entropy

MDPI

Article

Beware the Black-Box: On the Robustness of Recent Defenses to Adversarial Examples

Kaleel Mahmood [1,*], Deniz Gurevin [2], Marten van Dijk [3] and Phuoung Ha Nguyen [4]

[1] Department of Computer Science and Engineering, University of Connecticut, Storrs, CT 06269, USA
[2] Department of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269, USA; deniz.gurevin@uconn.edu
[3] CWI, 1098 XG Amsterdam, The Netherlands; Marten.van.Dijk@cwi.nl
[4] eBay, San Jose, CA 95125, USA; phuongha.ntu@gmail.com
* Correspondence: kaleel.mahmood@uconn.edu

Abstract: Many defenses have recently been proposed at venues like NIPS, ICML, ICLR and CVPR. These defenses are mainly focused on mitigating white-box attacks. They do not properly examine black-box attacks. In this paper, we expand upon the analyses of these defenses to include adaptive black-box adversaries. Our evaluation is done on nine defenses including Barrage of Random Transforms, ComDefend, Ensemble Diversity, Feature Distillation, The Odds are Odd, Error Correcting Codes, Distribution Classifier Defense, K-Winner Take All and Buffer Zones. Our investigation is done using two black-box adversarial models and six widely studied adversarial attacks for CIFAR-10 and Fashion-MNIST datasets. Our analyses show most recent defenses (7 out of 9) provide only marginal improvements in security (<25%), as compared to undefended networks. For every defense, we also show the relationship between the amount of data the adversary has at their disposal, and the effectiveness of adaptive black-box attacks. Overall, our results paint a clear picture: defenses need both thorough white-box and black-box analyses to be considered secure. We provide this large scale study and analyses to motivate the field to move towards the development of more robust black-box defenses.

Keywords: adversarial machine learning; black-box attacks; security

check for updates

Citation: Mahmood, K.; Gurevin, D.; van Dijk, M.; Nguyen, P.H. Beware the Black-Box: On the Robustness of Recent Defenses to Adversarial Examples. *Entropy* **2021**, *23*, 1359. https://doi.org/10.3390/e23101359

Academic Editor: Luis Hernández-Callejo

Received: 16 September 2021
Accepted: 14 October 2021
Published: 18 October 2021

1. Introduction

Convolutional Neural Networks (CNNs) are widely used for image classification [1,2] and object detection. Despite their widespread use, CNNs have been shown to be vulnerable to adversarial examples [3]. Adversarial examples are clean images which have malicious noise added to them. This noise is small enough so that humans can visually recognize the images, but CNNs misclassify them.

Adversarial examples can be created through white-box or black-box attacks, depending on the assumed adversarial model. White-box attacks create adversarial examples by directly using information about the trained parameters in a classifier (e.g., the weights of a CNN). Black-box attacks on the other hand, assume an adversarial model where the trained parameters of the classifier are secret or unknown. In black-box attacks, the adversary generates adversarial examples by exploiting other information such as querying the classifier [4–6], or using the original dataset the classifier was trained on [7–10]. We can also further categorize black-box attacks based on whether the attack tries to tailor the adversarial example to specifically overcome the defense (adaptive black-box attacks), or if the attack is fixed regardless of the defense (non-adaptive black-box attacks). In terms of attacks, we focus on adaptive black-box adversaries. A natural question is why do we choose this scope?

(1) White-box robustness does not automatically mean black-box robustness. In security communities such as cryptology, black-box attacks are considered strictly weaker than

white-box attacks. This means that if a defense is shown to be secure against a white-box adversary, it would also be secure against a black-box adversary. In the field of adversarial machine learning, this principle does NOT always hold true. Why does this happen? In adversarial machine learning, white-box attacks use gradient information directly to create adversarial examples. It is possible to obfuscate this gradient, an effect known as gradient masking [9] and thus make white-box attacks fail. Black-box attacks do not directly use gradient information. As a result, black-box attacks may still be able to work on defenses that have gradient masking. This means adversarial machine learning defenses need to be analyzed under both white-box AND black-box attacks.

(2) White-box adversaries are well studied in most defense papers [11–18] as opposed to black-box adversaries. Less attention has been given to black-box attacks, despite the need to test defenses on both types of attacks (as mentioned in our first point). This paper offers a unique perspective by testing defenses under adaptive black-box attacks. By combining the white-box analyses already developed in the literature with the black-box analyses we present here, we give a full security picture.

Having explained our focus for the type of attacks, we next explain why we chose the following 9 defenses to investigate:

(1) Each defense is unique in the following aspect: No two defenses use the exact same set of underlying methods to try and achieve security. We illustrate this point in Table 1. Further in Section 3 we go into specifics about why each individual defense is chosen. As a whole, this diverse group of defenses allows us to evaluate many different competing approaches to security.

(2) Most of the defenses we analyze have been published at NIPS, ICML, ICLR or CVPR. This indicates the machine learning community and reviewers found these approaches worthy of examination and further study.

Major Contributions, Related Literature and Paper Organization

Having briefly introduced the notion of adversarial machine learning attacks and explained the scope of our work, we discuss several other important introductory points. First, we list our major contributions. Second, we discuss literature that is related but distinct from our work. Finally, we give an overview of the organization of the rest of our paper. Major contributions:

1. *Comprehensive black-box defense analysis*—Our experiments are comprehensive and rigorous in the following ways: we work with 9 recent defenses and a total of 12 different attacks. Every defense is trained on the same dataset and with the same base CNN architecture whenever possible. Every defense is attacked under the same adversarial model. This allows us to directly compare defense results. It is important to note some papers use different adversarial models which makes comparisons across papers invalid [19].
2. *Adaptive adversarial strength study*—In this paper we are the first (to the best of our knowledge) to show the relationship between each of the 9 defenses and the strength of an adaptive black-box adversary. Specifically, for every defense we are able to show how its security is effected by varying the amount of training data available to an adaptive black-box adversary (i.e., 100%, 75%, 50%, 25% and 1%).
3. *Open source code and detailed implementations*—One of our main goals of this paper is to help the community develop stronger black-box adversarial defenses. To this end, we publicly provide code for our experiments: https://github.com/MetaMain/BewareAdvML (accessed on 20 May 2021). In addition, in Appendix A we give detailed instructions for how we implemented each defense and what experiments we ran to fine tune the hyperparameters of the defense.

Related Literature: There are a few works that are related but distinctly different from our paper. We briefly discuss them here. As we previously mentioned, the field of adversarial machine learning has mainly been focused on white-box attacks on defenses. Works that consider white-box attacks and/or multiple defenses include [20–24].

In [20] the authors test white-box and black-box attacks on defenses proposed in 2017, or earlier. It is important to note, all the defenses in our paper are from 2018 or later. There is no overlap between our work and the work in [20] in terms of defenses studied. In addition, in [20], while they do consider a black-box attack, it is not adaptive because they do not give the attacker access to the defense training data.

In [21], an ensemble is studied by trying to combine multiple weak defenses to form a strong defense. Their work shows that such a combination does not produce a strong defense under a white-box adversary. None of the defenses covered in our paper are used in [21]. Also [21] does not consider a black-box adversary like our work.

In [23], the authors also do a large study on adversarial machine learning attacks and defenses. It is important to note that they do not consider adaptive black-box attacks, as we define them (see Section 2). They do test defenses on CIFAR-10 like us, but in this case only one defense (ADP [11]) overlaps with our study. To reiterate, the main threat we are concerned with is adaptive black-box attacks which is not covered in [23].

One of the closest studies to us is [22]. In [22] the authors also study adaptive attacks. However, unlike our analyses which use black-box attacks, they assume a white-box adversary. Our paper is a natural progression from [22] in the following sense: If the defenses studied in [22] are broken under an adaptive white-box adversary, could these defenses still be effective under under a weaker adversarial model? In this case, the model in question would be one that disallows white-box access to the defense, i.e., a black-box adversary. Whether these defenses are secure against adaptive black-box adversaries is an open question, and one of the main questions our paper seeks to answer.

Lastly, adaptive black-box adversaries have also been studied before in [24]. However, they do not consider variable strength adaptive black-box adversaries as we do. We also cover many defenses that are not included in their paper (Error Correcting Codes, Feature Distillation, Distribution Classifier, K-Winner Take All and ComDefend). Finally, the metric we use to compare defenses is fundamentally different from the metric proposed in [24]. They compare results using a metric that balances clean accuracy and security. In this paper, we study the performance of a defense relative to no defense (i.e., a vanilla classifier).

Paper Organization: Our paper is organized as follows: in Section 2, we describe the goal of the adversary mathematically, the capabilities given in different adversarial models and the categories of black-box attacks. In Section 3, we break down the defenses used in this paper in terms of their underlying defense mechanisms. We also explain why each individual defense was selected for analysis in this paper. In Section 4, we discuss the principal experimental results and compare the performances of the defenses. In Section 5, we analyze and discuss each defense individually. We also show the relationship between the security of each defense and the strength (amount of training data) available to an adaptive black-box adversary. We offer concluding remarks in Section 6. Lastly, full experimental details and defense implementation instructions are given in the Appendix A.

Table 1. Defenses analyzed in this paper and the corresponding defense mechanisms they employ. For definitions of the each defense mechanism see Section 3.

