
Buffer Zone based Defense against Adversarial Examples in Image Classification

Supplementary Material

A WHITE-BOX ATTACKS

As explained and motivated in the introduction, we restrict ourselves to the black-box setting where the parameters of our defense are kept secret. Hence, this disallows direct white-box attacks and zeroth order optimization based black-box attacks. However, it is important to note that once a synthetic model has been trained, 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.

Essentially any white-box attack can be run on the synthetic model to try to exploit the transferability between classifiers (Papernot et al., 2016b). We briefly introduce the following commonly used white-box attacks in the literature.

We briefly introduce the following commonly used white-box attacks in literature.

Fast Gradient Sign Method (FGSM) – (Goodfellow et al., 2014). $x' = x + \epsilon \times \text{sign}(\nabla_x L(x, l; \theta))$ where L is a loss function (e.g, cross entropy) of model f .

Basic Iterative Methods (BIM) – (Kurakin et al., 2017). $x'_i = \text{clip}_{x, \epsilon}(x'_{i-1} + \frac{\epsilon}{r} \times \text{sign}(\nabla_{x'_{i-1}} L(x'_{i-1}, l; \theta)))$ where $x'_0 = x$, r is the number of iterations, clip is a clipping operation.

Momentum Iterative Methods (MIM) – (Dong et al., 2018). This is a variant of BIM using momentum trick to create the gradient g_i , i.e., $x'_i = \text{clip}_{x, \epsilon}(x'_{i-1} + \frac{\epsilon}{r} \times \text{sign}(g_i))$.

Projected Gradient Descent (PGD) – (Madry et al., 2018). This is also a variant of BIM where the clipping operation is replaced by a projection operation.

Carlini and Wagner attack (C&W) – (Carlini & Wagner, 2017a). We define $x'(\omega) = \frac{1}{2}(\tanh \omega + 1)$ and $g(x) = \max(\max(s_i : i \neq l) - s_i, -\kappa)$ where $f(x) = (s_1, s_2, \dots)$ is the score vector of input x of classifier f and κ 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) – (Chen et al., 2018). This is the variant of C&W attack with the following objective function.

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

B DEFENSES

B.1 BARRAGE OF RANDOM TRANSFORMS (BART) – (RAFF ET AL., 2019)

Barrage of Random Transforms (BaRT) by (Raff et al., 2019) is a defense based on applying image transformations before classification. The defense works by randomly selecting a set of transformations and a random order in which the image transformations are applied. In addition, the

parameters for each transformation are also randomly selected at run time to further enhance the entropy of the defense. Broadly speaking, there are 10 different image transformations groups: JPEG compression, image swirling, noise injection, Fourier transform perturbations, zooming, color space changes, histogram equalization, grayscale transformations and denoising operations.

B.2 THE ODDS ARE ODD (ODDS) – (ROTH ET AL., 2019)

The Odds are Odd introduced in (Roth et al., 2019) is a defense based on a statistical test. This test is motivated by the following observation: the behaviors of benign and adversarial examples are different at the logits layer (i.e. the input to the softmax layer). The test works as follows: For a given input image, multiple copies are created and a random noise is added to each copy. This creates multiple random noisy images. The defense calculates the logits values of each noisy image and use them as the input for the statistical test.

B.3 IMPROVING ADVERSARIAL ROBUSTNESS VIA PROMOTING ENSEMBLE DIVERSITY (ADP) – (PANG ET AL., 2019)

Constructing ensembles of enhanced networks is one defense strategy to improve the adversarial robustness of classifiers. However, in an ensemble model, the lack of interaction among individual members may cause them to return similar predictions. This defense proposes a new notion of ensemble diversity by promoting the diversity among the predictions returned by members of an ensemble model using an adaptive diversity promoting (ADP) regularizer, which works with a logarithm of ensemble diversity term and an ensemble entropy term, see (Pang et al., 2019). The ADP regularizer helps non-maximal predictions of each ensemble member to be mutually orthogonal, while the maximal prediction is still consistent with the correct label. This defense employs a different training procedure where the ADP regularizer is used as the penalty term and the ensemble network is trained interactively.

B.4 MADRY’S ADVERSARIAL TRAINING (MADRY) – (MADRY ET AL., 2018)

(Madry et al., 2018) proposed a new approach to build a robust defense based on an adversarial training process. The training has many iterations and in each iteration there are two phases: (1) The attack phase, where for a given dataset and a classifier the designer uses some adversarial attacks (i.e., white-box attacks) to derive an adversarial dataset. (2) In the defense phase a new training dataset is constructed by combining the adversarial dataset together with the set of class labels of the original dataset. Next, this is used to train the current network.

The authors showed that this training approach produces a high robust defense against adversarial machine learning with respect to white-box and black-box attacks.

B.5 MULTI-MODEL-BASED DEFENSE (MUL-DEF) – (SRISAKAOKUL ET AL., 2018)

In unpublished work, (Srisakaokul et al., 2018) have proposed a defense against white-box attacks based on multiple networks with the *same* architecture. The authors develop their defense based on a retraining technique. First, the authors apply adversarial attacks on each network to generate a set of adversarial examples. For example, for each network j a white-box attack produces a set of adversarial examples \mathcal{S}_j . Next network j will be retrained with the clean training data set together with some of the adversarial sets \mathcal{S}_h , $h \neq j$. The authors argue that all the networks cannot be fooled at the same time for a given adversarial example and this leads to a low(er) attacker’s success rate. The final outputted class label is the predicted label of one of the networks chosen at random among all networks; this gives high clean accuracy.

B.6 COUNTERING ADVERSARIAL IMAGES USING INPUT TRANSFORMATIONS (GUO) – (GUO ET AL., 2017)

In (Guo et al., 2017), the designer selects a set of possible image transformations for a single network and keeps the selection of the chosen image transformation secret. The image transformation will distort the noise as explained in (Guo et al., 2017). This is BUZZ for a single protection layer (without multiple networks and threshold voting).

B.7 ENSEMBLE ADVERSARIAL TRAINING: ATTACKS AND DEFENSES (TRAMER) – (TRAMÈR ET AL., 2017)

(Tramèr et al., 2017) proposes another type of adversarial training method. The adversarial examples are generated by doing attacks on different networks with different attack methods. After this the designer trains the new network with the generated adversarial examples. The authors 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). In other words, the network is supposed to have better robustness against black-box attack generalization across models. As shown in (Athalye et al., 2018b), the adversarially trained network is vulnerable to white-box attack.

B.8 MIXED ARCHITECTURE – (LIU ET AL., 2017)

In (Liu et al., 2017), the authors study the transferability between different networks which have different structures for the ImageNet dataset. The authors report that the transferability between the networks is small (claimed to be 'close to zero'). For this reason, it may be possible to have a low attack success rate for the BUZZ defense where protected networks have different architectures.

