a two-pronged defense against adversarial examples," in *CSS*. ACM, 2017, pp. 135–147.

[20] C. Guo, M. Rana, M. Cissé, and L. van der Maaten, "Countering Adversarial Images using Input Transformations," *CoRR*, 2017.

[21] S. Srisakaokul, Z. Zhong, Y. Zhang, W. Yang, and T. Xie, "Muldef: Multi-model-based defense against adversarial examples for neural networks," *arXiv preprint arXiv:1809.00065*, 2018.

[22] E. Raff, J. Sylvester, S. Forsyth, and M. McLean, "Barrage of random transforms for adversarially robust defense," in *CVPR*, 2019, pp. 6528–6537.

[23] K. Roth, Y. Kilcher, and T. Hofmann, "The odds are odd: A statistical test for detecting adversarial examples," 2019.

[24] T. Pang, K. Xu, C. Du, N. Chen, and J. Zhu, "Improving Adversarial Robustness via Promoting Ensemble Diversity," in *ICML*, 2019, pp. 4970–4979.

[25] J. Chen, M. I. Jordan, and M. J. Wainwright, "Hopskipjumpattack: A query-efficient decision-based attack," in *IEEE S&P*, 2020, pp. 1277–1294.

[26] N. Papernot, P. D. McDaniel, I. J. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical Black-Box Attacks against Machine Learning," in *ACM AsiaCCS 2017*, 2017, pp. 506–519.

[27] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," *International Conference on Learn- ing Representations (ICLR)*, 2018.

[28] X. Yuan, P. He, Q. Zhu, R. R. Bhat, and X. Li, "Adversarial Examples: Attacks and Defenses for Deep Learning," *CoRR*, vol. abs/1712.07107, 2017. [Online]. Available: http://arxiv.org/abs/1712.07107

[29] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *International Con- ference on Learning Representations (ICLR)*, 2015.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3138966, IEEE Access

[30] N. Carlini, A. Athalye, N. Papernot, W. Brendel, J. Rauber, D. Tsipras, I. Goodfellow, A. Madry, and A. Kurakin, "On evaluating adversarial robustness," *arXiv preprint arXiv:1902.06705*, 2019.

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

[32] N. Papernot, P. D. McDaniel, and I. J. Goodfellow, "Transferability in Machine Learning: from Phenomena to Black-Box Attacks using Adversarial Samples," *CoRR*, 2016.

[33] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into Transferable Adversarial Examples and Black-box Attacks," *ICLR (Poster)*, 2017.

[34] J. Chen and Q. Gu, "Rays: A ray searching method for hard-label adversarial attack," 2020.

[35] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," 2018.

[36] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth Order Optimization Based Black-box Attacks to Deep Neural Networks Without Training Substitute Models," in *AISec*. ACM, 2017.

[37] C. Guo, J. R. Gardner, Y. You, A. G. Wilson, and K. Q. Weinberger, "Simple black-box adversarial attacks," *arXiv preprint arXiv:1905.07121*, 2019.

[38] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *International Conference on Learning Representations (ICLR) Workshop*, 2017.

[39] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, and J. Li, "Boosting adversarial attacks with momentum," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2018, pp. 9185–9193.

[40] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh, "Ead: elastic-net attacks to deep neural networks via adversarial examples," in *Thirty-second AAAI conference on artificial intelligence*, 2018.

[41] K. Mahmood, D. Gurevin, M. van Dijk, and P. H. Nguyen, "Beware the black-box: on the robustness of recent defenses to adversarial examples," 2020.

[42] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, and A. Madry, "Robustness may be at odds with accuracy," in *International Conference on Learning Representations*, 2019. [Online]. Available: https://openreview.net/forum?id=SyxAb30cY7

[43] H. Zhang, Y. Yu, J. Jiao, E. Xing, L. El Ghaoui, and M. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *International Conference on Machine Learning*. PMLR, 2019, pp. 7472–7482.

[44] H. Xiao, K. Rasul, and R. Vollgraf, "Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms," 2017. [Online]. Available: http://arxiv.org/abs/1708.07747

[45] A. Krizhevsky, V. Nair, and G. Hinton, "Cifar-10 (canadian institute for advanced research)." [Online]. Available: http://www.cs.toronto.edu/~kriz/cifar.html

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[47] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

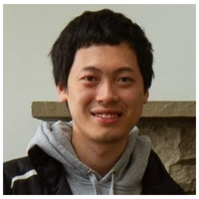
[48] N. Carlini and D. A. Wagner, "MagNet and "Efficient Defenses Against Adversarial Attacks" are Not Robust to Adversarial Examples," *CoRR*, vol. abs/1711.08478, 2017.

[49] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

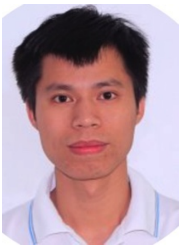
[50] A. Kolesnikov, L. Beyer, X. Zhai, J. Puigcerver, J. Yung, S. Gelly, and N. Houlsby, "Big transfer (bit): General visual representation learning," *Lecture Notes in Computer Science*, p. 491–507, 2020. [Online]. Available: http://dx.doi.org/10.1007/978-3-030-58558-7_29

KALEEL MAHMOOD received his B.S. and M.S. degrees in electrical engineering from the University of Connecticut in 2013 and 2016. He also received an M.S. degree in computer science from the University of Connecticut in 2017. He is currently working on his Ph.D in computer science. His research interests include adversarial machine learning, deep learning, computer vision and security.

PHUONG HA NGUYEN received his Specialist degree in computer science and mathematics from Moscow State University named Lomonosov, Russia in 2008. He also received a Ph.D degree in cryptography from Nangyang Technological University, Singapore in 2013. He is currently working as a researcher at eBay. His research interests include machine learning and cryptography.



LAM M. NGUYEN received his B.S degree in Applied Mathematics and Computer Science from Lomonosov Moscow State University in 2008; M.B.A. degree from McNeese State University in 2013; and Ph.D degree in Industrial and Systems Engineering from Lehigh University in 2018. He currently works as a Research Staff Member at IBM Research, Thomas J. Watson Research Center in the fields of Optimization and Machine Learning. His current research interests include optimization for representation learning, design and analysis of learning algorithms, deep reinforcement learning, and explainable AI.



THANH NGUYEN received his B.S in computer science from Hanoi University of Science and Technology, Vietnam in 2012. He also received a Ph.D degree in Computer Engineering from Iowa State University, USA in 2020. He is currently working as a researcher at Amazon AI. His research interests include machine learning, learning theory, generative modeling, and unsupervised learning.

MARTEN VAN DIJK is group leader of the Computer Security group at CWI, the Netherlands, with over 20 years of experience in both industry (Philips Research and RSA Laboratories) and academia (MIT and full Professor till June 2020 and full Research Professor after June 2020 at UConn). His work has been recognized by the A. Richard Newton Technical Impact Award in Electronic Design Automation, 2015, and has received several best (student) paper awards.

# APPENDIX

## APPENDIX. EXPERIMENTAL DEFENSE RESULTS

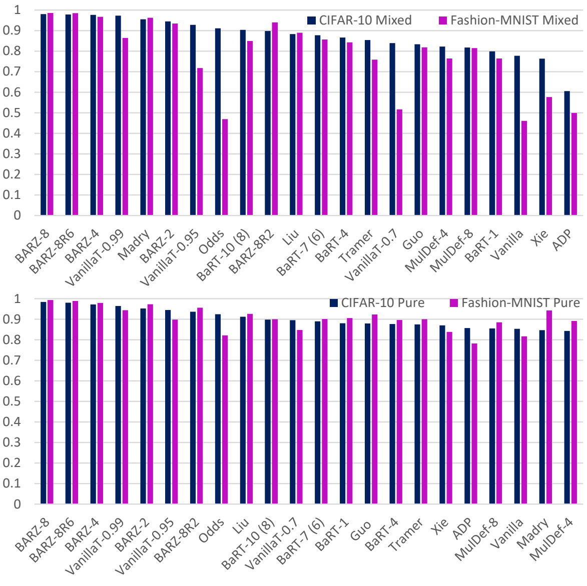


FIGURE 9: Robust accuracies for the targeted mixed black-box (top) and targeted pure black-box attacks (bottom).

In this section, we present our supplementary experimental results for

- the mixed targeted and untargeted black-box attacks,
- the pure targeted and untargeted black-box attacks and
- the boundary attacks – untargeted HopSkipJump [25] and RayS [34].

We run these attacks on **ten different defenses strategies**, Barrage of Random Transforms (BaRT) [22], The Odds are Odd (Odds) [23], Ensemble Diversity (ADP) [24], Madry's Adversarial Training (Madry) [27], Multi-model-based Defense (Mul-Def) [21], Countering Adversarial Images using Input Transformations (Guo) [20], Ensemble Adversarial Training: Attacks and Defenses (Tramer) [14], Mixed Architecture (Liu) [33], Mitigating adversarial effects through randomization (Xie) [18], Thresholding Networks (a basic proof of concept defense developed in this paper) and Barrier Zones (BARZ) with the CIFAR-10 [45] and Fashion-MNIST [44] datasets. The adversarial sample generation is done by running white-box attacks on synthetic models (a model obtained from either a pure or mixed black-box attack). The six white-box attacks used for adversarial sample generation are FGSM [8], BIM [38], MIM [39], PGD [27], C&W [11] and EAD [40]. We also test the defense under boundary black-box attacks (Hop Skip Jump [25] and RayS [34].