Defense Mechanism	**Ensemble Diversity (ADP) [11]**	**Error Correcting Codes (ECOC) [12]**	**Buffer Zones (BUZz) [24]**	**Com Defend [13]**	**Barrage (BaRT) [14]**	**Distribution Classifier (DistC) [16]**	**Feature Distillation (FD) [18]**	**Odds Are Odd [17]**	**K-Winner (k-WTA) [15]**
Mutiple Models	✓	✓	✓						
Fixed Input Transformation			✓	✓			✓		
Random Input Transformation				✓	✓	✓		✓	
Adversarial Detection			✓					✓	
Network Retraining	✓	✓			✓	✓			✓
Architecture Change		✓				✓			✓

2. Attacks

2.1. Attack Setup

The general setup for an attack in adversarial machine learning can be defined in the following way [25]: The adversary is given a trained classifier F which outputs a class label l for a given input x such that $F(x) = l$. In this paper, the classifiers we consider are deep Convolutional Neural Networks (CNN), and the inputs (x) are images. The goal of the adversary is to create an adversarial example from the original input x by adding a small noise η. The adversarial example that is created is a perturbed version of the original input: $x' = x + \eta$. There are two criteria for the attack to be considered successful:

1. The adversarial example x' must make the classifier produce a certain class label: $F(x') = c$. Here the certain class label c depends on whether the adversary is attempting a targeted, or untargeted type of attack. In a targeted attack c is a specific wrong class label (e.g., a picture of cat MUST be recognized as a dog by the classifier). On the other hand, if the attack is untargeted, the only criteria for c is that it must not be the same as the original class label: $c \neq l$ (e.g., as long as a picture of a cat is labeled by the classifier as anything except a cat, the attack is successful).
2. The noise η used to create the adversarial image x' must be barely recognizable by humans. This constraint is enforced by limiting the size of perturbation η such that the difference between the original input x and the perturbed input x' is less than a certain distance d. This distance d is typically measured [19] using the l_p norm: $\|x' - x\|_p \leq d$

In summary, an attack is considered successful if the classifier produces an output label desired by the adversary $F(x') = c$ and the difference between the original input x and the adversarial sample x' is small enough, $\|x' - x\|_p \leq d$.

2.2. Adversarial Capabilities

In this subsection, we go over what information the adversary can use to create adversarial examples. Specifically, the adversarial model defines what information is available to the attacker to assist them in crafting the perturbation η. In Table 2 we give an overview of the attacks and the adversarial capabilities need to run the attack. Such abilities can be broadly grouped into the following categories:

1. Having knowledge of the trained parameters and architecture of the classifier. For example, when dealing with CNNs (as is the focus of this paper) knowing the architecture means knowing precisely which type of CNN is used. Example CNN

architectures include VGG-16, ResNet56 etc. Knowing the trained parameters for a CNN means the values of the weights and biases of the network (as well as any other trainable parameters) are visible to the attacker [19].

2. Query access to the classifier. If the architecture and trained parameters are kept private, then the next best adversarial capability is having query access to the target model as a black-box. The main concept here is that the adversary can adaptively query the classifier [26] with different inputs to help create the adversarial perturbation η. Query access can come in two forms. In the stronger version, when the classifier is queried, the entire probability score vector is returned (i.e., the softmax output from a CNN). Naturally this gives the adversary more information to work with because the confidence in each label is given. In the weaker version, when the classifier is queried, only the final class label is returned (the index of the score vector with the highest value).
3. Having access to (part of the) training or testing data. In general, for any adversarial machine learning attack, at least one example must be used to start the attack. Hence, every attack requires some input data. However, how much input data the adversary has access to depends on the type of attack (or parameters in the attack). Knowing part or all of the training data used to build the classifier can be especially useful when the architecture and trained parameters of the classifier are not available. This is because the adversary can try to replicate the classifier in the defense, by training their own classifier with the given training data [8].

2.3. Types of Attacks

The types of attacks in machine learning can be grouped based on the capabilities the adversary needs to conduct the attack. We described these different capabilities in Section 2.2. In this section, we describe the attacks and what capabilities the adversary must have to run them.

White-box attacks: Examples of white-box attacks include the Fast Gradient Sign Method (FGSM) [3], Projected Gradient Descent (PGD) [27] and Carlini & Wagner (C&W) [28] to name a few. They require having knowledge of the trained parameters and architecture of the classifier, as well as query access. In white-box attacks like FGSM and PGD, having access to the classifier's trained parameters allows the adversary to use a form of backpropagation. By calculating the gradient with respect to the input, the adversarial perturbation η can be estimated directly. In some defenses, where directly backpropagating on the classifier may not be applicable or yield poor results, it is possible to create attacks tailored to the defense that are more effective. These are referred to as adaptive attacks [22]. In general, white-box attacks and defenses against them have been heavily focused on in the literature. In this paper, our focus is on black-box attacks. Hence, we only give a brief summary of the white-box attacks as mentioned above.

Black-Box Attacks: The biggest difference between white-box and black-box attacks is that black-box attacks lack access to the trained parameters and architecture of the defense. As a result, they need to either have training data to build a synthetic model, or use a large number of queries to create an adversarial example. Based on these distinctions, we can categorize black-box attacks as follows:

1. Query only black-box attacks [26]. The attacker has query access to the classifier. In these attacks, the adversary does not build any synthetic model to generate adversarial examples or make use of training data. Query only black-box attacks can further be divided into two categories: score based black-box attacks and decision based black-box attacks.
    - Score based black-box attacks. These are also referred to as zeroth order optimization based black-box attacks [5]. In this attack, the adversary adaptively queries the classifier with variations of an input x and receives the output from the softmax layer of the classifier $f(x)$. Using $x, f(x)$ the adversary attempts to approximate the gradient of the classifier ∇f and create an adversarial example.

SimBA is an example of one of the more recently proposed score based black-box attacks [29].

- Decision based black-box attacks. The main concept in decision based attacks is to find the boundary between classes using only the hard label from the classifier. In these types of attacks, the adversary does not have access to the output from the softmax layer (they do not know the probability vector). Adversarial examples in these attacks are created by estimating the gradient of the classifier by querying using a binary search methodology. Some recent decision based black-box attacks include HopSkipJump [6] and RayS [30].

2. Model black-box attacks. In model black-box attacks, the adversary has access to part or all of the training data used to train the classifier in the defense. The main idea here is that the adversary can build their own classifier using the training data, which is called the synthetic model. Once the synthetic model is trained, the adversary can run any number of white-box attacks (e.g., FGSM [3], BIM [31], MIM [32], PGD [27], C&W [28] and EAD [33]) on the synthetic model to create adversarial examples. The attacker then submits these adversarial examples to the defense. Ideally, adversarial examples that succeed in fooling the synthetic model will also fool the classifier in the defense. Model black-box attacks can further be categorized based on how the training data in the attack is used:

   - Adaptive model black-box attacks [4]. In this type of attack, the adversary attempts to adapt to the defense by training the synthetic model in a specialized way. Normally, a model is trained with dataset X and corresponding class labels Y. In an adaptive black-box attack, the original labels Y are discarded. The training data X is re-labeled by querying the classifier in the defense to obtain class labels $\hat{Y}$. The synthetic model is then trained on $(X, \hat{Y})$ before being used to generate adversarial examples. The main concept here is that by training the synthetic model with $(X, \hat{Y})$, it will more closely match or adapt to the classifier in the defense. If the two classifiers closely match, then there will (hopefully) be a higher percentage of adversarial examples generated from the synthetic model that fool the classifier in the defense. To run adaptive black-box attacks, access to at least part of the training data and query access to the defense is required. If only a small percentage of the training data is known (e.g., not enough training data to train a CNN), the adversary can also generate synthetic data and label it using query access to the defense [4].
   - Pure black-box attacks [7–10]. In this type of attack, the adversary also trains a synthetic model. However, the adversary does not have query access to make the attack adaptive. As a result, the synthetic model is trained on the original dataset and original labels (X, Y). In essence this attack is defense agnostic (the training of the synthetic model does not change for different defenses).

Table 2. Adversarial machine learning attacks and the adversarial capabilities required to execute the attack. For a full description of these capabilities, see Section 2.2.

	Adversarial Capabilities			
	Training/Testing Data	**Hard Label Query Access**	**Score Based Query Access**	**Trained Parameters**
White-Box		✓	✓	✓
Score Based Black-Box		✓	✓	
Decision Based Black-Box		✓		
Adaptive Black-Box	✓	✓		
Pure Black-Box	✓			

2.4. Our Black-Box Attack Scope

We focus on black-box attacks, specifically the adaptive black-box and pure black-box attacks. Why do we refine our scope in this way? First of all we don't focus on white-box attacks as mentioned in Section 1 as this is well documented in the current literature. In addition, simply showing white-box security is not enough in adversarial machine learning. Due to gradient masking [9], there is a need to demonstrate both white-box and black-box robustness. When considering black-box attacks, as we explained in the previous subsection, there are query only black-box attacks and model black-box attacks. Score based query black-box attacks can be neutralized by a form of gradient masking [19]. Furthermore, it has been noted that a decision based query black-box attack represents a more practical adversarial model [34]. However, even with these more practical attacks there are disadvantages. It has been claimed that decision based black-box attacks may perform poorly on randomized models [19,23]. It has also been shown that even adding a small Gaussian noise to the input may be enough to deter query black-box attacks [35]. Due to their poor performance in the presence of even small randomization, we do not consider query black-box attacks.

Focusing on black-box adversaries and discounting query black-box attacks, leaves model black-box attacks. In our analyses, we first use the pure black-box attack because this attack has no adaptation and no knowledge of the defense. In essence it is the least capable adversary. It may seem counter-intuitive to start with a weak adversarial model. However, by using a relatively weak attack we can see the security of the defense under idealized circumstances. This represents a kind of best-case defense scenario.

The second type of attack we focus on is the adaptive black-box attack. This is the strongest model black-box type of attack in terms of the powers given to the adversary. In our study on this attack, we also vary its strength by giving the adversary different amounts of the original training data (1%, 25%, 50%, 75% and 100%). For the defense, this represents a stronger adversary, one that has query access, training data and an adaptive way to try and tailor the attack to break the defense. In short, we chose to focus on the pure and adaptive black-box attacks. We do this because they do not suffer from the limitations of the query black-box attacks, and they can be used as an efficient and nearly universally applicable security test.