B.9 MITIGATING ADVERSARIAL EFFECTS THROUGH RANDOMIZATION (XIE) – (XIE ET AL., 2018)

(Xie et al., 2018) has a single network and uniformly selects an image transformation from an a-priori fixed set of a small number of image transformations to defeat white-box attacks. In the white-box setting (Athalye et al., 2018a) shows that this defense does not work. An open question is whether this defense is secure against black-box attacks.

B.10 THRESHOLDING NETWORKS (CONCEPT DEVELOPED IN THIS PAPER)

This technique is proposed in this paper and is applied to a single classifier. The idea of the technique is simple but it helps us to create buffer zones between the decision regions. For a given input x , if the highest probability score in the score vector is smaller than a threshold T , then we output NULL class label (\perp)

C PSEUDO ALGORITHMS: BLACK-BOX ATTACK & BUZZ

Synthetic network. Algorithm 1 depicts the construction of a synthetic network g for the oracle based black-box attack from (Papernot et al., 2017). The attacker uses as input an oracle \mathcal{O} which represents black-box access to the target model f which only returns the final 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., he knows $\mathcal{D} = \{(x, F(f(x))) : x \in \mathcal{X}_0\}$ for some $\mathcal{X}_0 \subseteq \mathcal{X}$. Notice that for a single iteration $N = 1$, Algorithm 1 therefore reduces to an algorithm which does not need any oracle access to \mathcal{O} ; this reduced algorithm is the one used in the pure black-box attack (Carlini & Wagner, 2017b; Athalye et al., 2018a; Liu et al., 2017). In this paper we assume the strongest black-box adversary in Algorithm 1 with access to the entire training data set $\mathcal{X}_0 = \mathcal{X}$ (notice that this excludes test data 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 θ_g need to be trained. The attacker uses known image-label pairs in \mathcal{D} to train θ_g using a training method M (e.g., Adam (Kingma & Ba, 2014)). 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 θ_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 λ – 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 θ_g for the final augmented data set \mathcal{D} .

Algorithm 1 Construction of synthetic network g in Papernot’s oracle based black-box attack

-
- 1: **Input:**
 - 2: \mathcal{O} represents black-box access to $F(f(\cdot))$ for target model f with output function F ;
 - 3: $\mathcal{X}_0 \subseteq \mathcal{X}$, where \mathcal{X} is the training data set of target model f ;
 - 4: substitute architecture G
 - 5: training method M ;
 - 6: constant λ ;
 - 7: number N of synthetic training epochs
 - 8: **Output:**
 - 9: synthetic model g defined by parameters θ_g
 - 10: (g also has output function F which selects the max confidence score;
 - 11: g fits architecture G)
 - 12:
 - 13: **Algorithm:** For N iterations
 - 14: $\mathcal{D} \leftarrow \{(x, \mathcal{O}(x)) : x \in \mathcal{X}_t\}$
 - 15: $\theta_g = M(G, \mathcal{D})$
 - 16: $\mathcal{X}_{t+1} \leftarrow \{x + \lambda \cdot \text{sgn}(J_{\theta_g}(x)[\mathcal{O}(x)]) : x \in \mathcal{X}_t\} \cup \mathcal{X}_t$
 - 17: Output θ_g
-

The precise set-up for our experiments is given in Tables 2, 3, and 4. Table 2 details the used training method M in Algorithm 1. For the evaluated data sets Fashion-MNIST and CIFAR-10 without data augmentation, we enumerate in Table 3 the amount $|\mathcal{X}_0|$ of training data together with parameters λ and N in Algorithm 1 ($\lambda = 0.1$ and $N = 6$ are taken from the oracle based black-box attack paper of (Papernot et al., 2017)); 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 4 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 BUZZ which is an image transformation together with a whole CNN in itself). The adversary attempts to attack BUZZ and will first learn a synthetic network g with architecture G (used as input in Algorithm 1 that corresponds to Table 4. 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 BUZZ are taken and can potentially try to learn a synthetic CNN for each possible image transformation and do some majority vote (like BUZZ) on the outputted labels generated by these CNNs. However, there are exponentially many transformations making such an attack infeasible. For future research we will investigate whether a small sized subset of ‘representative’ image transformations can be used to generate a synthetic model which can be used to attack BUZZ in a more effective way. Nevertheless, we believe that BUZZ will remain secure because of the security argument given in Section 3.2 where is shown how a single perturbation η leads to very different perturbations at each protected layer in BUZZ. This leads to ‘wide’ buffer zones and their mere existence is enough to achieve our security goal – security is not derived from keeping the image transformations private. Keeping these transformations private just makes it harder for the adversary to construct a more effective attack but the resulting attack is expected to still have small attacker’s success rates. We leave this study for future work.

Table 2: Training parameters used in the experiments

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

White-box attack on the synthetic network. We perform the white-box attacks as described in Appendix A such as FGSM (Goodfellow et al., 2015), BIM (Kurakin et al., 2017), MIM (Dong et al.,

Table 3: Mixed black-box attack parameters

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

Table 4: Architectures of synthetic neural networks g from Carlini & Wagner (2017a)

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

2018), PGD (Madry et al., 2018), Carlini&Wagner(Carlini & Wagner, 2017a) and EAD (Chen et al., 2018) attacks on synthetic model in the mixed black box attacks. Appendix A.

When a certain white-box attack is used as a pure black-box attack, then no oracle access is available and comparison $l' = \mathcal{O}(x)$ is replaced by comparison $l' = F(g(x))$, which uses the synthetic network.

The parameters of the white-box attacks used in our paper can be found in the following table 5.

Table 5: 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, ϵ - noise magnitude, β - constant value of EAD attack. Binary Search = Bi.Sr

| Attacks | Fashion-MNIST | CIFAR-10 |
|---------|--|--|
| FGSM | $\epsilon = 0.15$ | $\epsilon = 0.05$ |
| BIM | $i = 10, \epsilon = 0.015$ | $i = 10, \epsilon = 0.005$ |
| PGD | $i = 10, r = 0.031, \epsilon = 0.015$ | $i = 10, r = 0.031, \epsilon = 0.005$ |
| MIM | $i = 10, d = 1.0, \epsilon = 0.015$ | $i = 10, d = 1.0, \epsilon = 0.005$ |
| C&W | $i = 1000, c = \text{Bi.Sr}$ | $i = 1000, c = \text{Bi.Sr}$ |
| EAD | $i = 1000, c = \text{Bi.Sr}, \beta = 0.01$ | $i = 1000, c = \text{Bi.Sr}, \beta = 0.01$ |

Success rate black-box attack. In order to implement the black-box attack we first run Algorithm 1 which outputs the parameters of a synthetic network g . Next, out of the test data (each data set has 10.000 samples in our set-up) we select the first 1000 samples (x, l) which the target model f (i.e., BUZZ in this paper) correctly classifies. 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.