We start our section with a discussion on the robustness of defenses under the black-box attacks in this paper.

### A. ROBUSTNESS OF THE DEFENSES

Figures 6 and 9 represent the robust accuracies of the defenses under the different black-box attacks with the Fashion-MNIST and CIFAR-10 datasets. For targeted attacks, Figure 10 shows how the defenses perform in two dimensions, clean accuracy versus delta ($\delta$). We have the following main observations from these figures.

1) Mixed black box attacks are stronger than pure black-box attacks and untargeted attacks are more powerful than targeted ones. Compared to pure black-box attacks, mixed black-box attacks are given more information about the target model (original training data and query access to the target model to label generated synthetic data); for this reason mixed black box attacks should be stronger than pure black box attacks. Because targeted attacks can be considered as an optimization problem with more constraints than untargeted attacks, targeted attacks should take more effort to run than untargeted ones, and are therefore less powerful.
2) Targeted pure black-box attacks seem to not present a strong attack model. This is supported by the fact that the vanilla scheme (which implements no defense at all) already offers very good robustness (i.e., it already has a high defense accuracy against targeted pure black-box attacks). As a result, almost all considered defenses offer good robustness and clean accuracy under this threat model. This explains why the defenses are relatively close together in the plots for targeted pure black-box attacks in Figure 10.
3) As observed and discussed above, mixed black-box attacks are stronger than pure black-box attacks. This explains why a subset of the considered defenses can still significantly improve over the vanilla scheme for targeted mixed black box attacks as shown in Figure 10.
4) For the untargeted boundary attacks, there are many defenses which have 0% robust accuracy. Hence, we do not see any bars for these in Figure 6, for example Vanilla, VanillaT-0.7, etc. have 0% robust accuracy.
5) The most interesting and important observations from Figures 5, 8, 2, 6, 9, and 10 are as follows:
   a) There exists a group of defenses which enjoy a high robustness and clean accuracy, i.e., the defenses lie in the upper left corner with small delta value and high clean accuracy and
   b) BARZ defenses always belong to that group in any of the aforementioned scenarios.

   These observations show that the BARZ family offers a good robustness and clean accuracy compared to other defenses in all scenarios.

We present more detailed attack and defense results in the next sections for Fashion-MNIST and CIFAR-10. Note that *all the detailed results in the next two sections have been visualized in* Figures 5, 8, 2, 6, 9, and 10, where *the most important discussions and observations on these detailed results have been summarized above*.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3138966, IEEE Access

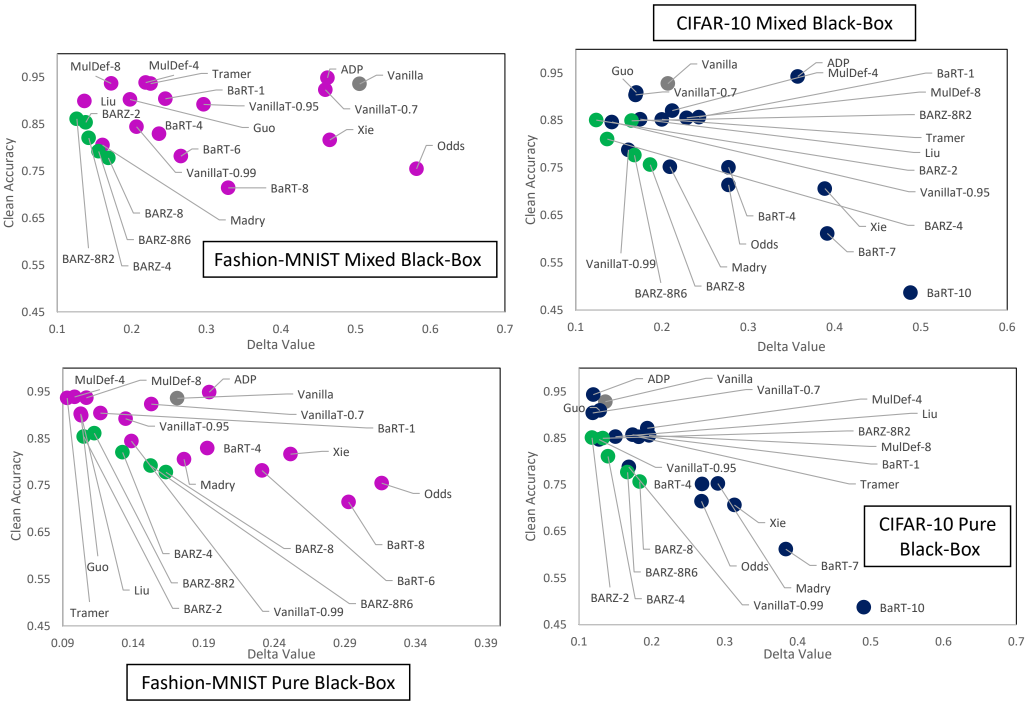


FIGURE 10: The $\delta$ metric vs clean accuracy for the targeted mixed black-box and targeted pure black box attacks. The BARZ results are shown in green and the vanilla result is shown in gray.

### B. FASHION-MNIST: ATTACKS AND DEFENSES

The results for Fashion-MNIST are described in Tables 3, 4, 5, 6, and 7. Recall the formula for the $\delta$ metric:

$$\delta = \gamma + (p - \gamma)\alpha = p - (p - \gamma)(1 - \alpha) = p - p_d \cdot \beta, \tag{6}$$

where $p$ is the clean accuracy of the vanilla classifier (i.e., no defense at all and without any adversarial presence), $\gamma$ is the drop in clean accuracy, i.e., $\gamma = p - p_d$ for $p_d$ representing the clean accuracy of the defense while no attacker is present, $\alpha$ is the attacker's success rate against the defense and $\beta$ is the robust accuracy or defense success rate (also called defense accuracy) and is equal to $1 - \alpha$.

$\delta$ can be used to measure the effectiveness of different defenses, the smaller the better. If two defenses offer roughly the same $\delta$, then it makes sense to consider their $(\gamma, \alpha)$ pairs and choose the defense that either has the smaller $\alpha$ or the smaller $\gamma$.

For Fashion-MNIST and CIFAR-10, $p = 0.9356$ and $0.9278$, respectively. The value of $\delta$ is computed by combining $p$ of the vanilla classifier and $p_d$ of the considered defense, and by looking at the best attack among all implemented attacks on the given defense (this corresponds to the maximum over the attacker's success rates $\alpha$ for the specific set of attacks considered, similarly, this corresponds to the minimum over the various defense success rates $\beta$). For example, the $\delta$ metric for BARZ-8 in Table 3 is computed as follows: we substitute $p = 0.9356$, $p_d = 0.7779$, and the minimal $\beta = 0.986$ among all (currently known) targeted mixed black-box attacks (in this case corresponding to the FGSM-T attack) into formula (Eq. 6) for $\delta$. This results in $\delta = 0.168591$.

**Discussion.** We have the following observations from the aforementioned tables:

1) The BARZ family achieves the smallest $\delta$ for any attack scenario. Figures 5, 8, 2 and 10 reflect this fact.
2) Many defenses (such as Guo, Liu, ADP, Tramer) have a very high clean accuracy (i.e., close to the clean accuracy of the vanilla classifier), but have a very large $\delta$. If we have a close look at the results presented in Figures 6 and 9 or Tables 3, 5, 6 and 7, we can see that they are vulnerable to black-box attacks. In other words, they offer no security.
3) By combining the drop $\gamma$ in clean accuracy and the increment in robust accuracy $\beta$, the $\delta$ metric can be used for understanding how well a defense performs in the presence of attackers. In order to have a further detailed evaluation, we need to separately look at the attack success rate $\alpha$ (or, equivalently, robust accuracy $\beta$) and clean accuracy of the defense $p_d$.
4) From Tables 3, 4, 5 and 6 we conclude that mixed black-box attacks are more efficient than pure black-box attacks and untargeted black-box attacks are stronger than targeted ones. When looking at Table 7, boundary attacks are much stronger than mixed and pure black-box attacks.