3. Defense Summaries, Metrics and Datasets

In this paper we investigate 9 recent defenses, Barrage of Random Transforms (BaRT) [14], End-to-End Image Compression Models (ComDefend) [13], The Odds are Odd (Odds) [17], Feature Distillation (FD) [18], Buffer Zones (BUZz) [24], Ensemble Diversity (ADP) [11], Distribution Classifier (DistC) [16], Error Correcting Output Codes (ECOC) [12] and K-Winner-Take-All (k-WTA) [15].

In Table 1. , we decompose these defenses into the underlying methods they use to try to achieve security. This is by no means the only way these defenses can be categorized and the definitions here are not absolute. We merely provide this hierarchy to provide a basic overview and show common defense themes. The defense themes are categorized as follows:

1. Multiple models—The defense uses multiple classifiers' for prediction. The classifiers outputs may be combined through averaging (i.e., ADP), majority voting (BUZz) or other methods (ECOC).
2. Fixed input transformation—A non-randomized transformation is applied to the input before classification. Examples of this include, image denoising using an autoencoder (Comdefend), JPEG compression (FD) or resizing and adding (BUZz).
3. Random input transformation—A random transformation is applied to the input before classification. For example both BaRT and DistC randomly select from multiple different image transformations to apply at run time.

4. Adversarial detection—The defense outputs a null label if the sample is considered to be adversarially manipulated. Both BUZz and Odds employ adversarial detection mechanisms.
5. Network retraining—The network is retrained to accommodate the implemented defense. For example BaRT and BUZz require network retraining to achieve acceptable clean accuracy. This is due to the significant transformations both defenses apply to the input. On the other hand, different architectures mandate the need for network retraining like in the case of ECOC, DistC and k-WTA. Note network retraining is different from adversarial training. In the case of adversarial training, it is a fundamentally different technique in the sense that it can be combined with almost every defense we study. Our interest however is not to make each defense as strong as possible. Our aim is to understand how much each defense improves security on its own. Adding in techniques beyond what the original defense focuses on is essentially adding in confounding variables. It then becomes even more difficult to determine from where security may arise. As a result, we limit the scope of our defenses to only consider retraining when required and do not consider adversarial training.
6. Architecture change—A change in the architecture which is made solely for the purposes of security. For example k-WTA uses different activation functions in the convolutional layers of a CNN. ECOC uses a different activation function on the output of the network.

3.1. *Barrage of Random Transforms*

Barrage of Random Transforms (BaRT) [14] 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 transformation groups: JPEG compression, image swirling, noise injection, Fourier transform perturbations, zooming, color space changes, histogram equalization, grayscale transformations and denoising operations.

Prior security studies: In terms of white-box analyses, the original BaRT paper tests PGD and FGSM. They also test a combined white-box attack designed to deal with randomization. This combinational white-box attack is composed of expectation over transformation [36] and backward pass differentiable approximation [9]. No analysis of the BaRT defense with black-box adversaries is done.

Why we selected it: In [19], they claim gradient free attacks (i.e., black-box attacks) most commonly fail due to randomization. Therefore BaRT is a natural candidate to test for black-box security. Also in the original paper, BaRT is only tested with ImageNet. We wanted to see if this defense could be expanded to work on other datasets.

3.2. *End-to-End Image Compression Models*

ComDefend [13] is a defense where image compression/reconstruction is done using convolutional autoencoders before classification. ComDefend consists of two modules: a compression convolutional neural network (ComCNN) and a reconstruction convolutional neural network (RecCNN). The compression network transforms the input image into a compact representation by compressing the original 24 bit pixels into compact 12 bit representations. Gaussian noise is then added to the compact representation. Decompression is then done using the reconstruction network and the final output is fed to the classifier. In this defense, retraining of the classifier on reconstructed input data is not required.

Prior security studies: White-box attacks such as FGSM, BIM and C&W are run on ComDefend. They also vary their threat model between using the l_∞ norm and l_2 norm to create white-box adversarial examples that have different constraints. No black-box attacks are ever presented for the defense.

Why we selected it: Other autoencoder defenses have fared poorly [37]. It is worth studying new autoencoder defenses to see if they work, or if they face the same vulnerabilities as older defense designs. Since ComDefend [13] does not study black-box adversaries, our analysis also provides new insight on this defense.

3.3. The Odds Are Odd

The Odds are Odd [17] 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 uses them as the input for the statistical test.

Prior security studies: In the original Odds paper, the statistical test is done in conjunction with adversarial examples generated using PGD (a white-box attack). Further white-box attacks on the Odds were done in [22]. The authors in [22] use PGD and a custom objective function to show the flaws in the statistical test under white-box adversarial model. To the best of our knowledge, no work has been done on the black-box security of the Odds defense.

Why we selected it: In [22], they mention that Odds is based on the common misconception that building a test for certain adversarial examples will then work for all adversarial examples. However, in the black-box setting this still brings up an interesting question: if the attacker is unaware of the type of test, can they still adaptively query the defense and come up with adversarial examples that circumvent the test?

3.4. Feature Distillation

Feature Distillation (FD) implements a unique JPEG compression and decompression technique to defend against adversarial examples. Standard JPEG compression/decompression preserves low frequency components. However, it is claimed in [18] that CNNs learn features which are based on high frequency components. Therefore, the authors propose a compression technique where a smaller quantization step is used for CNN accuracy-sensitive frequencies and a larger quantization step is used for the remaining frequencies. The goal of this technique is two-fold. First, by maintaining high frequency components, the defense aims to preserve clean accuracy. Second, by reducing the other frequencies, the defense tries to eliminate the noise that make samples adversarial. Note this defense does have some parameters which need to be selected through experimentation. For the sake of brevity, we provide the experiments for selecting these parameters in the Appendix A.

Prior security studies: In the original FD paper, the authors test their defense against standard white-box attacks like FGSM, BIM and C&W. They also analyze their defense against the backward pass differentiable approximation [9] white-box attack. In terms of black-box adversaries, they do test a very simple black-box attack. In this attack, samples are generated by first training a substitute model. However, this black-box adversary cannot query the defense to label its training data, making it extremely limited. Under our attack definitions, this is not an adaptive black-box attack.

Why we selected it: A common defense theme is the utilization of multiple image transformations like in the case of BaRT, BUZz and DistC. However, this requires a cost in the form of network retraining and/or clean accuracy. If a defense could use only one type of transformation (as done in FD), it may be possible to significantly reduce those costs. To the best of our knowledge, so far no single image transformation has accomplished this, which makes the investigation of FD interesting.

3.5. Buffer Zones

Buffer Zones (BUZz) employs a combination of techniques to try and achieve security. The defense is based on unanimous majority voting using multiple classifiers. Each classifier applies a different fixed secret transformation to its input. If the classifiers are unable to agree on a class label, the defense marks the input as adversarial. The authors also note that a large drop in clean accuracy is incurred due to the number of defense techniques employed.

Prior security studies: BUZz is the only defense on our list that experiments with a similar black-box adversary (one that has access to the training data and can query the defense). However, as we explain below, their study has room to further be expanded upon.

Why we selected it: We selected this defense to study because it specifically claims to deal with the exact adversarial model (adaptive black-box) that we work with. However, in their paper they only use a single strength adversary (i.e., one that uses the entire training dataset). We test across multiple strength adversaries (see Section 5) to see how well their defense holds up.

3.6. Improving Adversarial Robustness via Promoting Ensemble Diversity

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 [11]. 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.

Prior security studies: ADP has widely been studied in the context of white-box security in [11,22,23]. In the original paper in which ADP was proposed, they tested the defense against white-box attacks like FGSM, BIM, PGD, C&W and EAD. In [22], they use different attack parameters (more iterations) in order to show the defense was not as robust as previously thought. These results are further supported by white-box attacks done on ADP and reported in [23].They use FGSM, BIM and MIM (as well as others white-box attacks) in [23] to further analyze the robustness of ADP. They also test some black-box attacks on ADP in [23], but these attacks are transfer based and boundary based. They do not test our adaptive type of black-box attack in [23].

Why we selected it: It has been shown that adversarial samples can have high transferability [4]. Model black-box attacks have a basic underlying assumption: adversarial samples that fool the synthetic model will also fool the defense. ADP trains networks to specifically enhance diversity which could mitigate the transferability phenomena. If the adversarial transferability between networks is indeed really mitigated, then black-box attacks should not be effective.

3.7. Enhancing Transformation-Based Defenses against Adversarial Attacks with a Distribution Classifier

The basic idea of this defense is that if the input is adversarial, basing the predicted class on the softmax output may yield a wrong result. Instead in this defense the input is randomly transformed multiple times, to create many different inputs. Each transformed input yields a softmax output from the classifier. Prediction is then done on the distribution of the softmax outputs [16]. To classify the softmax distributions, a separate distributional classifier is trained.

Prior security studies: In [16], white-box attacks on the defense were done using methods like FGSM, IFGSM and C&W. Query only black-box attacks were also studied, but by our definition, no adaptive black-box attacks were ever considered for this defense.

Why we selected it: In [16], the defense is tested with query only black-box attacks as we previously mentioned. However, it does not test any model black-box attacks. This defense is built on [38] which was initially a promising randomization defense that was broken in [9]. Whether the combination of a new classification scheme and randomization can achieve model black-box security is an open question.

3.8. Error Correcting Output Codes

The Error Correcting Output Codes (ECOC) [12] defense uses the idea of coding theory and changes the output representation in a network to codewords. There are three main ideas of the defense. First, is the use of a special sigmoid decoding activation function instead of the softmax function. This function allocates the non-trivial volume in logit space to uncertainty. This makes the attack surface smaller to the attacker who tries to craft adversarial examples. Second, a larger Hamming distance between the codewords is used to increase the distance between two high-probability regions for a class in logit space. This forces the adversary to use larger perturbations in order to succeed. Lastly, the correlation between outputs is reduced by training an ensemble model.