Image transformations for BUZZ. In the BUZZ, we use image transformations that are composed of a resizing operation $i(x)$ and a linear transformation $c(x) = Ax + b$. An input image x at a protected layer in BUZZ is linearly transformed into an image $i(c(x))$ before it enters the corresponding CNN network with ResNet architecture for CIFAR-10 and for Fashion-MNIST. In a network implementation one can think of $i(c(x))$ as an extra layer in the CNN architecture of ResNet itself.

For the resize operations $i(\cdot)$ used in each of the protected layers in BUZZ, 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 down sizing would create (and this would hurt the clean accuracy of BUZZ). In our experiments we use BUZZ 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 buffer zone which diminishes the attacker’s success rate. When using 4 protected layers, we use a copy of the 4 protected layers from BUZZ 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 BUZZ with 8 networks that correspond to the image resizing operations from 32 to 32 and 104. In our implementation we use resizing operation from github https://github.com/cihangxie/NIPS2017_adv_challenge_defense (Xie et al., 2018).

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). Transformation $c(x)$ takes an image of size $n_1 \times n_2 \times 3$ as input and considers this as a vector of length $k = n_1 n_2 n_3$. Here, n_1 and n_2 denote the horizontal and vertical width in pixels of image x ; $n_3 = 3$ means that each pixel has a red, blue, and green values; $n_3 = 1$ means that each pixel only has one black/white value. CIFAR-10 has $32 \times 32 \times 3$ images and Fashion-MNIST has $28 \times 28 \times 1$ images. All the values in vector x are converted from integers $[0..255]$ to the range $[-0.5, +0.5]$ of real numbers. Notice that the entries of $c(x)$ may have their values outside of this range.

In our implementation we do not consider x to be in vector representation; we think of x as n_3 times a $n_1 \times n_2$ matrix. For example, $x = (X_1, X_2, X_3)$ for $n_3 = 1$. We restrict $c(x) = Ax + b$ to linear operations

$$c(X_1, X_2, X_3) = (X_1 A_1 + b_1, X_2 A_2 + b_2, X_3 A_3 + b_3),$$

where A_i are $n_2 \times n_2$ matrices and b_i are $n_1 \times n_2$ matrices.

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 $\sigma = 0.1$.

D EXPERIMENTAL RESULTS

In this section, we present our experimental results, i.e., the **mixed targeted and untargeted black box attacks, pure targeted and untargeted black box attacks, and boundary attacks** – untargeted HopSkipJump (Chen et al., 2020) and RayS (Chen & Gu, 2020) on **eleven different defenses strategies**, i.e., Barrage of Random Transforms (BaRT) (Raff et al., 2019), The Odds are Odd (Odds) (Roth et al., 2019), Ensemble Diversity (ADP) (Pang et al., 2019), Madry’s Adversarial Training (Madry) (Madry et al., 2018), Multi-model-based Defense (Mul-Def) (Srisakaokul et al., 2018), Countering Adversarial Images using Input Transformations (Guo) (Guo et al., 2017), Ensemble Adversarial Training: Attacks and Defenses (Tramer) (Tramèr et al., 2017), Mixed Architecture (Liu et al., 2017), Mitigating adversarial effects through randomization (Xie) (Xie et al., 2018), Thresholding Networks (a basic proof of concept defense developed in this paper) and Buffer Zones (BUZZ) with CIFAR-10 (Krizhevsky et al.) and Fashion-MNIST (Xiao et al., 2017) datasets. The **six white-box attacks** on the synthetic models are FGSM (Goodfellow et al., 2014), BIM (Kurakin et al.,

2017), MIM (Dong et al., 2018), PGD (Madry et al., 2018), C&W (Carlini & Wagner, 2017a) and EAD (Chen et al., 2018).

The last subsection explains our experiments that demonstrate the existence of buffer zones.

D.1 FASHION-MNIST

The results for Fashion-MNIST are described in Tables 6, 7 and 8. We recall the formula for the δ metric:

$$\delta = p - (p - \gamma)(1 - \alpha) = p - p_d \cdot \beta, \quad (3)$$

where p is the clean accuracy of the vanilla classifier (i.e., no defense at all and without any adversarial presence), γ 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), α is the attacker’s success rate against the defense and β is the defense success rate (also called defense accuracy) and is equal to $1 - \alpha$.

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

For Fashion-MNIST and CIFAR-10, $p = 0.94$ and 0.93 , respectively. The value of δ 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 α for the specific set of attacks considered, similarly, this corresponds to the minimum over the various defense success rates β). For example, the δ metric for BUZZ-8 in Table 6 is computed as follows: we substitute $p = 0.94$, $p_d = 0.78$, and the minimal $\beta = 0.91$ among all (currently known) mixed black-box attacks (in this case corresponding to the FGSM-U attack) into formula (Eq. 3) for δ . This results in $\delta = 0.23$.

Tables 6, 7 and 8 enumerate δ for mixed black-box attacks, pure black-box attacks, and boundary attacks. As noted above, for mixed black-box attacks Table 6 shows $\delta = 0.23$ for BUZZ-8. Similarly, Tables 7 and 8 show $\delta = 0.24$ for pure black-box attacks and $\delta = 0.24$ for boundary attacks. This means that across the three classes of black-box attacks BUZZ-8 achieves $\delta = 0.24$.

As another example, Madry achieves $\delta = 0.54$, $\delta = 0.41$, and $\delta = 0.16$, respectively. This shows that Madry defends well (the best among all defenses) against boundary attacks but does not perform well against mixed and pure black-box attacks. Across the three classes of black-box attacks Madry only scores $\delta = 0.54$ while BUZZ-8 achieves $\delta = 0.24$.