TABLE 3: Fashion-MNIST targeted mixed black-box attack results. Note the $\beta$ column refers to the minimum robust accuracy across all targeted mixed black-box attacks.

| | FGSM-T | IFGSM-T | MIM-T | PGD-T | CW-T | EAD-T | $\beta$ | $p_d$ | $\delta$ |
|---|---|---|---|---|---|---|---|---|---|
| Vanilla | 0.707 | 0.529 | 0.46 | 0.531 | 0.993 | 0.991 | 0.46 | 0.9356 | 0.505224 |
| VanillaT-0.7 | 0.757 | 0.601 | 0.516 | 0.588 | 0.989 | 0.989 | 0.516 | 0.9232 | 0.459229 |
| VanillaT-0.95 | 0.848 | 0.767 | 0.717 | 0.787 | 0.997 | 0.997 | 0.717 | 0.892 | 0.296036 |
| VanillaT-0.99 | 0.935 | 0.891 | 0.864 | 0.891 | 0.998 | 0.998 | 0.864 | 0.8442 | 0.206211 |
| Liu | 0.91 | 0.899 | 0.889 | 0.894 | 0.996 | 0.996 | 0.889 | 0.899 | 0.136389 |
| ADP | 0.793 | 0.515 | 0.499 | 0.519 | 0.987 | 0.986 | 0.499 | 0.9486 | 0.462249 |
| Xie | 0.711 | 0.62 | 0.576 | 0.626 | 0.963 | 0.954 | 0.576 | 0.8164 | 0.465354 |
| Madry | 0.963 | 0.996 | 0.995 | 0.996 | 0.983 | 0.962 | 0.962 | 0.8055 | 0.160709 |
| Tramer | 0.808 | 0.833 | 0.759 | 0.826 | 0.992 | 0.992 | 0.759 | 0.9361 | 0.2251 |
| MulDef-4 | 0.831 | 0.804 | 0.764 | 0.811 | 0.988 | 0.984 | 0.764 | 0.9386 | 0.21851 |
| MulDef-8 | 0.837 | 0.851 | 0.815 | 0.84 | 0.994 | 0.992 | 0.815 | 0.9365 | 0.172353 |
| Guo | 0.818 | 0.917 | 0.879 | 0.914 | 1 | 0.999 | 0.818 | 0.9023 | 0.197519 |
| BARZ-2 | 0.934 | 0.961 | 0.949 | 0.961 | 0.999 | 0.998 | 0.934 | 0.8537 | 0.138244 |
| BARZ-4 | 0.967 | 0.987 | 0.973 | 0.987 | 1 | 1 | 0.967 | 0.8204 | 0.142273 |
| BARZ-8 | 0.986 | 0.998 | 0.995 | 0.997 | 1 | 1 | 0.986 | 0.7779 | 0.168591 |
| BARZ-8R2 | 0.94 | 0.974 | 0.959 | 0.964 | 0.998 | 0.999 | 0.94 | 0.8611 | 0.126166 |
| BARZ-8R6 | 0.989 | 0.998 | 0.985 | 0.994 | 1 | 0.999 | 0.985 | 0.792 | 0.15548 |
| Odds | 0.671 | 0.54 | 0.469 | 0.548 | 0.991 | 0.987 | 0.469 | 0.7547 | 0.581646 |
| BaRT-1 | 0.836 | 0.807 | 0.764 | 0.805 | 0.978 | 0.976 | 0.764 | 0.9039 | 0.24502 |
| BaRT-4 | 0.872 | 0.848 | 0.843 | 0.848 | 0.941 | 0.959 | 0.843 | 0.8294 | 0.236416 |
| BaRT-6 | 0.882 | 0.876 | 0.857 | 0.877 | 0.935 | 0.947 | 0.857 | 0.7817 | 0.265683 |
| BaRT-8 | 0.866 | 0.873 | 0.849 | 0.858 | 0.942 | 0.958 | 0.849 | 0.7144 | 0.329074 |

TABLE 4: Fashion-MNIST targeted pure black-box attack results. Note the $\beta$ column refers to the minimum robust accuracy across all targeted pure black-box attacks.

| | FGSM-T | IFGSM-T | MIM-T | PGD-T | CW-T | EAD-T | $\beta$ | $p_d$ | $\delta$ |
|---|---|---|---|---|---|---|---|---|---|
| Vanilla | 0.865 | 0.889 | 0.817 | 0.879 | 0.995 | 0.992 | 0.817 | 0.9356 | 0.171215 |
| VanillaT-0.7 | 0.892 | 0.912 | 0.848 | 0.909 | 0.995 | 0.994 | 0.848 | 0.9232 | 0.152726 |
| VanillaT-0.95 | 0.937 | 0.945 | 0.898 | 0.937 | 0.998 | 0.999 | 0.898 | 0.892 | 0.134584 |
| VanillaT-0.99 | 0.962 | 0.965 | 0.944 | 0.965 | 1 | 1 | 0.944 | 0.8442 | 0.138675 |
| Liu | 0.948 | 0.962 | 0.926 | 0.949 | 0.999 | 0.998 | 0.926 | 0.899 | 0.103126 |
| ADP | 0.875 | 0.839 | 0.782 | 0.861 | 0.992 | 0.99 | 0.782 | 0.9486 | 0.193795 |
| Xie | 0.882 | 0.899 | 0.838 | 0.914 | 0.974 | 0.973 | 0.838 | 0.8164 | 0.251457 |
| Madry | 0.952 | 0.971 | 0.971 | 0.972 | 0.951 | 0.943 | 0.943 | 0.8055 | 0.176014 |
| Tramer | 0.9 | 0.967 | 0.917 | 0.968 | 0.995 | 0.993 | 0.9 | 0.9361 | 0.09311 |
| MulDef-4 | 0.895 | 0.941 | 0.892 | 0.942 | 0.992 | 0.992 | 0.892 | 0.9386 | 0.098369 |
| MulDef-8 | 0.885 | 0.946 | 0.902 | 0.958 | 0.994 | 0.992 | 0.885 | 0.9365 | 0.106798 |
| Guo | 0.923 | 0.982 | 0.959 | 0.981 | 0.993 | 0.994 | 0.923 | 0.9023 | 0.102777 |
| BARZ-2 | 0.973 | 0.989 | 0.978 | 0.988 | 0.998 | 0.998 | 0.973 | 0.8537 | 0.10495 |
| BARZ-4 | 0.979 | 0.995 | 0.985 | 0.998 | 1 | 1 | 0.979 | 0.8204 | 0.132428 |
| BARZ-8 | 0.993 | 0.998 | 0.993 | 0.999 | 1 | 1 | 0.993 | 0.7779 | 0.163145 |
| BARZ-8R2 | 0.956 | 0.99 | 0.976 | 0.994 | 0.997 | 0.999 | 0.956 | 0.8611 | 0.112388 |
| BARZ-8R6 | 0.989 | 0.998 | 0.991 | 0.998 | 1 | 1 | 0.989 | 0.792 | 0.152312 |
| Odds | 0.865 | 0.891 | 0.821 | 0.882 | 0.998 | 0.993 | 0.821 | 0.7547 | 0.315991 |
| BaRT-1 | 0.906 | 0.944 | 0.906 | 0.941 | 0.993 | 0.992 | 0.906 | 0.9039 | 0.116667 |
| BaRT-4 | 0.907 | 0.939 | 0.896 | 0.936 | 0.976 | 0.977 | 0.896 | 0.8294 | 0.192458 |
| BaRT-6 | 0.914 | 0.927 | 0.901 | 0.931 | 0.966 | 0.954 | 0.901 | 0.7817 | 0.231288 |
| BaRT-8 | 0.92 | 0.907 | 0.9 | 0.938 | 0.959 | 0.945 | 0.9 | 0.7144 | 0.29264 |

5) BARZ can realize different combinations of defender accuracy $p_d$ and attacker's success rate $\alpha$ by tuning the number of classifiers in the defense.
6) BARZ-8R2, Madry and MulDef have the smallest $\delta$ values for boundary attacks. For the BARZ and MulDef defenses the reason is that for a given input $x$, for each evaluation, these defenses introduce some randomness. As a consequence, the output class label can be changed. This strongly affects the efficiency of boundary attacks which need to accurately estimate the gradients of many images (and due to the introduced randomness these estimates become less accurate).

### C. CIFAR-10: ATTACKS AND DEFENSES

The results for CIFAR-10 are described in Tables 8, 9, 10, 11 and 12.

**Discussion.** We have the following observations from aforementioned tables (identical to Fashion-MNIST with a slight difference in item 6):

1) The BARZ family achieves the smallest $\delta$ for any attack scenario. Figures 5, 8, 2 and 10 reflect this fact.
2) Many defenses (such as Guo, Liu, ADP, Tramer) have a very high clean accuracy (i.e., close to the clean accuracy of the vanilla classifier), but have a very large

TABLE 5: Fashion-MNIST untargeted mixed black-box attack results. Note the $\beta$ column refers to the minimum robust accuracy across all untargeted mixed black-box attacks.