Prior security studies: In [12], the authors test ECOC against white-box attacks like PGD and C&W. A further white-box analysis of ECOC is done in [22], where PGD with a custom loss function is used. Through this modified PGD, the authors in [22] are able to significantly reduce the robustness of the ECOC defense in the white-box setting. No black-box analyses of ECOC are ever considered in [22] or [12].

Why we selected it: Much like ADP, this method relies on an ensemble of models. However unlike ADP, this defense is based on coding theory and the original paper does not consider a black-box adversary. The authors in [22] were only able to come up with an effective attack on ECOC in the white-box setting. Therefore, exploring the black-box security of this defense is of interest.

3.9. k-Winner-Take-All

In k-Winner-Take-All (k-WTA) [15] a special activation function is used that is C^0 discontinuous. This activation function mitigates white-box attacks through gradient masking. The authors claim this architecture change is nearly free in terms of the drop in clean accuracy.

Prior security studies: In the original k-WTA paper [15] the authors test their defense against white-box attacks like PGD, MIM and C&W. They also test against a weak transfer based black-box attack that is not adaptive. They do not consider a black-box adversary that has access to the entire training dataset and query access like we assume in our adversarial model. Further white-box attacks against k-WTA were done in [22]. The authors in [22] used PGD with more iterations (400) and also considered a special averaging technique to better estimate the gradient of the network.

Why we selected it: The authors of the defense claim that k-WTA performs better under model black-box attacks than networks that use ReLU activation functions. If this claim is true, this would be the first defense in which gradient masking could mitigate both white-box and black-box attacks. In [22], they already showed the vulnerability of this defense to white-box attacks. Additionally, in [22] they hypothesize a black-box adversary that queries the network may work well against this defense, but do not follow up with any experiments. Therefore, this indicates k-WTA still lacks proper black-box security experiments and analyses.

3.10. Defense Metric

In this paper, our goal is to demonstrate what kind of gain in security can be achieved by using each defense against a black-box adversary. Our aim is not to claim any defense is broken. To measure the improvement in security, we use a simple metric: Defense accuracy improvement.

Defense accuracy improvement is the percent increase in correctly recognized adversarial examples gained when implementing the defense as compared to having no defense. The formula for defense accuracy improvement for the *i*th defense is defined as:

$$A_i = D_i - V \tag{1}$$

We compute the defense accuracy improvement A_i by first conducting a specific black-box attack on a vanilla network (no defense). This gives us a vanilla defense accuracy score V. The vanilla defense accuracy is the percent of adversarial examples the vanilla network correctly identifies. We run the same attack on a given defense. For the ith defense, we will obtain a defense accuracy score of D_i. By subtracting V from D_i we essentially measure how much security the defense provides as compared to not having any defense on the classifier.

For example if $V \approx 99\%$, then the defense accuracy improvement A_i can be 0, but at the very least should not be negative. If $V \approx 85\%$, then a defense accuracy improvement of 10% may be considered good. If $V \approx 40\%$, then we want at least a 25% defense accuracy improvement, for the defense to be considered effective (i.e. the attack fails more than half of the time when the defense is implemented). While sometimes an improvement is not possible (e.g. when $V \approx 99\%$) there are many cases where attacks works well on the undefended network and hence there are places where large improvements can be made. Note to make these comparisons as precise as possible, almost every defense is built with the same CNN architecture. Exceptions to this occur in some cases, which we fully explain in the Appendix A.

3.11. Datasets

In this paper, we test the defenses using two distinct datasets, CIFAR-10 [39] and Fashion-MNIST [40]. CIFAR-10 is a dataset comprised of 50,000 training images and 10,000 testing images. Each image is 32 × 32 × 3 (a 32 × 32 color image) and belongs to 1 of 10 classes. The 10 classes in CIFAR-10 are airplane, car, bird, cat, deer, dog, frog, horse, ship and truck. Fashion-MNIST is a 10 class dataset with 60,000 training images and 10,000 test images. Each image in Fashion-MNIST is 28 × 28 (grayscale image). The classes in Fashion-MNIST correspond to t-shirt, trouser, pullover, dress, coat, sandal, shirt, sneaker, bag and ankle boot.

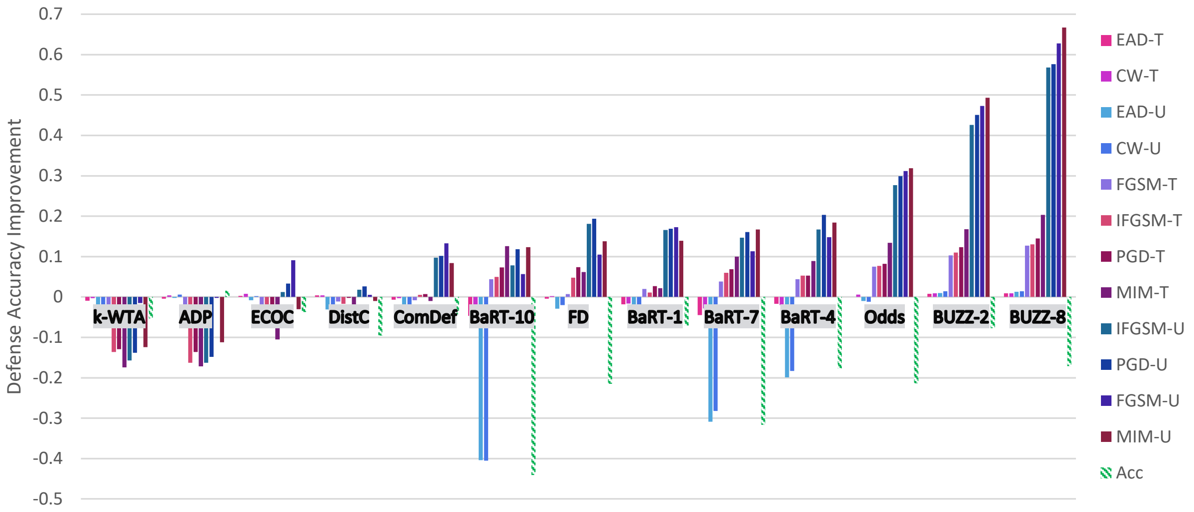
Why we selected them: We chose the CIFAR-10 defense because many of the existing defenses had already been configured with this dataset. Those defenses already configured for CIFAR-10 include ComDefend, Odds, BUZz, ADP, ECOC, the distribution classifier defense and k-WTA. We also chose CIFAR-10 because it is a fundamentally challenging dataset. CNN configurations like ResNet do not often achieve above 94% accuracy on this dataset [41]. In a similar vein, defenses often incur a large drop in clean accuracy on CIFAR-10 (which we will see later in our experiments with BUZz and BaRT for example). This is because the amount of pixels that can be manipulated without hurting classification accuracy is limited. For CIFAR-10, each image only has in total 1024 pixels. This is relatively small when compared to a dataset like ImageNet [42], where images are usually 224 × 224 × 3 for a total of 50,176 pixels (49 times more pixels than CIFAR-10 images). In short, we chose CIFAR-10 as it is a challenging dataset for adversarial machine learning and many of the defenses we test were already configured with this dataset in mind.

For Fashion-MNIST, we primarily chose it for two main reasons. First, we wanted to avoid a trivial dataset on which all defenses might perform well. For example, CNNs can already achieve a clean accuracy of 99.7% on a dataset like MNIST [40]. Testing on such types of datasets would not work towards the main aim of our paper, which is to distinguish defenses that perform significantly better in terms of security and clean accuracy. The second reason we chose Fashion-MNIST is for its differences from CIFAR-10. Specifically, Fashion-MNIST is a non-color dataset and contains very different types of images than CIFAR-10. In addition, many of the defenses we tested were not originally designed for Fashion-MNIST. This brings up an interesting question, can previously proposed defenses be readily adapted to work with different datasets. To summarize, we chose Fashion-MNIST for its difficult to learn and its differences from CIFAR-10.

4. Principal Experimental Results

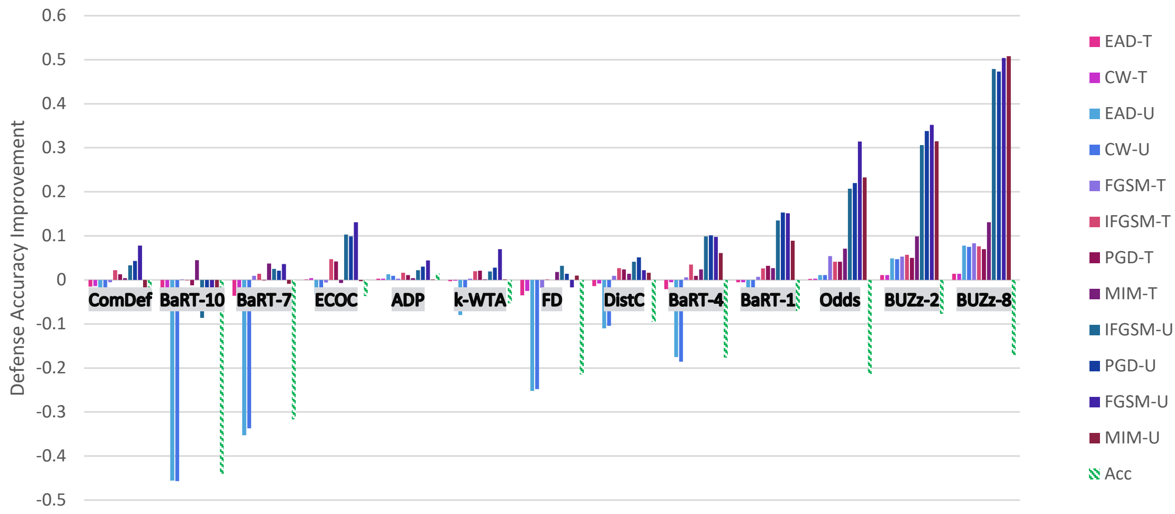
In this section, we conduct experiments to test the black-box security of the 9 defenses. We measure the results using the metric defense accuracy improvement (see Section 3.10). For each defense, we test it under a pure black-box adversary, and five different strength adaptive black-box adversaries. The strength of the adaptive black-box adversary is determined by how much of the original training dataset they are given access to (either 100%, 75%, 50%, 25% or 1%). For every adversary, once the synthetic model is trained, we use 6 different methods (FGSM [3], BIM [31], MIM [32], PGD [27], C&W [28] and EAD [33]) to generate adversarial examples. We test both targeted and untargeted styles of attack. In these experiments we use the l_∞ norm with maximum perturbation $\epsilon = 0.05$ for CIFAR-10 and $\epsilon = 0.1$ for Fashion-MNIST. Further attack details can be found in our Appendix A.