Table 6: Targeted (T) and Untargeted (U) mixed black-box attacks on different defenses for Fashion-MNIST. Minimum defense efficiency - MIN β , Clean prediction accuracy p_d , Drop in clean prediction accuracy γ .

| | FGSM-T | IFGSM-T | MIM-T | PGD-T | CW-T | EAD-T | FGSM-U | IFGSM-U | MIM-U | PGD-U | CW-U | EAD-U | MIN β | p_d | γ | δ |
|------------------|--------|---------|-------|-------|------|-------|--------|---------|-------|-------|------|-------|-------------|-------|----------|----------|
| Vanilla | 0.71 | 0.53 | 0.46 | 0.53 | 0.99 | 0.99 | 0.23 | 0.12 | 0.11 | 0.12 | 0.96 | 0.94 | 0.11 | 0.94 | 0.00 | 0.83 |
| Guo (BUZZ-1) | 0.82 | 0.92 | 0.88 | 0.91 | 1.00 | 1.00 | 0.38 | 0.55 | 0.50 | 0.56 | 0.99 | 0.98 | 0.38 | 0.90 | 0.03 | 0.60 |
| BUZZ-2 | 0.93 | 0.96 | 0.95 | 0.96 | 1.00 | 1.00 | 0.70 | 0.78 | 0.73 | 0.77 | 1.00 | 1.00 | 0.70 | 0.85 | 0.08 | 0.34 |
| BUZZ-4 | 0.97 | 0.99 | 0.97 | 0.99 | 1.00 | 1.00 | 0.82 | 0.85 | 0.82 | 0.85 | 1.00 | 1.00 | 0.82 | 0.82 | 0.12 | 0.27 |
| BUZZ-8 | 0.99 | 1.00 | 1.00 | 1.00 | 1.00 | 1.00 | 0.91 | 0.94 | 0.92 | 0.95 | 1.00 | 1.00 | 0.91 | 0.78 | 0.16 | 0.23 |
| Liu (Mixed Arch) | 0.91 | 0.90 | 0.89 | 0.89 | 1.00 | 1.00 | 0.68 | 0.56 | 0.61 | 0.55 | 0.99 | 0.98 | 0.55 | 0.90 | 0.04 | 0.44 |
| ADP | 0.79 | 0.52 | 0.50 | 0.52 | 0.99 | 0.99 | 0.14 | 0.09 | 0.10 | 0.09 | 0.93 | 0.91 | 0.09 | 0.95 | -0.01 | 0.85 |
| VanillaT-0.7 | 0.76 | 0.60 | 0.52 | 0.59 | 0.99 | 0.99 | 0.35 | 0.17 | 0.18 | 0.18 | 0.97 | 0.95 | 0.17 | 0.92 | 0.01 | 0.78 |
| VanillaT-0.95 | 0.85 | 0.77 | 0.72 | 0.79 | 1.00 | 1.00 | 0.57 | 0.31 | 0.34 | 0.30 | 0.99 | 0.99 | 0.30 | 0.89 | 0.04 | 0.66 |
| VanillaT-0.99 | 0.94 | 0.89 | 0.86 | 0.89 | 1.00 | 1.00 | 0.72 | 0.50 | 0.52 | 0.50 | 0.99 | 1.00 | 0.50 | 0.84 | 0.09 | 0.51 |
| Xie | 0.71 | 0.62 | 0.58 | 0.63 | 0.96 | 0.95 | 0.21 | 0.20 | 0.19 | 0.20 | 0.79 | 0.77 | 0.19 | 0.82 | 0.12 | 0.78 |
| Madry | 0.96 | 1.00 | 1.00 | 1.00 | 0.98 | 0.96 | 0.49 | 0.96 | 0.95 | 0.96 | 0.92 | 0.87 | 0.49 | 0.81 | 0.13 | 0.54 |
| Tramer | 0.81 | 0.83 | 0.76 | 0.83 | 0.99 | 0.99 | 0.34 | 0.38 | 0.35 | 0.40 | 0.98 | 0.97 | 0.34 | 0.94 | 0.00 | 0.61 |
| MulDef-4 | 0.83 | 0.80 | 0.76 | 0.81 | 0.99 | 0.98 | 0.35 | 0.42 | 0.36 | 0.40 | 0.95 | 0.94 | 0.35 | 0.94 | 0.00 | 0.61 |
| MulDef-8 | 0.84 | 0.85 | 0.82 | 0.84 | 0.99 | 0.99 | 0.38 | 0.47 | 0.41 | 0.47 | 0.95 | 0.95 | 0.38 | 0.94 | 0.00 | 0.58 |
| BaRT-1 | 0.84 | 0.81 | 0.76 | 0.81 | 0.98 | 0.98 | 0.36 | 0.38 | 0.34 | 0.38 | 0.86 | 0.88 | 0.34 | 0.90 | 0.03 | 0.63 |
| BaRT-4 | 0.87 | 0.85 | 0.84 | 0.85 | 0.94 | 0.96 | 0.41 | 0.40 | 0.38 | 0.41 | 0.78 | 0.79 | 0.38 | 0.83 | 0.11 | 0.62 |
| BaRT-6 | 0.88 | 0.88 | 0.86 | 0.88 | 0.94 | 0.95 | 0.37 | 0.44 | 0.42 | 0.41 | 0.72 | 0.73 | 0.37 | 0.78 | 0.15 | 0.65 |
| BaRT-8 | 0.87 | 0.87 | 0.85 | 0.86 | 0.94 | 0.96 | 0.40 | 0.41 | 0.37 | 0.39 | 0.71 | 0.70 | 0.37 | 0.71 | 0.22 | 0.67 |
| Odds | 0.67 | 0.54 | 0.47 | 0.55 | 0.22 | 0.15 | 0.14 | 0.15 | 0.99 | 0.94 | 0.99 | 0.94 | 0.14 | 0.75 | 0.18 | 0.83 |

Discussion. We have the following observations from Tables 6, 7 and 8:

1. The BUZZ family achieves the smallest δ for mixed black box attacks, the smallest δ for pure black box attacks and the second smallest δ for boundary attacks. The BUZZ family achieves

Table 7: Targeted (T) and Untargeted (U) Pure black-box attacks on different defenses for Fashion-MNIST. Minimum defense efficiency - $\text{MIN } \beta$, Clean prediction accuracy p_d , Drop in clean prediction accuracy γ .