| | FGSM-U | IFGSM-U | MIM-U | PGD-U | CW-U | EAD-U | $\beta$ | $p_d$ | $\delta$ |
|---|---|---|---|---|---|---|---|---|---|
| Vanilla | 0.234 | 0.123 | 0.111 | 0.118 | 0.961 | 0.939 | 0.111 | 0.9356 | 0.831748 |
| VanillaT-0.7 | 0.345 | 0.172 | 0.184 | 0.178 | 0.972 | 0.953 | 0.172 | 0.9232 | 0.77681 |
| VanillaT-0.95 | 0.573 | 0.307 | 0.335 | 0.304 | 0.994 | 0.991 | 0.304 | 0.892 | 0.664432 |
| VanillaT-0.99 | 0.717 | 0.501 | 0.518 | 0.501 | 0.994 | 0.996 | 0.501 | 0.8442 | 0.512656 |
| Liu | 0.683 | 0.562 | 0.609 | 0.554 | 0.987 | 0.983 | 0.554 | 0.899 | 0.437554 |
| ADP | 0.141 | 0.085 | 0.104 | 0.089 | 0.934 | 0.909 | 0.085 | 0.9486 | 0.854969 |
| Xie | 0.212 | 0.197 | 0.19 | 0.201 | 0.794 | 0.772 | 0.19 | 0.8164 | 0.780484 |
| Madry | 0.489 | 0.959 | 0.954 | 0.963 | 0.921 | 0.866 | 0.489 | 0.8055 | 0.541711 |
| Tramer | 0.344 | 0.379 | 0.35 | 0.395 | 0.977 | 0.971 | 0.344 | 0.9361 | 0.613582 |
| MulDef-4 | 0.35 | 0.416 | 0.356 | 0.404 | 0.947 | 0.94 | 0.35 | 0.9386 | 0.60709 |
| MulDef-8 | 0.377 | 0.467 | 0.405 | 0.465 | 0.951 | 0.953 | 0.377 | 0.9365 | 0.58254 |
| Guo | 0.376 | 0.55 | 0.496 | 0.555 | 0.989 | 0.982 | 0.376 | 0.9023 | 0.596335 |
| BARZ-2 | 0.696 | 0.78 | 0.731 | 0.771 | 0.998 | 0.996 | 0.696 | 0.8537 | 0.341425 |
| BARZ-4 | 0.82 | 0.851 | 0.817 | 0.847 | 1 | 1 | 0.817 | 0.8204 | 0.265333 |
| BARZ-8 | 0.906 | 0.941 | 0.92 | 0.953 | 1 | 1 | 0.906 | 0.7779 | 0.230823 |
| BARZ-8R2 | 0.752 | 0.796 | 0.744 | 0.73 | 0.996 | 0.986 | 0.73 | 0.8611 | 0.306997 |
| BARZ-8R6 | 0.873 | 0.92 | 0.911 | 0.925 | 1 | 1 | 0.873 | 0.792 | 0.244184 |
| Odds | 0.224 | 0.153 | 0.135 | 0.154 | 0.944 | 0.937 | 0.135 | 0.7547 | 0.833716 |
| BaRT-1 | 0.359 | 0.383 | 0.339 | 0.376 | 0.861 | 0.877 | 0.339 | 0.9039 | 0.629178 |
| BaRT-4 | 0.41 | 0.399 | 0.384 | 0.406 | 0.779 | 0.791 | 0.384 | 0.8294 | 0.61711 |
| BaRT-6 | 0.37 | 0.437 | 0.417 | 0.411 | 0.724 | 0.726 | 0.37 | 0.7817 | 0.646371 |
| BaRT-8 | 0.4 | 0.41 | 0.365 | 0.392 | 0.706 | 0.696 | 0.365 | 0.7144 | 0.674844 |

TABLE 6: Fashion-MNIST untargeted pure black-box attack results. Note the $\beta$ column refers to the minimum robust accuracy across all untargeted pure black-box attacks.

| | FGSM-U | IFGSM-U | MIM-U | PGD-U | CW-U | EAD-U | $\beta$ | $p_d$ | $\delta$ |
|---|---|---|---|---|---|---|---|---|---|
| Vanilla | 0.429 | 0.363 | 0.351 | 0.374 | 0.914 | 0.905 | 0.351 | 0.9356 | 0.607204 |
| VanillaT-0.7 | 0.536 | 0.415 | 0.399 | 0.42 | 0.935 | 0.93 | 0.399 | 0.9232 | 0.567243 |
| VanillaT-0.95 | 0.688 | 0.512 | 0.505 | 0.513 | 0.964 | 0.962 | 0.505 | 0.892 | 0.48514 |
| VanillaT-0.99 | 0.801 | 0.622 | 0.621 | 0.625 | 0.978 | 0.977 | 0.621 | 0.8442 | 0.411352 |
| Liu | 0.753 | 0.67 | 0.67 | 0.682 | 0.96 | 0.959 | 0.67 | 0.899 | 0.33327 |
| ADP | 0.398 | 0.357 | 0.321 | 0.35 | 0.926 | 0.923 | 0.321 | 0.9486 | 0.631099 |
| Xie | 0.386 | 0.401 | 0.395 | 0.409 | 0.789 | 0.754 | 0.386 | 0.8164 | 0.62047 |
| Madry | 0.655 | 0.789 | 0.787 | 0.789 | 0.717 | 0.705 | 0.655 | 0.8055 | 0.407998 |
| Tramer | 0.538 | 0.6 | 0.548 | 0.6 | 0.926 | 0.923 | 0.538 | 0.9361 | 0.431978 |
| MulDef-4 | 0.531 | 0.545 | 0.517 | 0.562 | 0.932 | 0.923 | 0.517 | 0.9386 | 0.450344 |
| MulDef-8 | 0.495 | 0.556 | 0.516 | 0.566 | 0.924 | 0.929 | 0.495 | 0.9365 | 0.472033 |
| Guo | 0.61 | 0.725 | 0.674 | 0.729 | 0.899 | 0.897 | 0.61 | 0.9023 | 0.385197 |
| BARZ-2 | 0.791 | 0.845 | 0.798 | 0.843 | 0.954 | 0.956 | 0.791 | 0.8537 | 0.260323 |
| BARZ-4 | 0.834 | 0.879 | 0.845 | 0.878 | 0.97 | 0.971 | 0.834 | 0.8204 | 0.251386 |
| BARZ-8 | 0.899 | 0.929 | 0.903 | 0.937 | 0.983 | 0.983 | 0.899 | 0.7779 | 0.236268 |
| BARZ-8R2 | 0.756 | 0.794 | 0.765 | 0.827 | 0.951 | 0.957 | 0.756 | 0.8611 | 0.284608 |
| BARZ-8R6 | 0.865 | 0.918 | 0.882 | 0.92 | 0.982 | 0.977 | 0.865 | 0.792 | 0.25052 |
| Odds | 0.43 | 0.391 | 0.366 | 0.397 | 0.94 | 0.926 | 0.366 | 0.7547 | 0.65938 |
| BaRT-1 | 0.548 | 0.536 | 0.488 | 0.533 | 0.893 | 0.888 | 0.488 | 0.9039 | 0.494497 |
| BaRT-4 | 0.547 | 0.535 | 0.5 | 0.524 | 0.782 | 0.799 | 0.5 | 0.8294 | 0.5209 |
| BaRT-6 | 0.52 | 0.502 | 0.469 | 0.507 | 0.74 | 0.725 | 0.469 | 0.7817 | 0.568983 |
| BaRT-8 | 0.47 | 0.453 | 0.459 | 0.466 | 0.675 | 0.683 | 0.453 | 0.7144 | 0.611977 |

$\delta$. If we have a close look at the results presented in Figures 6 and 9 or Tables 8, 10, 11 and 12, we can see that they are vulnerable to black-box attacks. In other words, they offer no security.

3) By combining the drop $\gamma$ in clean accuracy and the increment in robust accuracy $\beta$, the $\delta$ metric can be used for understanding how well a defense performs in the presence of attackers. In order to have a further detailed evaluation, we need to separately look at the attack success rate $\alpha$ (or, equivalently, robust accuracy $\beta$) and clean accuracy of the defense $p_d$.
4) From Tables 8, 9, 10, and 11 we conclude that mixed black-box attacks are more efficient than pure black-box attacks and untargeted black-box attacks are stronger than targeted ones. When looking at Table 12, boundary attacks are much stronger than mixed and pure black-box attacks.
5) BARZ can realize different combinations of defender accuracy $p_d$ and attacker's success rate $\alpha$ by tuning the number of classifiers in the defense.
6) BARZ-8R6/2, Xie and MulDef have the smallest $\delta$ values for boundary attacks. The reason is that for a given input $x$, for each evaluation, these defenses introduce some randomness. As a consequence, the output class

TABLE 7: Fashion-MNIST untargeted boundary attack results. Note the $\beta$ column refers to the minimum robust accuracy across all untargeted boundary black-box attacks.

| | HSJA | RayS | $\beta$ | $p_d$ | $\delta$ |
|---|---|---|---|---|---|
| Vanilla | 0 | 0.09 | 0 | 0.9356 | 0.9356 |
| VanillaT-0.7 | 0 | 0.1 | 0 | 0.9232 | 0.9356 |
| VanillaT-0.95 | 0 | 0.18 | 0 | 0.892 | 0.9356 |
| VanillaT-0.99 | 0 | 0.47 | 0 | 0.8442 | 0.9356 |
| Liu | 0 | 0.18 | 0 | 0.899 | 0.9356 |
| ADP | 0 | 0.04 | 0 | 0.9486 | 0.9356 |
| Xie | 0.85 | 0.63 | 0.63 | 0.8164 | 0.421268 |
| Madry | 0.99 | 0.96 | 0.96 | 0.8055 | 0.16232 |
| Tramer | 0 | 0.18 | 0 | 0.9361 | 0.9356 |
| MulDef-4 | 0.82 | 0.66 | 0.66 | 0.9386 | 0.316124 |
| MulDef-8 | 0.92 | 0.7 | 0.7 | 0.9365 | 0.28005 |
| Guo | 0 | 0.32 | 0 | 0.9023 | 0.9356 |
| BARZ-2 | 0.1 | 0.61 | 0.1 | 0.8537 | 0.85023 |
| BARZ-4 | 0.53 | 0.93 | 0.53 | 0.8204 | 0.500788 |
| BARZ-8 | 0.89 | 1 | 0.89 | 0.7779 | 0.243269 |
| BARZ-8R2 | 0.99 | 0.9 | 0.9 | 0.8611 | 0.16061 |
| BARZ-8R6 | 1 | 0.92 | 0.92 | 0.792 | 0.20696 |
| Odds | NA | NA | NA | NA | NA |
| BaRT-1 | NA | NA | NA | NA | NA |
| BaRT-4 | NA | NA | NA | NA | NA |
| BaRT-6 | NA | NA | NA | NA | NA |
| BaRT-8 | NA | NA | NA | NA | NA |