Before going into a thorough analysis of our results, we briefly introduce the figures and tables that show our experimental results. Figures 1 and 2 illustrate the defense accuracy improvement of all the defenses under a 100% strength adaptive black-box adversary (Figure 1) and a pure black-box adversary (Figure 2) for the CIFAR-10 dataset. Likewise, for Fashion-MNIST, Figure 3 shows the defense accuracy improvement under a 100% strength adaptive black-box adversary and Figure 4 shows the defense accuracy improvement under a pure black-box adversary. For each of these figures, we report the vanilla accuracy numbers in a chart below the graph. Figure 5 through Figure 6 show the relationship between the defense accuracy and the strength of the adversary (how much training data the adversary has access to). Figure 5 through Figure 6 show this relationship for each defense, on both CIFAR-10 and Fashion-MNIST. The corresponding values for the figures are given in Table A4 through Table A15.



	EAD-T	CW-T	EAD-U	CW-U	FGSM-T	IFGSM-T	PGD-T	MIM-T	IFGSM-U	PGD-U	FGSM-U	MIM-U	Acc
Vanilla	0.991	0.991	0.987	0.986	0.866	0.861	0.848	0.777	0.387	0.374	0.334	0.259	0.9278

Figure 1. CIFAR-10 adaptive black-box attack on each defense. Here the U/T refers to whether the attack is untargeted/targeted. Negative values means the defense performs worse than the no defense (vanilla) case. The Acc value refers to the drop in clean accuracy incurred by implementing the defense. The chart below the graph gives the vanilla defense accuracy numbers.



	EAD-T	CW-T	EAD-U	CW-U	FGSM-T	IFGSM-T	PGD-T	MIM-T	IFGSM-U	PGD-U	FGSM-U	MIM-U	Acc
Vanilla	0.984	0.984	0.919	0.923	0.902	0.917	0.924	0.853	0.453	0.455	0.443	0.384	0.9278

Figure 2. CIFAR-10 pure black-box attack on each defense. Here the U/T refers to whether the attack is untargeted/targeted. Negative values means the defense performs worse than the no defense (vanilla) case. The Acc value refers to the drop in clean accuracy incurred by implementing the defense. The chart below the graph gives the vanilla defense accuracy numbers. For all the experimental numbers see Table A4.

Considering the range of our experiments (9 defenses, 6 adversarial models, 6 methods to generate adversarial samples and 2 datasets), it is infeasible to report all the results and experimental details in just one section. Instead, we organize our experimental analysis as follows. In this section, we present the most pertinent results in Figures 1 and 3 and give the principal takeaways. For readers interested in a specific defense or attack results, in Section 5 we give a comprehensive break down of the results for each defense, dataset and attack. For anyone wishing to recreate our experimental results, we give complete implementation details for every attack and defense in the Appendix A.

Principal Results

1. **Marginal or negligible improvements over no defense**: Figure 1 shows the defense results for CIFAR-10 with a 100% strength adaptive black-box adversary. In this figure, we can clearly see 7 out of 9 defenses give marginal (less than 25%) increases in defense accuracy for any attack. BUZz and the Odds defense are the only ones to break this trend for CIFAR-10. For example, BUZz-8 gives a 66.7% defense accuracy improvement for the untargeted MIM attack. Odds gives a 31.9% defense accuracy improvement for the untargeted MIM attack. Likewise, for Fashion-MNIST again, 7 out of 9 defenses give only marginal improvements (see Figure 3). BUZz and BaRT are the exceptions for this dataset.

2. **Security is not free (yet)**: Thus far, no defense we experimented with that offers significant (greater than 25% increase) improvements comes for free. For example, consider the defenses that give significant defense accuracy improvements. BUZz-8 drops the clean accuracy by 17% for CIFAR-10. BaRT-6 drops the clean accuracy by 15% for Fashion-MNIST. As defenses improve, we expect to see this trade-off between clean accuracy and security become more favorable. However, our experiments show we have not reached this point with the current defenses.

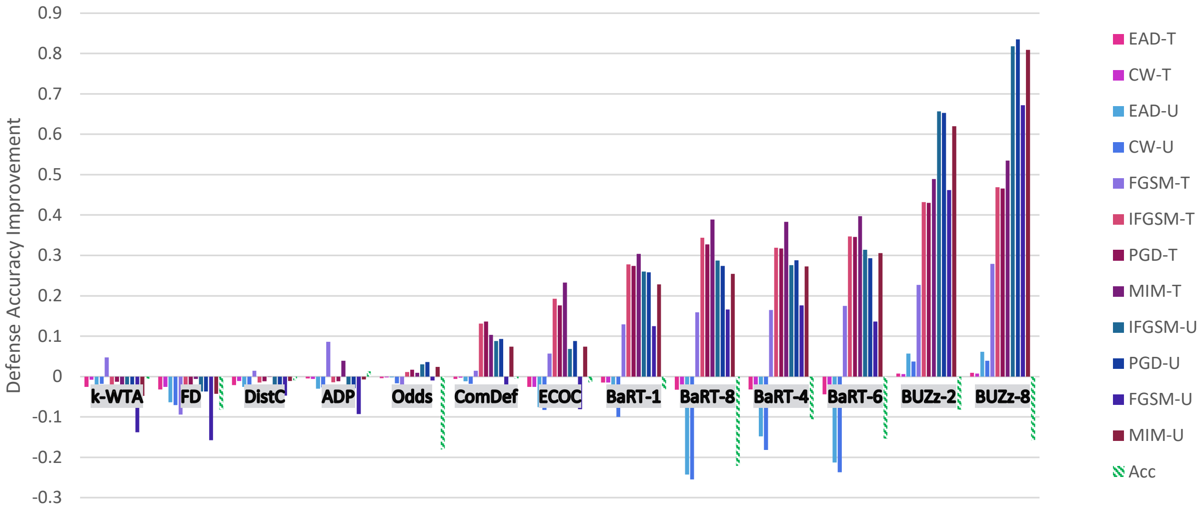
3. **Common defense mechanisms**: It is difficult to decisively prove any one defense mechanism guarantees security. However, among the defenses that provide more than marginal improvements (Odds, BUZz and BaRT), we do see common defense trends. Both Odds and BUZz use adversarial detection. This indirectly deprives the adaptive black-box adversary of training data. When an input sample is marked as adversarial, the black-box attacker cannot use it to train the synthetic model. This is because the synthetic model has

no adversarial class label. It is worth noting that in the Appendix A, we also argue why a synthetic model should not be trained to output an adversarial class label.

Along similar lines, both BaRT and BUZz offer significant defense accuracy improvements for Fashion-MNIST. Both employ image transformations so jarring that the classifier must be retrained on transformed data. The experiments show that increasing the number of the transformations only increases security up to a certain point though. For example, BaRT-8 does not perform better than BaRT defenses that use less image transformations (see BaRT-6 and BaRT-4 in Figure 3).

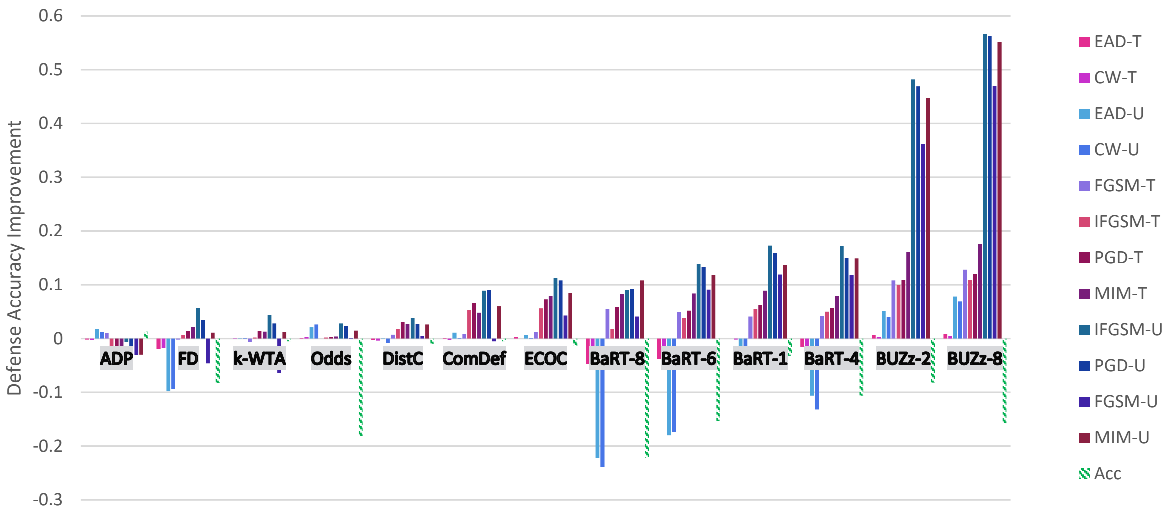
4. **Adaptive and pure black-box follow similar trends.** In Figures 2 and 4 we show results for the pure black-box attack for CIFAR-10 and Fashion-MNIST. Just like for the adaptive black-box attack, we see similar trends in terms of which defenses provide the highest security gains. For CIFAR-10, the defenses that give at least 25% greater defense accuracy than the vanilla defense include BUZz and Odds. For Fashion-MNIST, the only defense that gives this significant improvement is BUZz.