| | FGSM-T | IFGSM-T | MIM-T | PGD-T | CW-T | EAD-T | FGSM-U | IFGSM-U | MIM-U | PGD-U | CW-U | EAD-U | $\text{MIN } \beta$ | p_d | γ | δ |
|------------------|--------|---------|-------|-------|------|-------|--------|---------|-------|-------|------|-------|---------------------|-------|----------|----------|
| Vanilla | 0.87 | 0.89 | 0.82 | 0.88 | 1.00 | 0.99 | 0.43 | 0.36 | 0.35 | 0.37 | 0.91 | 0.91 | 0.35 | 0.94 | 0.00 | 0.61 |
| Guo (BUZZ-1) | 0.92 | 0.98 | 0.96 | 0.98 | 0.99 | 0.99 | 0.61 | 0.73 | 0.67 | 0.73 | 0.90 | 0.90 | 0.61 | 0.90 | 0.03 | 0.39 |
| BUZZ-2 | 0.97 | 0.99 | 0.98 | 0.99 | 1.00 | 1.00 | 0.79 | 0.85 | 0.80 | 0.84 | 0.95 | 0.96 | 0.79 | 0.85 | 0.08 | 0.26 |
| BUZZ-4 | 0.98 | 1.00 | 0.99 | 1.00 | 1.00 | 1.00 | 0.83 | 0.88 | 0.85 | 0.88 | 0.97 | 0.97 | 0.83 | 0.82 | 0.12 | 0.25 |
| BUZZ-8 | 0.99 | 1.00 | 0.99 | 1.00 | 1.00 | 1.00 | 0.90 | 0.93 | 0.90 | 0.94 | 0.98 | 0.98 | 0.90 | 0.78 | 0.16 | 0.24 |
| Liu (Mixed Arch) | 0.95 | 0.96 | 0.93 | 0.95 | 1.00 | 1.00 | 0.75 | 0.67 | 0.67 | 0.68 | 0.96 | 0.96 | 0.67 | 0.90 | 0.04 | 0.33 |
| ADP | 0.88 | 0.84 | 0.78 | 0.86 | 0.99 | 0.99 | 0.40 | 0.36 | 0.32 | 0.35 | 0.93 | 0.92 | 0.32 | 0.95 | -0.01 | 0.63 |
| VanillaT-0.7 | 0.89 | 0.91 | 0.85 | 0.91 | 1.00 | 0.99 | 0.54 | 0.42 | 0.40 | 0.42 | 0.94 | 0.93 | 0.40 | 0.92 | 0.01 | 0.57 |
| VanillaT-0.95 | 0.94 | 0.95 | 0.90 | 0.94 | 1.00 | 1.00 | 0.69 | 0.51 | 0.51 | 0.51 | 0.96 | 0.96 | 0.51 | 0.89 | 0.04 | 0.49 |
| VanillaT-0.99 | 0.96 | 0.97 | 0.94 | 0.97 | 1.00 | 1.00 | 0.80 | 0.62 | 0.62 | 0.63 | 0.98 | 0.98 | 0.62 | 0.84 | 0.09 | 0.41 |
| Xie | 0.88 | 0.90 | 0.84 | 0.91 | 0.97 | 0.97 | 0.39 | 0.40 | 0.40 | 0.41 | 0.79 | 0.75 | 0.39 | 0.82 | 0.12 | 0.62 |
| Madry | 0.95 | 0.97 | 0.97 | 0.97 | 0.95 | 0.94 | 0.66 | 0.79 | 0.79 | 0.79 | 0.72 | 0.71 | 0.66 | 0.81 | 0.13 | 0.41 |
| Tramer | 0.90 | 0.97 | 0.92 | 0.97 | 1.00 | 0.99 | 0.54 | 0.60 | 0.55 | 0.60 | 0.93 | 0.92 | 0.54 | 0.94 | 0.00 | 0.43 |
| MulDef-4 | 0.90 | 0.94 | 0.89 | 0.94 | 0.99 | 0.99 | 0.53 | 0.55 | 0.52 | 0.56 | 0.93 | 0.92 | 0.52 | 0.94 | 0.00 | 0.45 |
| MulDef-8 | 0.89 | 0.95 | 0.90 | 0.96 | 0.99 | 0.99 | 0.50 | 0.56 | 0.52 | 0.57 | 0.92 | 0.93 | 0.50 | 0.94 | 0.00 | 0.47 |
| BaRT-1 | 0.91 | 0.94 | 0.91 | 0.94 | 0.99 | 0.99 | 0.55 | 0.54 | 0.49 | 0.53 | 0.89 | 0.89 | 0.49 | 0.90 | 0.03 | 0.49 |
| BaRT-4 | 0.91 | 0.94 | 0.90 | 0.94 | 0.98 | 0.98 | 0.55 | 0.54 | 0.50 | 0.52 | 0.78 | 0.80 | 0.50 | 0.83 | 0.11 | 0.52 |
| BaRT-6 | 0.91 | 0.93 | 0.90 | 0.93 | 0.97 | 0.95 | 0.52 | 0.50 | 0.47 | 0.51 | 0.74 | 0.73 | 0.47 | 0.78 | 0.15 | 0.57 |
| BaRT-8 | 0.92 | 0.91 | 0.90 | 0.94 | 0.96 | 0.95 | 0.47 | 0.45 | 0.46 | 0.47 | 0.68 | 0.68 | 0.45 | 0.71 | 0.22 | 0.61 |
| Odds | 0.87 | 0.89 | 0.82 | 0.88 | 1.00 | 0.99 | 0.43 | 0.39 | 0.37 | 0.40 | 0.94 | 0.93 | 0.37 | 0.75 | 0.18 | 0.66 |

Table 8: Boundary attacks – HopSkipJump (HSJA) (Chen & Jordan, 2019) and RayS (Chen & Gu, 2020) attacks – on different defenses for Fashion-MNIST. Minimum defense efficiency - $\text{MIN } \beta$, Clean prediction accuracy p_d , Drop in clean prediction accuracy γ .

| | HSJA | RayS | $\text{MIN } \beta$ | p_d | γ | δ |
|------------------|------|------|---------------------|-------|----------|----------|
| Vanilla | 0 | 0.09 | 0 | 0.94 | 0.00 | 0.94 |
| Guo (BUZZ-1) | 0 | 0.32 | 0 | 0.90 | 0.03 | 0.94 |
| BUZZ-2 | 0.1 | 0.61 | 0.1 | 0.85 | 0.08 | 0.85 |
| BUZZ-4 | 0.53 | 0.93 | 0.53 | 0.82 | 0.12 | 0.50 |
| BUZZ-8 | 0.89 | 1 | 0.89 | 0.78 | 0.16 | 0.24 |
| Liu (Mixed Arch) | 0 | 0.18 | 0 | 0.90 | 0.04 | 0.94 |
| ADP | 0 | 0.04 | 0 | 0.95 | -0.01 | 0.94 |
| VanillaT-0.7 | 0 | 0.1 | 0 | 0.92 | 0.01 | 0.94 |
| VanillaT-0.95 | 0 | 0.18 | 0 | 0.89 | 0.04 | 0.94 |
| VanillaT-0.99 | 0 | 0.47 | 0 | 0.84 | 0.09 | 0.94 |
| Xie | 0.85 | 0.63 | 0.63 | 0.82 | 0.12 | 0.42 |
| Madry | 0.99 | 0.96 | 0.96 | 0.81 | 0.13 | 0.16 |
| Tramer | 0 | 0.18 | 0 | 0.94 | 0.00 | 0.94 |
| MulDef-4 | 0.82 | 0.66 | 0.66 | 0.94 | 0.00 | 0.32 |
| MulDef-8 | 0.92 | 0.7 | 0.7 | 0.94 | 0.00 | 0.28 |

the smallest $\delta = 0.24$ across the union of all three classes of attacks. Madry achieves the second smallest $\delta = 0.54$ across the union of all three classes of attacks – this shows the significant improvement realized by the BUZZ family for the Fashion-MNIST data set.