TABLE 8: CIFAR-10 targeted mixed black-box attack results. Note the $\beta$ column refers to the minimum robust accuracy across all targeted mixed black-box attacks.

| | FGSM-T | IFGSM-T | MIM-T | PGD-T | CW-T | EAD-T | $\beta$ | $p_d$ | $\delta$ |
|---|---|---|---|---|---|---|---|---|---|
| Vanilla | 0.866 | 0.861 | 0.777 | 0.848 | 0.991 | 0.991 | 0.777 | 0.9278 | 0.206899 |
| VanillaT-0.7 | 0.911 | 0.891 | 0.839 | 0.893 | 0.995 | 0.996 | 0.839 | 0.9038 | 0.169512 |
| VanillaT-0.95 | 0.961 | 0.956 | 0.928 | 0.955 | 0.998 | 0.999 | 0.928 | 0.8468 | 0.14197 |
| VanillaT-0.99 | 0.984 | 0.983 | 0.973 | 0.984 | 1 | 1 | 0.973 | 0.7879 | 0.161173 |
| Liu | 0.939 | 0.942 | 0.883 | 0.943 | 1 | 1 | 0.883 | 0.8528 | 0.174778 |
| ADP | 0.843 | 0.698 | 0.605 | 0.712 | 0.995 | 0.987 | 0.605 | 0.943 | 0.357285 |
| Xie | 0.83 | 0.821 | 0.763 | 0.858 | 0.982 | 0.981 | 0.763 | 0.7064 | 0.388817 |
| Madry | 0.96 | 0.979 | 0.955 | 0.98 | 0.999 | 0.995 | 0.955 | 0.7524 | 0.209258 |
| Tramer | 0.901 | 0.934 | 0.854 | 0.942 | 0.998 | 0.996 | 0.854 | 0.8524 | 0.19985 |
| MulDef-4 | 0.889 | 0.902 | 0.822 | 0.913 | 0.987 | 0.986 | 0.822 | 0.8709 | 0.21192 |
| MulDef-8 | 0.89 | 0.908 | 0.817 | 0.893 | 0.983 | 0.992 | 0.817 | 0.8556 | 0.228775 |
| Guo | 0.891 | 0.902 | 0.833 | 0.901 | 0.994 | 0.991 | 0.833 | 0.9092 | 0.170436 |
| BARZ-2 | 0.969 | 0.971 | 0.945 | 0.971 | 1 | 0.999 | 0.945 | 0.8507 | 0.123889 |
| BARZ-4 | 0.986 | 0.987 | 0.976 | 0.984 | 1 | 1 | 0.976 | 0.8106 | 0.136654 |
| BARZ-8 | 0.993 | 0.991 | 0.98 | 0.993 | 1 | 1 | 0.98 | 0.7565 | 0.18643 |
| BARZ-8R2 | 0.941 | 0.962 | 0.898 | 0.959 | 0.999 | 1 | 0.898 | 0.8495 | 0.164949 |
| BARZ-8R6 | 0.986 | 0.989 | 0.978 | 0.993 | 1 | 1 | 0.978 | 0.7767 | 0.168187 |
| Odds | 0.941 | 0.938 | 0.911 | 0.93 | 0.997 | 0.991 | 0.911 | 0.7141 | 0.277255 |
| BaRT-1 | 0.886 | 0.872 | 0.799 | 0.875 | 0.975 | 0.973 | 0.799 | 0.8571 | 0.242977 |
| BaRT-4 | 0.91 | 0.914 | 0.866 | 0.901 | 0.971 | 0.974 | 0.866 | 0.7513 | 0.277174 |
| BaRT-7 | 0.904 | 0.921 | 0.877 | 0.917 | 0.963 | 0.946 | 0.877 | 0.6114 | 0.391602 |
| BaRT-10 | 0.91 | 0.911 | 0.903 | 0.921 | 0.928 | 0.944 | 0.903 | 0.4869 | 0.488129 |

label can be changed. This strongly affects the efficiency of boundary attacks which need to accurately estimate the gradients of many images (and due to the introduced randomness these estimates become less accurate).

## APPENDIX. EXPERIMENTAL ATTACK RESULTS

As we mentioned in the main body of the paper, the mixed black-box attack can be thought of as an extension of the Papernot attack. In this section we give experimental evidence with the CIFAR-10 dataset to support our claims. In Figure 11 we show a graphical representation of the attack success rate as a function of training data. On the $x$-axis of the graph is the percent of training data used at the start of the attack to build the synthetic model. On the $y$-axis of the graph is the attack success rate of the attack on a vanilla (undefended) model.

For this experiment we fix several variables in order to make the comparison. We use the FGSM attack on the synthetic model with $\epsilon = 0.05$ to create adversarial samples. We fix the number of iterations in the attack to be $N = 4$ for all the experiments and $\lambda = 0.1$. In Papernot's original attack on an MNIST classifier $0.3\%$ of the original training data is used. We show that as you increase the amount of training data (and subsequent queries) the attack success rate

TABLE 9: CIFAR-10 targeted pure black-box attack results. Note the $\beta$ column refers to the minimum robust accuracy across all targeted pure black-box attacks.

| | FGSM-T | IFGSM-T | MIM-T | PGD-T | CW-T | EAD-T | $\beta$ | $p_d$ | $\delta$ |
|---|---|---|---|---|---|---|---|---|---|
| Vanilla | 0.902 | 0.917 | 0.853 | 0.924 | 0.984 | 0.984 | 0.853 | 0.9278 | 0.136387 |
| VanillaT-0.7 | 0.93 | 0.947 | 0.895 | 0.947 | 0.991 | 0.989 | 0.895 | 0.9038 | 0.118899 |
| VanillaT-0.95 | 0.962 | 0.974 | 0.945 | 0.972 | 0.996 | 0.996 | 0.945 | 0.8468 | 0.127574 |
| VanillaT-0.99 | 0.978 | 0.985 | 0.964 | 0.987 | 0.997 | 0.996 | 0.964 | 0.7879 | 0.168264 |
| Liu | 0.939 | 0.965 | 0.912 | 0.971 | 0.994 | 0.993 | 0.912 | 0.8528 | 0.150046 |
| ADP | 0.905 | 0.933 | 0.857 | 0.935 | 0.987 | 0.987 | 0.857 | 0.943 | 0.119649 |
| Xie | 0.898 | 0.929 | 0.87 | 0.926 | 0.957 | 0.961 | 0.87 | 0.7064 | 0.313232 |
| Madry | 0.904 | 0.917 | 0.894 | 0.914 | 0.895 | 0.847 | 0.847 | 0.7524 | 0.290517 |
| Tramer | 0.904 | 0.958 | 0.875 | 0.955 | 0.977 | 0.981 | 0.875 | 0.8524 | 0.18195 |
| MulDef-4 | 0.883 | 0.933 | 0.843 | 0.937 | 0.981 | 0.984 | 0.843 | 0.8709 | 0.193631 |
| MulDef-8 | 0.879 | 0.952 | 0.855 | 0.944 | 0.982 | 0.978 | 0.855 | 0.8556 | 0.196262 |
| Guo | 0.911 | 0.938 | 0.879 | 0.935 | 0.983 | 0.985 | 0.879 | 0.9092 | 0.128613 |
| BARZ-2 | 0.955 | 0.974 | 0.952 | 0.974 | 0.995 | 0.995 | 0.952 | 0.8507 | 0.117934 |
| BARZ-4 | 0.972 | 0.992 | 0.972 | 0.991 | 0.995 | 0.995 | 0.972 | 0.8106 | 0.139897 |
| BARZ-8 | 0.985 | 0.993 | 0.984 | 0.994 | 0.998 | 0.998 | 0.984 | 0.7565 | 0.183404 |
| BARZ-8R2 | 0.947 | 0.977 | 0.936 | 0.97 | 0.994 | 0.993 | 0.936 | 0.8495 | 0.132668 |
| BARZ-8R6 | 0.98 | 0.99 | 0.981 | 0.992 | 0.998 | 0.998 | 0.98 | 0.7767 | 0.166634 |
| Odds | 0.956 | 0.958 | 0.924 | 0.965 | 0.987 | 0.986 | 0.924 | 0.7141 | 0.267972 |
| BaRT-1 | 0.909 | 0.943 | 0.88 | 0.956 | 0.979 | 0.979 | 0.88 | 0.8571 | 0.173552 |
| BaRT-4 | 0.908 | 0.952 | 0.877 | 0.933 | 0.979 | 0.963 | 0.877 | 0.7513 | 0.26891 |
| BaRT-7 | 0.911 | 0.931 | 0.89 | 0.923 | 0.952 | 0.948 | 0.89 | 0.6114 | 0.383654 |
| BaRT-10 | 0.903 | 0.916 | 0.898 | 0.912 | 0.932 | 0.931 | 0.898 | 0.4869 | 0.490564 |