5. **Future defense analyses should be broad**: From our first point in this subsection, it is clear that a majority of these defenses give marginal improvements or less. This brings up an important question, what impact does our security study have for future defenses? The main lesson is future defense designers need to test against a broad spectrum of attacks. From the literature, we see the majority of the 9 defenses already considered white-box attacks like PGD or FGSM and some weak black-box attacks. However, in the face of adaptive attacks, these defenses perform poorly. Future defense analyses at the very least need white-box attacks *and* adaptive black-box attacks. By providing our paper's results and code we hope to help future defense designers perform these analyses and advance the field of adversarial machine learning.



	EAD-T	CW-T	EAD-U	CW-U	FGSM-T	IFGSM-T	PGD-T	MIM-T	IFGSM-U	PGD-U	FGSM-U	MIM-U	Acc
Vanilla	0.991	0.993	0.939	0.961	0.707	0.529	0.531	0.46	0.123	0.118	0.234	0.111	0.9356

Figure 3. Fashion-MNIST adaptive black-box attack on each defense. Here the U/T refers to whether the attack is untargeted/targeted. Negative values means the defense performs worse than the no defense (vanilla) case. The Acc value refers to the drop in clean accuracy incurred by implementing the defense. The chart below the graph gives the vanilla defense accuracy numbers.



	EAD-T	CW-T	EAD-U	CW-U	FGSM-T	IFGSM-T	PGD-T	MIM-T	IFGSM-U	PGD-U	FGSM-U	MIM-U	Acc
Vanilla	0.992	0.995	0.905	0.914	0.865	0.889	0.879	0.817	0.363	0.374	0.429	0.351	0.9356

Figure 4. Fashion-MNIST pure black-box attack on each defense. Here the U/T refers to whether the attack is untargeted/targeted. Negative values means the defense performs worse than the no defense (vanilla) case. The Acc value refers to the drop in clean accuracy incurred by implementing the defense. The chart below the graph gives the vanilla defense accuracy numbers. For all the experimental numbers see Table A10.

5. Individualized Experimental Defense Results

In the previous section, we discussed the overarching themes represented in the adaptive black-box attack experimental results. In this section, we take a more fine grained approach and consider each defense individually.

Both the 100% adaptive black-box and pure black-box attack have access to the entire original training dataset. The difference between them lies in the fact that the adaptive black-box attack can generate synthetic data, and label the training data by querying the defense. Since both attacks are similar in terms of how much data they start with, a question arises. How effective is the attack if the attacker doesn't have access to the full training dataset? In the following subsections, we seek to answer that question by considering each defense under a variable strength adversary in the adaptive black-box setting. Specifically we test out adversaries that can query the defense but only have 75%, 50%, 25% or 1% of the original training dataset.

To simplify things with the variable strength adaptive black-box adversary, we only consider the untargeted MIM attack for generating adversarial examples. We use the MIM attack because it is the best performing attack on the vanilla (no defense) network for both datasets. Therefore, this attack represents the place where the most improvement in security can be made. For the sake of completeness, we do report all the defense accuracies for all six types of attacks for the variable strength adaptive black-box adversaries in the tables at the end of this section.

After discussing defense results, we also present brief experiment and discussion on why the adaptive black-box attack is actually considered *adaptive*. We do this by comparing the attack success rate of the adaptive attack to the non-adaptive pure black-box attack while simultaneously fixing the underlying method to generate adversarial examples, fixing the dataset and fixing the amount of training data available to the attacker.

5.1. Barrage of Random Transforms Analysis

The adaptive black-box attack with variable strength for BaRT defenses is shown in Figure 5. There are several interesting observations that can be made about this defense. First, for CIFAR-10, the maximum transformation defense (BaRT-10) actually performs worse than the vanilla defense in most cases. BaRT-1, BaRT-4 and BaRT-7 perform approximately the same as the vanilla defense. These statements hold except for the 100% strength adaptive black-box adversary. Here, all BaRT defenses show a 12% or greater improvement over the vanilla defense.

Where as the performance of BaRT is rather varied for CIFAR-10, for Fashion-MNIST this is not the case. All BaRT defenses show improvement for the MIM attack for adversaries with 25% strength or greater.

When examining the results of BaRT on CIFAR-10 and Fashion-MNIST, we see a clear discrepancy in performance. One possible explanation is as follows: the image transformations in a defense must be selected in a way that does not greatly impact the original clean accuracy of the classifier. In the case of BaRT-10 (the maximum number of transformations) for CIFAR-10, it performs much worse than the vanilla case. However, BaRT-8 for Fashion-MNIST (again the maximum number of transformations) performs much better than the vanilla case. If we look at the clean accuracy of BaRT-10, it is approximately 48% on CIFAR-10. This is a drop of more than 40% as compared to the vanilla clean accuracy. For BaRT-8, the clean accuracy is approximately 72% on Fashion-MNIST which is a drop of about 21%. Here we do not use precise numbers when describing the clean accuracy because as a randomized defense, the clean accuracy may drop or rise a few percentage points every time the test set is evaluated.

From the above stated results, we can make the following conclusion: A defense that employs random image transformations cannot be applied naively to every dataset. The set of image transformations must be selected per dataset in such a manner that the clean accuracy is not drastically impacted. In this sense, while random image transformations may be a promising defense direction, it seems they may need to be designed on a per dataset basis.

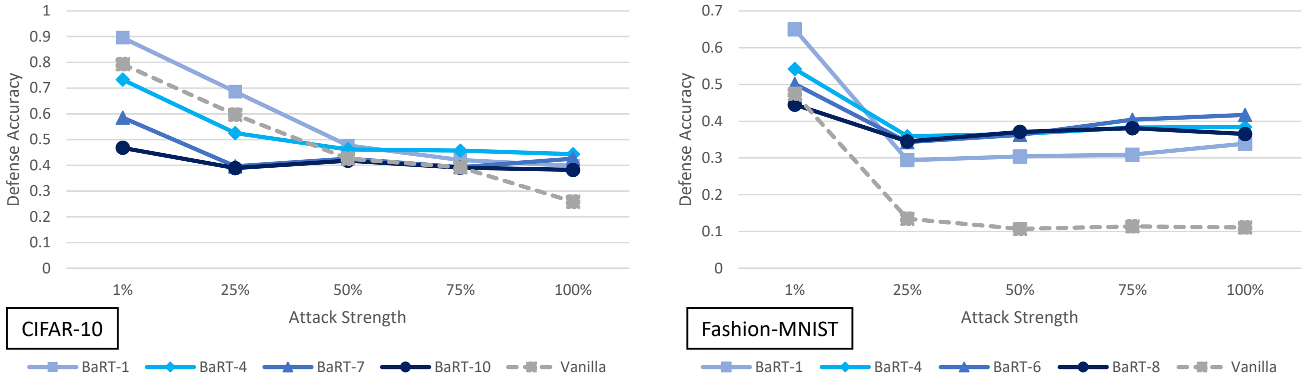


Figure 5. Defense accuracy of barrage of random transforms defense on various strength adaptive black-box adversaries for CIFAR-10 and Fashion-MNIST. The defense accuracy in these graphs is measured on the adversarial samples generated from the untargeted MIM adaptive black-box attack. The % strength of the adversary corresponds to what percent of the original training dataset the adversary has access to. For full experimental numbers for CIFAR-10, see Table A5 through Table A9. For full experimental numbers for Fashion-MNIST, see Table A11 through Table A15.

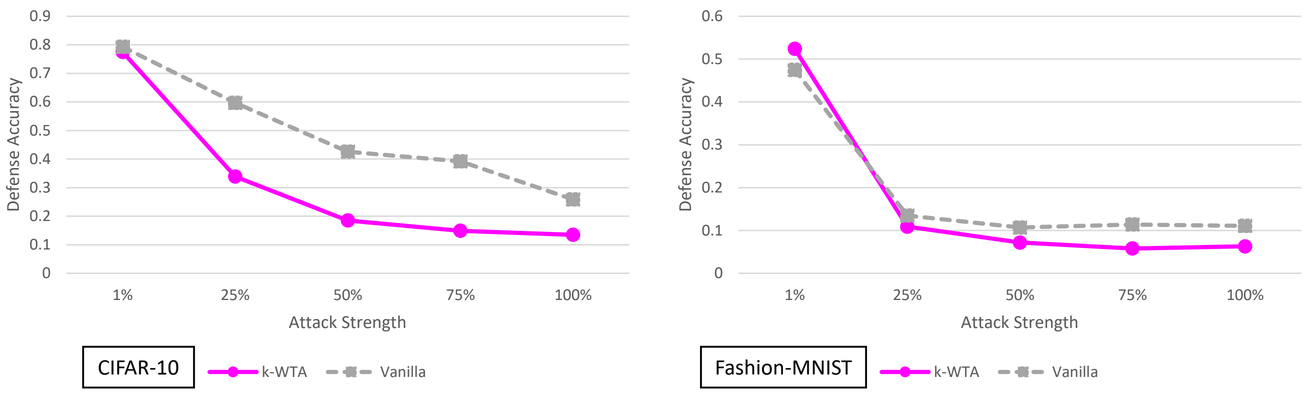


Figure 6. Defense accuracy of the k-Winners-Take-All defense on various strength adaptive black-box adversaries for CIFAR-10 and Fashion-MNIST. The defense accuracy in these graphs is measured on the adversarial samples generated from the untargeted MIM adaptive black-box attack. The % strength of the adversary corresponds to what percent of the original training dataset the adversary has access to. For full experimental numbers for CIFAR-10, see Table A5 through Table A9. For full experimental numbers for Fashion-MNIST, see Table A11 through Table A15.