- 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 δ . If we have a close look at the results presented in Tables 6, 7 and 8, we can see that they are vulnerable to black-box attacks. In other words, they offer no security.
- Defenses that have poorer clean accuracy compared to that of the vanilla classifier achieve a smaller δ such as Odds, Madry and BUZZ. These defenses do offer some security. Among them, BUZZ is the best performing one in terms of security.
- By combining the drop γ in clean accuracy and the increment in defense accuracy β , the δ 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 α (or, equivalently, defense accuracy β) and clean accuracy of the defense p_d . For example, in Table 8, we can see that Madry has a smaller δ than BUZZ-8 but if we have a close look at β and p_d , then we can see that their performances are more alike than what the δ metric alone would suggest.

- From Tables 6 and 7 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 8, boundary attacks are much stronger than mixed and pure black-box attacks and this is understandable because boundary attacks use much more queries than the other black-box attacks.
- BUZZ can realize different combinations of defender accuracy p_d and attacker's success rate α by tuning the number of protected classifiers in the defense. We notice that BUZZ can adopt the strategy of MulDef to increase the robustness against boundary attacks. We have not investigated this direction because we want to see how strong our defense stands on itself. Nevertheless, this shows an advantage of BUZZ, i.e., BUZZ is can be combined with other defense strategies in a flexible way.

D.2 CIFAR-10

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

Table 9: Targeted (T) and Untargeted (U) mixed black-box attacks on different defenses for CIFAR-10. Minimum defense efficiency - MIN β , Clean prediction accuracy p_d , Drop in clean prediction accuracy γ .

| | FGSM-T | IFGSM-T | MIM-T | PGD-T | CW-T | EAD-T | FGSM-U | IFGSM-U | MIM-U | PGD-U | CW-U | EAD-U | MIN β | p_d | γ | δ |
|------------------|--------|---------|-------|-------|------|-------|--------|---------|-------|-------|------|-------|-------------|-------|----------|----------|
| Vanilla | 0.87 | 0.86 | 0.78 | 0.85 | 0.99 | 0.99 | 0.33 | 0.39 | 0.26 | 0.37 | 0.99 | 0.99 | 0.26 | 0.93 | 0.00 | 0.69 |
| Guo (BUZZ-1) | 0.89 | 0.90 | 0.83 | 0.90 | 0.99 | 0.99 | 0.48 | 0.57 | 0.45 | 0.56 | 0.99 | 0.99 | 0.45 | 0.91 | 0.02 | 0.52 |
| BUZZ-2 | 0.97 | 0.97 | 0.95 | 0.97 | 1.00 | 1.00 | 0.81 | 0.81 | 0.75 | 0.83 | 1.00 | 1.00 | 0.75 | 0.85 | 0.08 | 0.29 |
| BUZZ-4 | 0.99 | 0.99 | 0.98 | 0.98 | 1.00 | 1.00 | 0.92 | 0.90 | 0.88 | 0.91 | 1.00 | 1.00 | 0.88 | 0.81 | 0.12 | 0.21 |
| BUZZ-8 | 0.99 | 0.99 | 0.98 | 0.99 | 1.00 | 1.00 | 0.96 | 0.96 | 0.93 | 0.95 | 1.00 | 1.00 | 0.93 | 0.76 | 0.17 | 0.23 |
| Liu (Mixed Arch) | 0.94 | 0.94 | 0.88 | 0.94 | 1.00 | 1.00 | 0.73 | 0.70 | 0.63 | 0.71 | 0.99 | 1.00 | 0.63 | 0.85 | 0.08 | 0.39 |
| ADP | 0.84 | 0.70 | 0.61 | 0.71 | 1.00 | 0.99 | 0.33 | 0.22 | 0.15 | 0.23 | 0.99 | 0.99 | 0.15 | 0.94 | -0.01 | 0.79 |
| VanillaT-0.7 | 0.91 | 0.89 | 0.84 | 0.89 | 1.00 | 1.00 | 0.55 | 0.59 | 0.52 | 0.59 | 0.99 | 0.99 | 0.52 | 0.90 | 0.03 | 0.46 |
| VanillaT-0.95 | 0.96 | 0.96 | 0.93 | 0.96 | 1.00 | 1.00 | 0.80 | 0.81 | 0.77 | 0.82 | 1.00 | 1.00 | 0.77 | 0.85 | 0.08 | 0.27 |
| VanillaT-0.99 | 0.98 | 0.98 | 0.97 | 0.98 | 1.00 | 1.00 | 0.93 | 0.92 | 0.89 | 0.91 | 1.00 | 1.00 | 0.89 | 0.79 | 0.14 | 0.22 |
| Xie | 0.83 | 0.82 | 0.76 | 0.86 | 0.98 | 0.98 | 0.30 | 0.38 | 0.26 | 0.37 | 0.84 | 0.86 | 0.26 | 0.71 | 0.22 | 0.74 |
| Madry | 0.96 | 0.98 | 0.96 | 0.98 | 1.00 | 1.00 | 0.78 | 0.84 | 0.77 | 0.84 | 0.99 | 0.98 | 0.77 | 0.75 | 0.18 | 0.35 |
| Tramer | 0.90 | 0.93 | 0.85 | 0.94 | 1.00 | 1.00 | 0.57 | 0.62 | 0.44 | 0.64 | 0.99 | 0.98 | 0.44 | 0.85 | 0.08 | 0.55 |
| MulDef-4 | 0.89 | 0.90 | 0.82 | 0.91 | 0.99 | 0.99 | 0.49 | 0.53 | 0.37 | 0.54 | 0.93 | 0.92 | 0.37 | 0.87 | 0.06 | 0.60 |
| MulDef-8 | 0.89 | 0.91 | 0.82 | 0.89 | 0.98 | 0.99 | 0.49 | 0.57 | 0.40 | 0.56 | 0.92 | 0.92 | 0.40 | 0.86 | 0.07 | 0.59 |
| BaRT-1 | 0.89 | 0.87 | 0.80 | 0.88 | 0.98 | 0.97 | 0.51 | 0.55 | 0.40 | 0.54 | 0.92 | 0.93 | 0.40 | 0.86 | 0.07 | 0.59 |
| BaRT-4 | 0.91 | 0.91 | 0.87 | 0.90 | 0.97 | 0.97 | 0.48 | 0.55 | 0.44 | 0.58 | 0.80 | 0.79 | 0.44 | 0.75 | 0.18 | 0.59 |
| BaRT-7 | 0.90 | 0.92 | 0.88 | 0.92 | 0.96 | 0.95 | 0.45 | 0.53 | 0.43 | 0.54 | 0.70 | 0.68 | 0.43 | 0.61 | 0.32 | 0.67 |
| BaRT-10 | 0.91 | 0.91 | 0.90 | 0.92 | 0.93 | 0.94 | 0.39 | 0.47 | 0.38 | 0.49 | 0.58 | 0.58 | 0.38 | 0.49 | 0.44 | 0.74 |
| Odds | 0.94 | 0.94 | 0.91 | 0.93 | 1.00 | 0.99 | 0.65 | 0.66 | 0.58 | 0.67 | 0.97 | 0.98 | 0.58 | 0.71 | 0.22 | 0.52 |