TABLE 10: CIFAR-10 untargeted mixed black-box attack results. Note the $\beta$ column refers to the minimum robust accuracy across all untargeted mixed black-box attacks.

| | FGSM-U | IFGSM-U | MIM-U | PGD-U | CW-U | EAD-U | $\beta$ | $p_d$ | $\delta$ |
|---|---|---|---|---|---|---|---|---|---|
| Vanilla | 0.334 | 0.387 | 0.259 | 0.374 | 0.986 | 0.987 | 0.259 | 0.9278 | 0.6875 |
| VanillaT-0.7 | 0.547 | 0.591 | 0.523 | 0.591 | 0.992 | 0.987 | 0.523 | 0.9038 | 0.455113 |
| VanillaT-0.95 | 0.803 | 0.812 | 0.773 | 0.818 | 0.999 | 1 | 0.773 | 0.8468 | 0.273224 |
| VanillaT-0.99 | 0.928 | 0.916 | 0.892 | 0.909 | 1 | 1 | 0.892 | 0.7879 | 0.224993 |
| Liu | 0.731 | 0.703 | 0.628 | 0.707 | 0.994 | 0.997 | 0.628 | 0.8528 | 0.392242 |
| ADP | 0.332 | 0.224 | 0.147 | 0.226 | 0.992 | 0.985 | 0.147 | 0.943 | 0.789179 |
| Xie | 0.296 | 0.377 | 0.262 | 0.372 | 0.837 | 0.857 | 0.262 | 0.7064 | 0.742723 |
| Madry | 0.777 | 0.838 | 0.767 | 0.838 | 0.989 | 0.982 | 0.767 | 0.7524 | 0.350709 |
| Tramer | 0.568 | 0.616 | 0.442 | 0.638 | 0.985 | 0.977 | 0.442 | 0.8524 | 0.551039 |
| MulDef-4 | 0.493 | 0.525 | 0.373 | 0.536 | 0.932 | 0.924 | 0.373 | 0.8709 | 0.602954 |
| MulDef-8 | 0.491 | 0.568 | 0.397 | 0.557 | 0.916 | 0.915 | 0.397 | 0.8556 | 0.588127 |
| Guo | 0.483 | 0.568 | 0.445 | 0.556 | 0.992 | 0.985 | 0.445 | 0.9092 | 0.523206 |
| BARZ-2 | 0.807 | 0.813 | 0.752 | 0.825 | 1 | 0.997 | 0.752 | 0.8507 | 0.288074 |
| BARZ-4 | 0.916 | 0.901 | 0.883 | 0.912 | 1 | 0.999 | 0.883 | 0.8106 | 0.21204 |
| BARZ-8 | 0.962 | 0.955 | 0.926 | 0.95 | 1 | 1 | 0.926 | 0.7565 | 0.227281 |
| BARZ-8R2 | 0.767 | 0.755 | 0.68 | 0.756 | 0.99 | 0.989 | 0.68 | 0.8495 | 0.35014 |
| BARZ-8R6 | 0.932 | 0.935 | 0.898 | 0.942 | 1 | 1 | 0.898 | 0.7767 | 0.230323 |
| Odds | 0.646 | 0.664 | 0.578 | 0.673 | 0.974 | 0.977 | 0.578 | 0.7141 | 0.51505 |
| BaRT-1 | 0.507 | 0.553 | 0.398 | 0.543 | 0.917 | 0.933 | 0.398 | 0.8571 | 0.586674 |
| BaRT-4 | 0.482 | 0.554 | 0.443 | 0.577 | 0.803 | 0.788 | 0.443 | 0.7513 | 0.594974 |
| BaRT-7 | 0.447 | 0.534 | 0.426 | 0.535 | 0.704 | 0.678 | 0.426 | 0.6114 | 0.667344 |
| BaRT-10 | 0.391 | 0.465 | 0.382 | 0.492 | 0.581 | 0.583 | 0.382 | 0.4869 | 0.741804 |

increases. When the percent of training data reaches $100\%$ we have what we refer to as the mixed black-box attack. This represents a substantial increase in the success rate of the attack. In our experiment for CIFAR-10 we show it increases from $24.7\%$ to $66.6\%$, an attack success rate increase of $41.9\%$.

On certain defenses the mixed black-box attack also outperforms other attacks. For example consider the randomized Xie defense. The robust accuracy for CIFAR-10 is $85\%$ under untargeted boundary attacks. However, the robust accuracy is the lowest under the untargeted mixed black-box attack, at just $26.2\%$. Likewise, the mixed black-box attack outperforms the boundary attacks on MulDef-4 and MulDef-8 (although pure black-box attacks here are the strongest by a slim $1\%$ margin). If we consider Fashion-MNIST we also can see defenses on which the mixed black-box outperforms the other attacks. On Fashion-MNIST the lowest robust accuracy is obtained under the mixed black-box attack for the Xie, MulDef and Madry defenses.

To conclude the purpose of our analysis here is two-fold. First through our experiments we show that when the conditions are held the same, the mixed black-box attack clearly outperforms the original Papernot attack. Second we show the mixed black-box attack is the most effective attack

TABLE 11: CIFAR-10 untargeted pure black-box attack results. Note the $\beta$ column refers to the minimum robust accuracy across all untargeted pure black-box attacks.

| | FGSM-U | IFGSM-U | MIM-U | PGD-U | CW-U | EAD-U | $\beta$ | $p_d$ | $\delta$ |
|---|---|---|---|---|---|---|---|---|---|
| Vanilla | 0.443 | 0.453 | 0.384 | 0.455 | 0.923 | 0.919 | 0.384 | 0.9278 | 0.571525 |
| VanillaT-0.7 | 0.587 | 0.593 | 0.535 | 0.605 | 0.946 | 0.943 | 0.535 | 0.9038 | 0.444267 |
| VanillaT-0.95 | 0.804 | 0.772 | 0.718 | 0.78 | 0.974 | 0.974 | 0.718 | 0.8468 | 0.319798 |
| VanillaT-0.99 | 0.899 | 0.856 | 0.831 | 0.864 | 0.987 | 0.989 | 0.831 | 0.7879 | 0.273055 |
| Liu | 0.735 | 0.701 | 0.649 | 0.691 | 0.972 | 0.972 | 0.649 | 0.8528 | 0.374333 |
| ADP | 0.487 | 0.475 | 0.385 | 0.485 | 0.932 | 0.932 | 0.385 | 0.943 | 0.564745 |
| Xie | 0.408 | 0.438 | 0.352 | 0.409 | 0.694 | 0.705 | 0.352 | 0.7064 | 0.679147 |
| Madry | 0.55 | 0.602 | 0.534 | 0.6 | 0.694 | 0.663 | 0.534 | 0.7524 | 0.526018 |
| Tramer | 0.563 | 0.541 | 0.441 | 0.561 | 0.849 | 0.845 | 0.441 | 0.8524 | 0.551892 |
| MulDef-4 | 0.496 | 0.493 | 0.36 | 0.482 | 0.861 | 0.859 | 0.36 | 0.8709 | 0.614276 |
| MulDef-8 | 0.515 | 0.51 | 0.384 | 0.499 | 0.854 | 0.844 | 0.384 | 0.8556 | 0.59925 |
| Guo | 0.492 | 0.542 | 0.446 | 0.565 | 0.904 | 0.899 | 0.446 | 0.9092 | 0.522297 |
| BARZ-2 | 0.795 | 0.759 | 0.699 | 0.793 | 0.97 | 0.968 | 0.699 | 0.8507 | 0.333161 |
| BARZ-4 | 0.887 | 0.879 | 0.835 | 0.882 | 0.992 | 0.991 | 0.835 | 0.8106 | 0.250949 |
| BARZ-8 | 0.947 | 0.932 | 0.892 | 0.928 | 0.998 | 0.997 | 0.892 | 0.7565 | 0.253002 |
| BARZ-8R2 | 0.766 | 0.771 | 0.707 | 0.797 | 0.967 | 0.966 | 0.707 | 0.8495 | 0.327204 |
| BARZ-8R6 | 0.932 | 0.908 | 0.865 | 0.919 | 0.995 | 0.995 | 0.865 | 0.7767 | 0.255955 |
| Odds | 0.757 | 0.66 | 0.617 | 0.675 | 0.934 | 0.93 | 0.617 | 0.7141 | 0.4872 |
| BaRT-1 | 0.594 | 0.588 | 0.473 | 0.608 | 0.853 | 0.853 | 0.473 | 0.8571 | 0.522392 |
| BaRT-4 | 0.541 | 0.552 | 0.445 | 0.556 | 0.737 | 0.744 | 0.445 | 0.7513 | 0.593472 |
| BaRT-7 | 0.479 | 0.478 | 0.375 | 0.476 | 0.586 | 0.566 | 0.375 | 0.6114 | 0.698525 |
| BaRT-10 | 0.404 | 0.367 | 0.365 | 0.414 | 0.466 | 0.463 | 0.365 | 0.4869 | 0.750082 |