5.2. *End-to-End Image Compression Models Analysis*

The adaptive black-box attack with variable strength for ComDefend is shown in Figure 7. For CIFAR-10, we see the defense performs very close to the vanilla network (and sometimes slightly worse). On the other hand, for Fashion-MNIST, the defense does offer a modest average defense accuracy improvement of 8.84% across all adaptive black-box adversarial models.

In terms of understanding the performance of ComDefend, it is important to note the following: In general it has been shown that more complex architectures (e.g., deeper networks) can better resist transfer based adversarial attacks [10]. In essence an autoencoder/decoder setup can be viewed as additional layers in the CNN and hence a more complex model. Although this concept was shown for ImageNet [10], it may be a phenomena that occurs in other datasets as well.

This more complex model can partially explain why ComDefend slightly outperforms the vanilla defense in most cases. In short, a slightly more complex model is slightly more difficult to learn and attack. Of course this begs the question, if a more complex model yields more security, why does the model complexity even have come from an autoencoder/decoder? Why not use ResNet164 or ResNet1001?

These are all valid questions which are possible directions of future studies. While ComDefend itself does not yield significant (greater than 25%) improvements in security, it does bring up an interesting question: Under a black-box adversarial model, to what extent can increasing model complexity also increase defense accuracy? We leave this as an open question for possible future work.

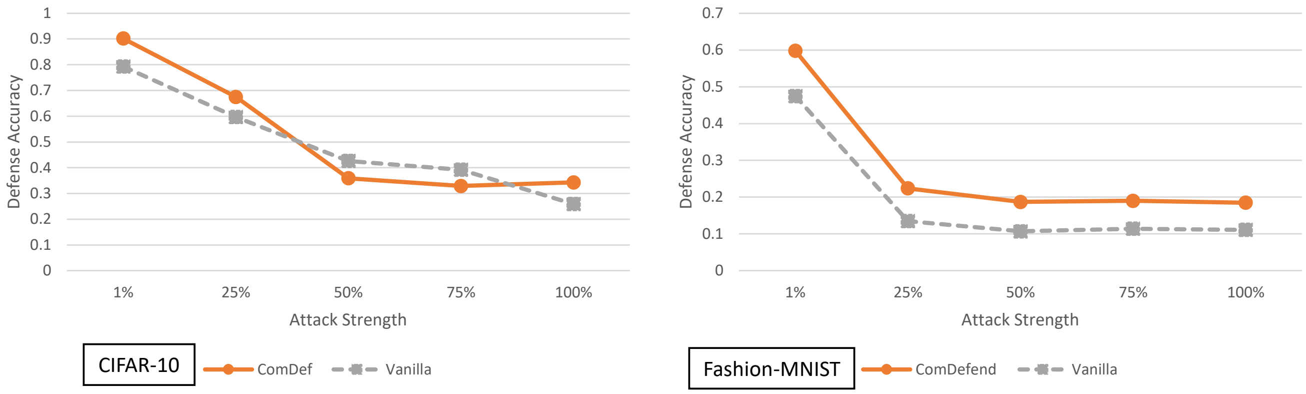


Figure 7. Defense accuracy of ComDefend on various strength adaptive black-box adversaries for CIFAR-10 and Fashion-MNIST. The defense accuracy in these graphs is measured on the adversarial samples generated from the untargeted MIM adaptive black-box attack. The % strength of the adversary corresponds to what percent of the original training dataset the adversary has access to. For full experimental numbers for CIFAR-10, see Table A5 through Table A9. For full experimental numbers for Fashion-MNIST, see Table A11 through Table A15.

5.3. The Odds Are Odd Analysis

In Figure 8, the adaptive black-box attack with different strengths is shown for the Odds defense. For CIFAR-10 the Odds has an average improvement of 19.3% across all adversarial models. However, for Fashion-MNIST the average improvement over the vanilla model is only 2.32%. As previously stated, this defense relies on the underlying assumption that creating a test for one set of adversarial examples will then generalize to all adversarial examples.

When the test used in the Odds does provide security improvements (as in the case for CIFAR-10), it does highlight one important point. If the defense can mark some samples as adversarial, it is possible to deprive the adaptive black-box adversary of data to train the synthetic model. This in turn weakens the overall effectiveness of the adaptive black-box attack. We stress however that this occurs only when the test is accurate and does not greatly hurt the clean prediction accuracy of the classifier.

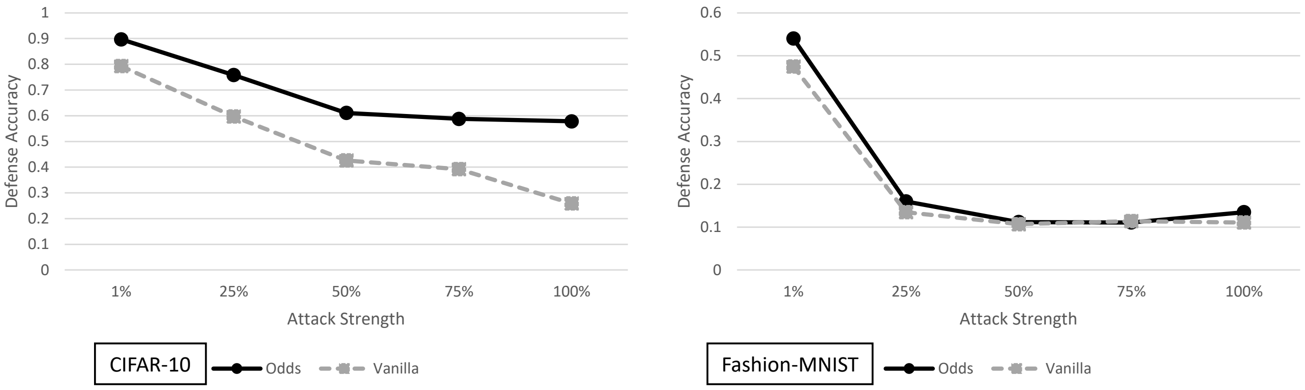


Figure 8. Defense accuracy of the odds defense on various strength adaptive black-box adversaries for CIFAR-10 and Fashion-MNIST. The defense accuracy in these graphs is measured on the adversarial samples generated from the untargeted MIM adaptive black-box attack. The % strength of the adversary corresponds to what percent of the original training dataset the adversary has access to. For full experimental numbers for CIFAR-10, see Table A5 through Table A9. For full experimental numbers for Fashion-MNIST, see Table A11 through Table A15.

5.4. Feature Distillation Analysis

Figure 9 shows the adaptive black-box with a variable strength adversary for the feature distillation defense. In general feature distillation performs worse than the vanilla network for all Fashion-MNIST adversaries. It performs worse or roughly the same for all CIFAR-10 adversaries, except for the 100% case where it shows a marginal improvement of 13.8%.

In the original feature distillation paper the authors claim that they test a black-box attack. However, our understanding of their black-box attack experiment is that the synthetic model used in their experiment was not trained in an adaptive way. To be specific, the adversary they use does not have query access to the defense. Hence, this may explain why when an adaptive adversary is considered, the feature distillation defense performs roughly the same as the vanilla network.

As we stated in the main paper, it seems unlikely a single image transformation would be capable of providing significant defense accuracy improvements. Thus far, the experiments on feature distillation support that claim for the JPEG compression/decompression transformation. The study of this image transformation and the defense are still very useful. The idea of JPEG compression/decompression when combined with other image transformations may still provide a viable defense, similar to what is done in BaRT.

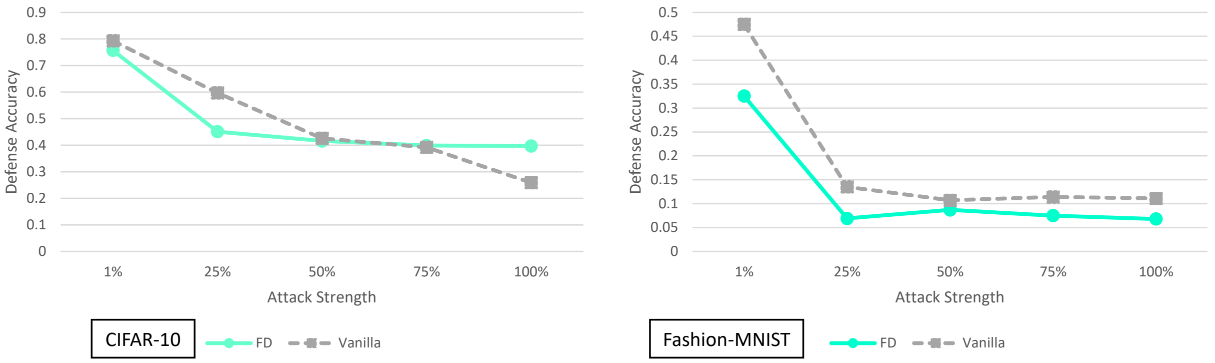


Figure 9. Defense accuracy of feature distillation on various strength adaptive black-box adversaries for CIFAR-10 and Fashion-MNIST. The defense accuracy in these graphs is measured on the adversarial samples generated from the untargeted MIM adaptive black-box attack. The % strength of the adversary corresponds to what percent of the original training dataset the adversary has access to. For full experimental numbers for CIFAR-10, see Table A5 through Table A9. For full experimental numbers for Fashion-MNIST, see Table A11 through Table A15.