Table 10: Targeted (T) and Untargeted (U) pure black-box attacks on different defenses for CIFAR-10. Minimum defense efficiency - MIN β , Clean prediction accuracy p_d , Drop in clean prediction accuracy γ .

| | FGSM-T | IFGSM-T | MIM-T | PGD-T | CW-T | EAD-T | FGSM-U | IFGSM-U | MIM-U | PGD-U | CW-U | EAD-U | MIN β | p_d | γ | δ |
|------------------|--------|---------|-------|-------|------|-------|--------|---------|-------|-------|------|-------|-------------|-------|----------|----------|
| Vanilla | 0.90 | 0.92 | 0.85 | 0.92 | 0.98 | 0.98 | 0.44 | 0.45 | 0.38 | 0.46 | 0.92 | 0.92 | 0.38 | 0.93 | 0.00 | 0.57 |
| Guo (BUZZ-1) | 0.91 | 0.94 | 0.88 | 0.94 | 0.98 | 0.99 | 0.49 | 0.54 | 0.45 | 0.57 | 0.90 | 0.90 | 0.45 | 0.91 | 0.02 | 0.52 |
| BUZZ-2 | 0.96 | 0.97 | 0.95 | 0.97 | 1.00 | 1.00 | 0.80 | 0.76 | 0.70 | 0.79 | 0.97 | 0.97 | 0.70 | 0.85 | 0.08 | 0.33 |
| BUZZ-4 | 0.97 | 0.99 | 0.97 | 0.99 | 1.00 | 1.00 | 0.89 | 0.88 | 0.84 | 0.88 | 0.99 | 0.99 | 0.84 | 0.81 | 0.12 | 0.25 |
| BUZZ-8 | 0.99 | 0.99 | 0.98 | 0.99 | 1.00 | 1.00 | 0.95 | 0.93 | 0.89 | 0.93 | 1.00 | 1.00 | 0.89 | 0.76 | 0.17 | 0.25 |
| Liu (Mixed Arch) | 0.94 | 0.97 | 0.91 | 0.97 | 0.99 | 0.99 | 0.74 | 0.70 | 0.65 | 0.69 | 0.97 | 0.97 | 0.65 | 0.85 | 0.08 | 0.37 |
| ADP | 0.91 | 0.93 | 0.86 | 0.94 | 0.99 | 0.99 | 0.49 | 0.48 | 0.39 | 0.49 | 0.93 | 0.93 | 0.39 | 0.94 | -0.02 | 0.56 |
| VanillaT-0.7 | 0.93 | 0.95 | 0.90 | 0.95 | 0.99 | 0.99 | 0.59 | 0.59 | 0.54 | 0.61 | 0.95 | 0.94 | 0.54 | 0.90 | 0.02 | 0.44 |
| VanillaT-0.95 | 0.96 | 0.97 | 0.95 | 0.97 | 1.00 | 1.00 | 0.80 | 0.77 | 0.72 | 0.78 | 0.97 | 0.97 | 0.72 | 0.85 | 0.08 | 0.32 |
| VanillaT-0.99 | 0.98 | 0.99 | 0.96 | 0.99 | 1.00 | 1.00 | 0.90 | 0.86 | 0.83 | 0.86 | 0.99 | 0.99 | 0.83 | 0.79 | 0.14 | 0.27 |
| Xie | 0.90 | 0.93 | 0.87 | 0.93 | 0.96 | 0.96 | 0.41 | 0.44 | 0.35 | 0.41 | 0.69 | 0.71 | 0.35 | 0.71 | 0.22 | 0.68 |
| Madry | 0.90 | 0.92 | 0.89 | 0.91 | 0.90 | 0.85 | 0.55 | 0.60 | 0.53 | 0.60 | 0.69 | 0.66 | 0.53 | 0.75 | 0.18 | 0.53 |
| Tramer | 0.90 | 0.96 | 0.88 | 0.96 | 0.98 | 0.98 | 0.56 | 0.54 | 0.44 | 0.56 | 0.85 | 0.85 | 0.44 | 0.85 | 0.08 | 0.55 |
| MulDef-4 | 0.88 | 0.93 | 0.84 | 0.94 | 0.98 | 0.98 | 0.50 | 0.49 | 0.36 | 0.48 | 0.86 | 0.86 | 0.36 | 0.87 | 0.06 | 0.61 |
| MulDef-8 | 0.88 | 0.95 | 0.86 | 0.94 | 0.98 | 0.98 | 0.52 | 0.51 | 0.38 | 0.50 | 0.85 | 0.84 | 0.38 | 0.86 | 0.07 | 0.60 |
| BaRT-1 | 0.91 | 0.94 | 0.88 | 0.96 | 0.98 | 0.98 | 0.59 | 0.59 | 0.47 | 0.61 | 0.85 | 0.85 | 0.47 | 0.86 | 0.07 | 0.52 |
| BaRT-4 | 0.91 | 0.95 | 0.88 | 0.93 | 0.98 | 0.96 | 0.54 | 0.55 | 0.45 | 0.56 | 0.74 | 0.74 | 0.45 | 0.75 | 0.18 | 0.59 |
| BaRT-7 | 0.91 | 0.93 | 0.89 | 0.92 | 0.95 | 0.95 | 0.48 | 0.48 | 0.38 | 0.48 | 0.59 | 0.57 | 0.38 | 0.61 | 0.32 | 0.70 |
| BaRT-10 | 0.90 | 0.92 | 0.90 | 0.91 | 0.93 | 0.93 | 0.40 | 0.37 | 0.37 | 0.41 | 0.47 | 0.46 | 0.37 | 0.49 | 0.44 | 0.75 |
| Odds | 0.96 | 0.96 | 0.92 | 0.97 | 0.99 | 0.99 | 0.76 | 0.66 | 0.62 | 0.68 | 0.93 | 0.93 | 0.62 | 0.71 | 0.21 | 0.49 |

Discussion. We have the following observations from Tables 9, 10 and 11:

- The BUZZ family achieves the smallest δ for mixed black box attacks, the smallest δ for pure black box attacks and ranks below Xie, Madry, and MulDef with a higher $\delta = 0.63$ for boundary attacks.