TABLE 12: CIFAR-10 untargeted boundary attack results. Note the $\beta$ column refers to the minimum robust accuracy across all boundary attacks.

| | RayS | HSJA | $\beta$ | $p_d$ | $\delta$ |
|---|---|---|---|---|---|
| Vanilla | 0.02 | 0 | 0 | 0.9278 | 0.9278 |
| VanillaT-0.7 | 0.12 | 0 | 0 | 0.9038 | 0.9278 |
| VanillaT-0.95 | 1 | 0 | 0 | 0.8468 | 0.9278 |
| VanillaT-0.99 | 1 | 0 | 0 | 0.7879 | 0.9278 |
| Liu | 0.29 | 0 | 0 | 0.8528 | 0.9278 |
| ADP | 0.05 | 0 | 0 | 0.943 | 0.9278 |
| Xie | 0.85 | 0.84 | 0.84 | 0.7064 | 0.334424 |
| Madry | 0.66 | 0.52 | 0.52 | 0.7524 | 0.536552 |
| Tramer | 0.02 | 0 | 0 | 0.8524 | 0.9278 |
| MulDef-4 | 0.7 | 0.83 | 0.7 | 0.8709 | 0.31817 |
| MulDef-8 | 0.74 | 0.88 | 0.74 | 0.8556 | 0.294656 |
| Guo | 0.01 | 0 | 0 | 0.9092 | 0.9278 |
| BARZ-2 | 0.04 | 0 | 0 | 0.8507 | 0.9278 |
| BARZ-4 | 0.27 | 0.16 | 0.16 | 0.8106 | 0.798104 |
| BARZ-8 | 0.6 | 0.39 | 0.39 | 0.7565 | 0.632765 |
| BARZ-8R2 | 0.76 | 0.99 | 0.76 | 0.8495 | 0.28218 |
| BARZ-8R6 | 0.87 | 0.99 | 0.87 | 0.7767 | 0.252071 |
| Odds | NA | NA | NA | NA | NA |
| BaRT-1 | NA | NA | NA | NA | NA |
| BaRT-4 | NA | NA | NA | NA | NA |
| BaRT-7 | NA | NA | NA | NA | NA |
| BaRT-10 | NA | NA | NA | NA | NA |

against certain defenses. To be clear we DO NOT claim to have the universally strongest black-box attack. We merely show that as different defenses employ different defense techniques, certain black-box attacks will be more effective than others. Thus, it is imperative to test a wide range of black-box attacks (as is done in this paper). From this range of attacks to be tested, the mixed black-box is clearly necessary for validation of a defense.

## APPENDIX. ADVERSARIAL ATTACK DESCRIPTIONS

### A. PURE AND MIXED BLACK-BOX ATTACK

As we mentioned in the main paper, the mixed black-box attack is an extension of the original attack proposed by Papernot [26]. Here we denote $g$ as the synthetic network for the oracle based black-box attack from [26]. The attacker uses an oracle $\mathcal{O}$ which represents black-box access to the target model $f$. The oracle access in this case provides a class label $F(f(x))$ for a query $x$ (and not the score vector $f(x)$). Initially, the attacker has part of the training data set $\mathcal{X}$, i.e., they know $\mathcal{D} = \{(x, F(f(x))) : x \in \mathcal{X}_0\}$ for some $\mathcal{X}_0 \subseteq \mathcal{X}$. Notice that for a single iteration $N = 1$ reduces the attack to an algorithm which does not need any oracle

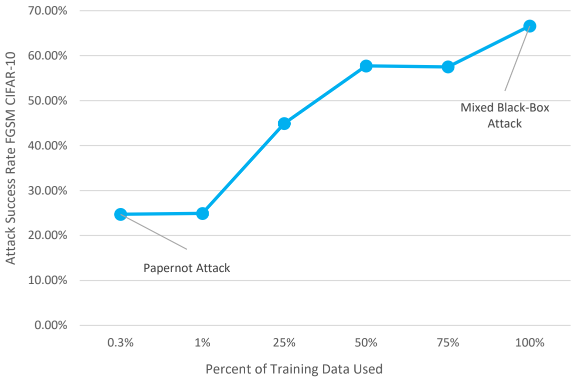


FIGURE 11: FGSM mixed black-box attack success rate as a function of the $\%$ of training data used in the attack.

access to $\mathcal{O}$ build the synthetic model; this reduced algorithm is the one used in the pure black-box attack [10], [33], [48]. In the mixed black-box attack we assume the most capable black-box adversary in Algorithm 1 with access to the entire training data set $\mathcal{X}_0 = \mathcal{X}$ (notice that this excludes the test data used for evaluating the attack success rate).

In order to construct a synthetic network the attacker chooses a-priori a substitute architecture $G$ for which the synthetic model parameters $\theta_g$ need to be trained. The attacker uses known image-label pairs in $\mathcal{D}$ to train $\theta_g$ using a training method $M$ (e.g., Adam [49]). In each iteration the known data is doubled using the following data augmentation technique: For each image $x$ in the current data set $\mathcal{D}$, black-box access to the target model gives label $l = \mathcal{O}(x)$. The Jacobian of the synthetic network score vector $g$ with respect to its parameters $\theta_g$ is evaluated/computed for image $x$. The signs of the column in the Jacobian matrix that correspond to class label $l$ are multiplied with a (small) constant $\lambda$ – this constitutes a vector which is added to $x$. This gives one new image for each $x$ and this leads to a doubling of $\mathcal{D}$. After $N$ iterations the algorithm outputs the trained parameters $\theta_g$ for the final augmented data set $\mathcal{D}$.

### B. ADVERSARIAL SAMPLE GENERATION

After the synthetic model is trained, adversarial samples need to be created from the synthetic model to attack the defense. Hence any white-box attack can be run on the synthetic model to create an adversarial example. The adversary can then check if this example fools the defense. To reiterate, in this paper we focus on a black-box adversary so running white-box attacks *directly* on any defense is not within the scope of our adversarial model. We briefly introduce the following commonly used white-box attacks that we use for adversarial sample generation:

**Fast Gradient Sign Method (FGSM) – [8].** Computes $x' = x' + \epsilon \times sign(\nabla_x L(x, l; \theta)$ where $L$ is a loss function (e.g, cross entropy) of model $f$.

**Basic Iterative Methods (BIM) – [38].** $x'_i = \text{clip}_{x,\epsilon}(x'_{i-1} + \frac{\epsilon}{r} \times sign(\nabla_{x'_{i-1}} L(x'_{i-1}, l; \theta))$ where $x'_0 = x$, $r$ is the number of iterations, clip is a clipping operation.

**Momentum Iterative Methods (MIM) – [39].** This is a variant of BIM using momentum trick to create the gradient $g_i$, i.e., $x'_i = \text{clip}_{x,\epsilon}(x'_{i-1} + \frac{\epsilon}{r} \times sign(g_i))$.

**Projected Gradient Descent (PGD) – [27].** This is also a variant of BIM where the clipping operation is replaced by a projection operation.

**Carlini and Wagner attack (C&W) – [11].** We define $x'(\omega) = \frac{1}{2}(\tanh \omega + 1)$ and $g(x) = \max(\max(s_i : i \neq l) - s_i, -\kappa)$ where $f(x) = (s_1, s_2, \ldots)$ is the score vector of input $x$ of classifier $f$ and $\kappa$ controls the confidence on the adversarial examples. The adversary builds the following objective function for finding the adversarial noise.

$$\min_{\omega} \|x'(\omega) - x\|_2^2 + cf(x'(\omega)),$$

where $c$ is a constant chosen by a modified binary search.

**Elastic Net Attack (EAD) – [40].** This is the variant of C&W attack with the following objective function.

$$\min_{\omega} \|x'(\omega) - x\|_2^2 + \beta\|x'(\omega) - x\|_1 + cf(x'(\omega)).$$

## APPENDIX. EXPERIMENTAL IMPLEMENTATION DETAILS AND MISC.

### A. IMPLEMENTATION OF BARZ

In the BARZ, we use image transformations that are composed of a resizing operation $i(x)$ and a linear transformation $c(x) = Ax + b$. In a CNN implementation one can think of $i(c(x))$ as an extra layer in the CNN architecture itself. We refer to this extra layer as the protected layer. An input image $x$ at a protected layer in BARZ is linearly transformed into an image $i(c(x))$ before it enters the corresponding CNN network.

For the resize operations $i(\cdot)$ used in each of the protected layers in BARZ, we choose sizes that are larger than the original dimensions of the image data. We do this to prevent loss of information in the images that downsizing would create (and this would hurt the clean accuracy of BARZ). In our experiments we use BARZ with 2, 4, and 8 protected layers. Each protected layer gets its own resize operation $i(\cdot)$. When using 8 protected layers, we use image resizing operations from 32 to 32, 40, 48, 64, 72, 80, 96, 104. Each protected layer will be differentiated from each other protected layer due to the difference in how much resizing each layer implements. This will lead to less transferability between the protected layers and as a result we expect to see a wider barrier zone which diminishes the attacker's success rate. When using 4 protected layers, we use a copy of the 4 protected layers from BARZ with 8 networks that correspond to the image resizing operations from 32 to 32, 48, 72, 96. When using 2 protected layers, we use a copy of the 2 protected layers from BARZ with 8 networks that correspond to the image resizing operations from 32 to 32 and 104.

For each protected layer, the linear transformation $c(x) = Ax + b$ is randomly chosen from some statistical distribution (the distribution is public knowledge and therefore known by the adversary). Design of the statistical distribution depends on the complexity of the considered data set (in our case we experiment with Fashion-MNIST and CIFAR-10).For CIFAR-10 we take matrices $A_i$ to be identity matrices (this also makes $A$ the identity matrix in the vector representation of $c(x)$) and we use the same matrix $b$ for each of the matrices $b_i$, i.e.,

$$b' = b_1 = b_2 = b_3.$$

This means that we use the same random offset in the red, blue, and green values of a pixel. The reason for making this design decision is because for CIFAR-10 we found that fully random $A$ creates large drops in clean accuracy, even when the network is trained to learn such distortions. As a result, for data sets with high spatial complexity like CIFAR-10, we do not select $A$ randomly. We choose $A$ to be the identity matrix. Likewise for $b'$ we only randomly generate $35\%$ of the matrix values and leave the rest as $0$. For the randomly generated values, we choose them from a uniform distribution from $-0.5$ to $0.5$.

For datasets with less spatial complexity like Fashion-MNIST, we equate matrices $A' = A_1 = A_2 = A_3$ and $b' = b_1 = b_2 = b_3$ and select $A'$ and $b'$ as random matrices: The values of $A'$ and $b'$ are selected from a Gaussian distribution with $\mu = 0$ and $\sigma = 0.1$.

### B. ATTACK AND DEFENSE PARAMETERS

In order to implement a black-box attack we first run Algorithm 1 which trains a synthetic network $g$. Next, out of the test data (each dataset has 10,000 samples in our setup) we select the first 1000 samples correctly identified by the defense. For each of the 1000 samples we run a certain white-box attack to produce 1000 adversarial examples. The attacker's success rate is the fraction of adversarial examples which change $l$ to the desired new randomly selected $l'$ in a targeted attack or any other label $l' \neq \perp$ for an untargeted attack.

The parameters for the adversarial generation techniques( white-box attacks) used in conjunction with our synthetic model for both the mixed black-box attack and pure black-box attack can be found in table 13. For all attacks we use the $\|l\|_\infty$ norm except for the Carlini and Wagner attack. For the Carlini and Wagner attack only the $\|l\|_2$ implementation (given by the authors) has a run time efficient enough for our current hardware setup (to test on 10 defenses and 2 datasets). Future work may include trying mixed black-box attack with the $\|l\|_\infty$ if efficient implementations of the Carlini and Wagner attack become available in the future.

TABLE 13: Attacks' parameters. $i$ - number of iterations, $d$ - decaying factor, $r$ radius of the ball for generating the initial noise, $c$ - constant value of C&W attack, $\epsilon$ - noise magnitude, $\beta$ - constant value of EAD attack. Binary Search = Bi.Sr

| Attacks | Fashion-MNIST | CIFAR-10 |
|---|---|---|
| FGSM | $\epsilon = 0.1$ | $\epsilon = 0.05$ |
| BIM | $i = 10, \epsilon = 0.01$ | $i = 10, \epsilon = 0.005$ |
| PGD | $i = 10, r = 0.031, \epsilon = 0.01$ | $i = 10, r = 0.031, \epsilon = 0.005$ |
| MIM | $i = 10, d = 1.0, \epsilon = 0.01$ | $i = 10, d = 1.0, \epsilon = 0.005$ |
| C&W | $i = 1000, c = $ Bi.Sr | $i = 1000, c = $ Bi.Sr |
| EAD | $i = 1000, c = $ Bi.Sr, $\beta = 0.01$ | $i = 1000, c = $ Bi.Sr, $\beta = 0.01$ |

The precise set-up for our experiments is given in Tables 14, 15, and 16. Table 14 details the training method $T$ in Algorithm 1. For the evaluated data sets Fashion-MNIST and CIFAR-10 without data augmentation, we enumerate in Table 15 the amount $|\mathcal{X}_0|$ of training data together with parameters $\lambda$ and $N$ ($\lambda = 0.1$ and $N = 6$ are taken from the oracle based black-box attack paper of [26]; notice that a test data set of size 10,000 is standard practice; all remaining data serves training and this is entirely accessible by the attacker).

Table 16 depicts the architecture $G$ of the CNN network of the synthetic network $g$ for the different data sets; the structure has several layers (not to be confused with 'protection layer' in BARZ which is an image transformation together with a whole CNN in itself). The adversary attempts to attack BARZ and will first learn a synthetic network $g$ with architecture $G$ that corresponds to Table 16. Notice that the image transformations are kept secret and for this reason the attacker can at best train a synthetic vanilla network. Of course the attacker does know the set from which the image transformations in BARZ are taken and can potentially try to learn a synthetic CNN for each possible image transformation and do some majority vote (like BARZ) on the outputted labels generated by these CNNs. However, there are exponentially many transformations making such an attack infeasible.

### C. BOUNDARY ATTACK COMPUTATIONAL COMPLEXITY AND TARGETED BOUNDARY ATTACKS

In the main body of the paper we mention that both the Odds are Odd (Odds) and Barrage of random transforms (BaRT) are not applicable for boundary attacks. For pure and mixed

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2021.3138966, IEEE Access

TABLE 14: Training parameters used in the experiments

| Training Parameter | Value |
|---|---|
| Optimization Method | ADAM |
| Learning Rate | 0.0001 |
| Batch Size | 64 |
| Epochs | 100 |
| Data Augmentation | None |

TABLE 15: Mixed black-box attack parameters

| | $\lvert\mathcal{X}_0\rvert$ | $N$ | $\lambda$ |
|---|---|---|---|
| CIFAR-10 | 50000 | 4 | 0.1 |
| Fashion-MNIST | 60000 | 4 | 0.1 |

TABLE 16: Architectures of synthetic neural networks $g$ from [11]

| Layer Type | Fashion-MNIST and CIFAR-10 |
|---|---|
| Convolution + ReLU | $3 \times 3 \times 64$ |
| Convolution + ReLU | $3 \times 3 \times 64$ |
| Max Pooling | $2 \times 2$ |
| Convolution + ReLU | $3 \times 3 \times 128$ |
| Convolution + ReLU | $3 \times 3 \times 128$ |
| Max Pooling | $2 \times 2$ |
| Fully Connected + ReLU | 256 |
| Fully Connected + ReLU | 256 |
| Softmax | 10 |

black-box attacks we can efficiently parallelize the evaluation of many samples using either the GPU or multiple CPUs (in the case of image transformations). However, the boundary attacks require large number of evaluations done sequentially (e.g. 10,000 queries) so we cannot take advantage of the previously mentioned parallelism. This causes the run time of boundary attacks for these defenses with our standard implementation to be on the order of weeks. These attacks are not applicable for our current setup (28 core CPU machine and 2 Titan V GPUs).

It is also worth noting in this paper we do not directly consider targeted boundary attacks. Although we do provide experimental details for some other black-box target attacks, in this paper our main focus is on the untargeted attack. As we already have 12 targeted attacks presented in this paper (6 mixed black-box and 6 pure black-box types) we leave the targeted boundary attack as potential future work.

### D. FUTURE WORK

There are several promising directions for possible future work. From a security perspective, our paper has demonstrated the effectiveness of image transformations for black-box robustness. We experimented with a set of image transformations that we found to be effective in creating barrier zones. However, large scale studies on the transferability of single and fixed combinational image transformations has not yet been done, to the best of our knowledge. Determining exactly which image transformations are capable of distorting adversarial noise while maintain robustness would bring the field much closer to establishing a set of image transformations as security primitives.

On the machine learning side, enhancement to the clean accuracy of the BARZ defense may be possible through the introduction of novel architectures. Specifically, the Big Transfer Models [50] are a class of CNNs that have shown remarkable performance on datasets like CIFAR-10 and CIFAR-100. Using these new architectures could be one possible way to improve the clean accuracy of the BARZ defense.

On the attacker side in this work, we only consider an adversary that is interested in misclassification (either targeted or untargeted). The attacker starts with a clean example and specifically tries to avoid having the sample marked with the correct label or marked with the adversarial label. To the best of our knowledge, work has not been extensively done on what might be considered the inverse of this problem i.e., the attacker tries to overwhelm the system with legitimate examples that are marked as adversarial. While an interesting problem in its own right, this is beyond the scope of our current work. It may be a problem future defense designers would want to take into account and try to mitigate.

Lastly from the attacker side, optimizations can still be made to the adaptive black-box attack. In our paper, we found one simple CNN architecture (through experimentation) that was both simple to train and yielded highly transferable adversarial examples. However, it may still be possible to optimize the architecture in the attack, to potentially increase the attack success rate. In addition, as white-box attacks continue to improve, it may be possible to substitute the MIM adversarial generation method in the adaptive black-box attack with an even stronger technique.