Table 11: Boundary attacks – HopSkipJump (HSJA) (Chen & Jordan, 2019) and RayS (Chen & Gu, 2020) attacks – on different defenses for CIFAR-10. Minimum defense efficiency - MIN β , Clean prediction accuracy p_d , Drop in clean prediction accuracy γ .

| | HSJA | RayS | MIN β | p_d | γ | δ |
|------------------|------|------|-------------|-------|----------|----------|
| Vanilla | 0 | 0.02 | 0 | 0.93 | 0.00 | 0.93 |
| Guo (BUZz-1) | 0 | 0.01 | 0 | 0.91 | 0.02 | 0.93 |
| BUZz-2 | 0 | 0.04 | 0 | 0.85 | 0.08 | 0.93 |
| BUZz-4 | 0.16 | 0.27 | 0.16 | 0.81 | 0.12 | 0.80 |
| BUZz-8 | 0.39 | 0.6 | 0.39 | 0.76 | 0.17 | 0.63 |
| Liu (Mixed Arch) | 0 | 0.29 | 0 | 0.85 | 0.08 | 0.93 |
| ADP | 0 | 0.05 | 0 | 0.94 | -0.02 | 0.93 |
| VanillaT-0.7 | 0 | 0.12 | 0 | 0.90 | 0.02 | 0.93 |
| VanillaT-0.95 | 0 | 1 | 0 | 0.85 | 0.08 | 0.93 |
| VanillaT-0.99 | 0 | 1 | 0 | 0.79 | 0.14 | 0.93 |
| Xie | 0.84 | 0.85 | 0.84 | 0.71 | 0.22 | 0.33 |
| Madry | 0.52 | 0.66 | 0.52 | 0.75 | 0.18 | 0.54 |
| Tramer | 0 | 0.02 | 0 | 0.85 | 0.08 | 0.93 |
| MulDef-4 | 0.83 | 0.7 | 0.7 | 0.87 | 0.06 | 0.32 |
| MulDef-8 | 0.88 | 0.74 | 0.74 | 0.86 | 0.07 | 0.29 |

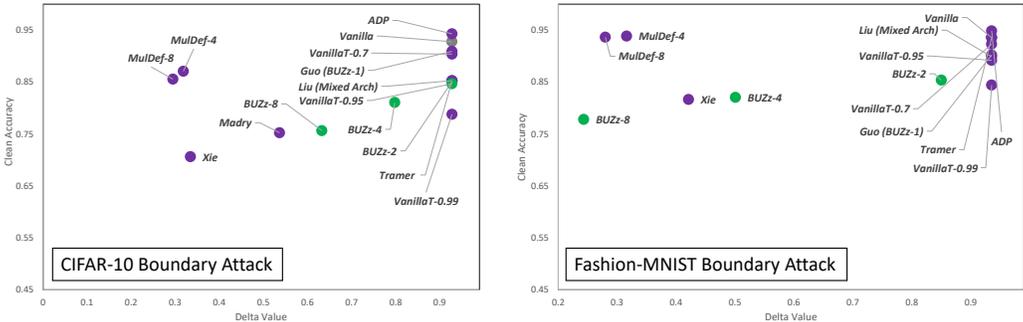


Figure 4: The δ metric vs clean accuracy for boundary attacks. The BUZZ results are shown in green and the vanilla result is shown in gray.

2. The BUZZ family achieves $\delta = 0.63$ across the union of all three classes of attacks while Xie, Madry, and MulDef achieve $\delta = 0.74$, $\delta = 0.54$ and $\delta = 0.60$ making BUZZ comparable to Xie, Madry and MulDef in this sense. A more detailed look reveals that, while BUZZ, Xie, Madry and MulDef have similar overall δ , they perform differently among the three attack classes: Xie and MulDef have low $\delta = 0.33$ and $\delta = 0.29$ for boundary attacks; BUZZ and Madry have low $\delta = 0.21$ and $\delta = 0.35$ for mixed black-box attacks; BUZZ has low $\delta = 0.25$ for pure black-box attacks. We conclude that these defense strategies are different and cover different strengths.
3. 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 δ . If we have a close look at the results presented in Tables 9, 10 and 11, we can see that they are vulnerable to black-box attacks. In other words, they offer no security.
4. By combining the drop γ in clean accuracy and the increment in defense accuracy β , the δ 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 α (or, equivalently, defense accuracy β) and clean accuracy of the defense p_d . For example, in Table 11, we can see that Madry has a smaller δ than BUZZ-8 but if we have a close look at β and p_d , then we can see that their performances are more alike than what the δ metric alone would suggest.

5. From Tables 9 and 10 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 11, boundary attacks are much stronger than mixed and pure black-box attacks and this is understandable because boundary attacks use much more queries than the other black-box attacks.
6. BUZZ can realize different combinations of defender accuracy p_d and attacker’s success rate α by tuning the number of protected classifiers in the defense.
7. Xie and MulDef have the smallest δ 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 outputted 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). We notice that we can also adopt this approach to enhance the robustness of BUZZ against boundary attacks. We have not investigated this direction because we want to see how strong our defense stands on itself. Nevertheless, this shows an advantage of BUZZ, i.e., BUZZ is can be combined with other defense strategies in a flexible way.

D.3 BOUNDARY ATTACK COMPUTATIONAL COMPLEXITY

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 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).

D.4 BUFFER ZONE GRAPHS

In Figure 2 we show buffer zone graphs for various defenses. These graphs are based on the decision region graphs originally presented in Liu et al. (2017). 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 using the following equation: $I' = I + x \cdot g + y \cdot r$. Here g represents the gradient of the loss function with respect to I . In the equation, r represents a normalized random matrix that is orthogonal to I (note g is also normalized). The other 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: At the origin I' is equal to I . The origin is 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, each network j will have a different gradient matrix g_j . To compensate for this, we average the individual g_j matrices together before normalizing to get g . It is important to note that while the graphs shown in Figure 2 give experimental proof of the concept of buffer zones, they cannot be used to attack BUZZ defenses in practice. When creating the graphs, we have knowledge of the individual gradient matrices g_j for each individual network j . This information is not available or obtainable by an adversary in a black-box setting, to the best of our knowledge.