5.5. Buffer Zones Analysis

The results for the buffer zone defense in regards to the adaptive black-box variable strength adversary are given in Figure 10. For all adversaries, and all datasets we see an improvement over the vanilla model. This improvement is quite small for the 1% adversary for the CIFAR-10 dataset at only a 10.3% increase in defense accuracy for BUZz-2. However, the increases are quite large for stronger adversaries. For example, the difference between the BUZz-8 and vanilla model for the Fashion-MNIST full strength adversary is 80.9%.

As we stated earlier, BUZz is one of the defenses that does provide more than marginal improvements in defense accuracy. This improvement comes at a cost in clean accuracy however. To illustrate: BUZz-8 has a drop of 17.13% and 15.77% in clean testing accuracy for CIFAR-10 and Fashion-MNIST respectively. An ideal defense is one in which the clean accuracy is not greatly impacted. In this regard, BUZz still leaves much room for improvement. The overall idea presented in BUZz of combining adversarial detection and image transformations does give some indications of where future black-box security may lie, if these methods can be modified to better preserve clean accuracy.

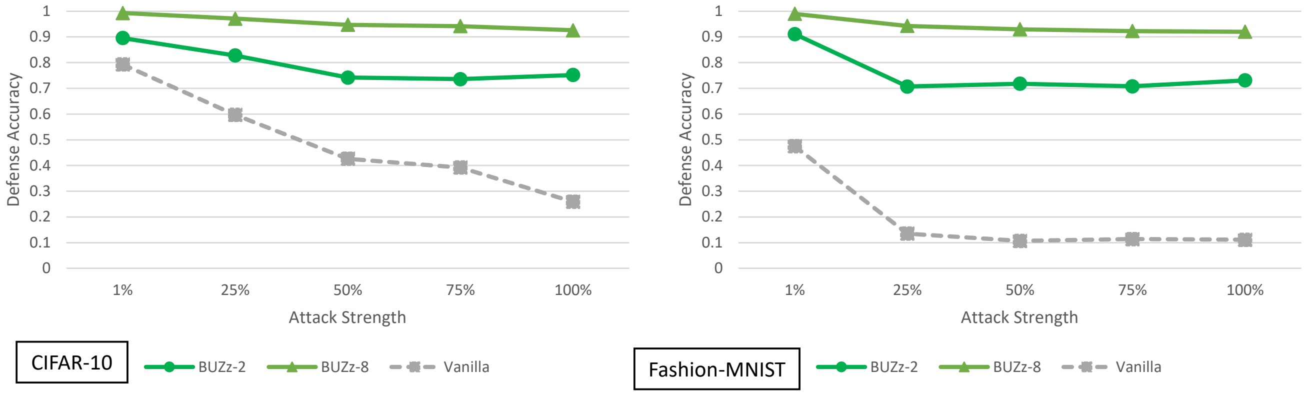


Figure 10. Defense accuracy of the buffer zones defense on various strength adaptive black-box adversaries for CIFAR-10 and Fashion-MNIST. The defense accuracy in these graphs is measured on the adversarial samples generated from the untargeted MIM adaptive black-box attack. The % strength of the adversary corresponds to what percent of the original training dataset the adversary has access to. For full experimental numbers for CIFAR-10, see Table A5 through Table A9. For full experimental numbers for Fashion-MNIST, see Table A11 through Table A15.

5.6. Improving Adversarial Robustness via Promoting Ensemble Diversity Analysis

The ADP defense and its performance under various strength adaptive black-box adversaries is shown in Figure 11. For CIFAR-10, the defense does slightly worse than the vanilla model. For Fashion-MNIST, the defense does almost the same as the vanilla model.

It has also been shown before in [24] that using multiple vanilla networks does not yield significant security improvements against a black-box adversary. The adaptive black-box attacks presented here support these claims when it comes to the ADP defense. At this time we do not have an adequate explanation as to why the ADP defense performs worse on CIFAR-10 given its clean accuracy is actually slightly higher than the vanilla model. We would expect slightly higher clean accuracy would result in slightly higher defense accuracy but this is not the case. Overall though, we do not see significant improvements in defense accuracy when implementing ADP against adaptive black-box adversaries of varying strengths for CIFAR-10 and Fashion-MNIST.

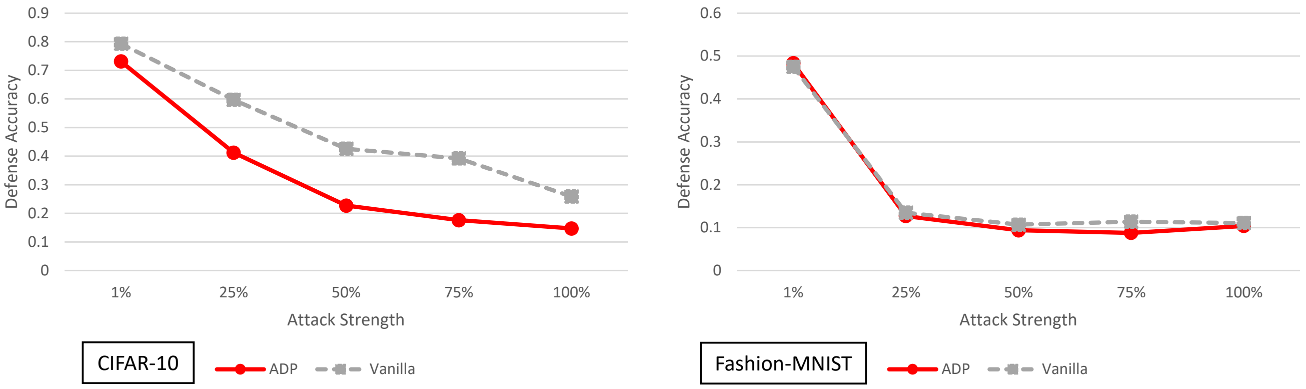


Figure 11. Defense accuracy of the ensemble diversity defense on various strength adaptive black-box adversaries for CIFAR-10 and Fashion-MNIST. The defense accuracy in these graphs is measured on the adversarial samples generated from the untargeted MIM adaptive black-box attack. The % strength of the adversary corresponds to what percent of the original training dataset the adversary has access to. For full experimental numbers for CIFAR-10, see Table A5 through Table A9. For full experimental numbers for Fashion-MNIST, see Table A11 through Table A15.

5.7. Enhancing Transformation-Based Defenses against Adversarial Attacks with a Distribution Classifier Analysis

The distribution classifier defense [16] results for adaptive black-box adversaries of varying strength are shown in Figure 12. This defense does not perform significantly better than the vanilla model for either CIFAR-10 or Fashion-MNIST. This defense employs randomized image transformations, just like BaRT. However, unlike BaRT, there is no clear improvement in defense accuracy. We can attribute this to two main reasons. First, the number of transformations in BaRT are significantly larger (i.e., 10 different image transformation groups in CIFAR-10, 8 different image transformation groups in Fashion-MNIST). In the distribution classifier defense, only resizing and zero padding transformations are used. Second, BaRT requires retraining the entire classifier to accommodate the transformations. This means all parts of the network from the convolutional layers, to the feed forward classifier are modified (retrained). The distribution classifier defense only retrains the final classifier after the soft-max output. This means the feature extraction layers (convolutional layers) between the vanilla model and the distributional classifier are virtually unchanged. If two networks have the same convolutional layers with the same weights, it is not surprising that the experiments show that they have similar defense accuracies.

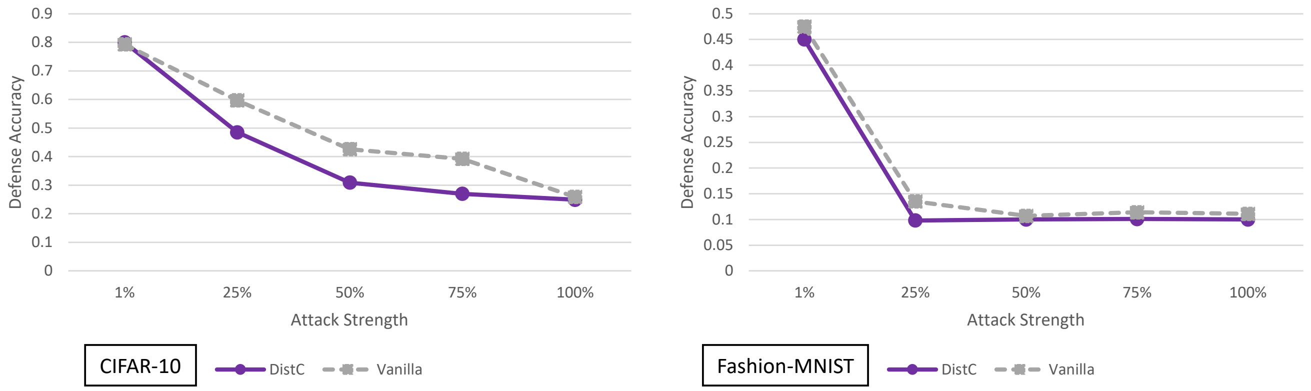


Figure 12. Defense accuracy of the distribution classifier defense on various strength adaptive black-box adversaries for CIFAR-10 and Fashion-MNIST. The defense accuracy in these graphs is measured on the adversarial samples generated from the untargeted MIM adaptive black-box attack. The % strength of the adversary corresponds to what percent of the original training dataset the adversary has access to. For full experimental numbers for CIFAR-10, see Table A5 through Table A9. For full experimental numbers for Fashion-MNIST, see Table A11 through Table A15.

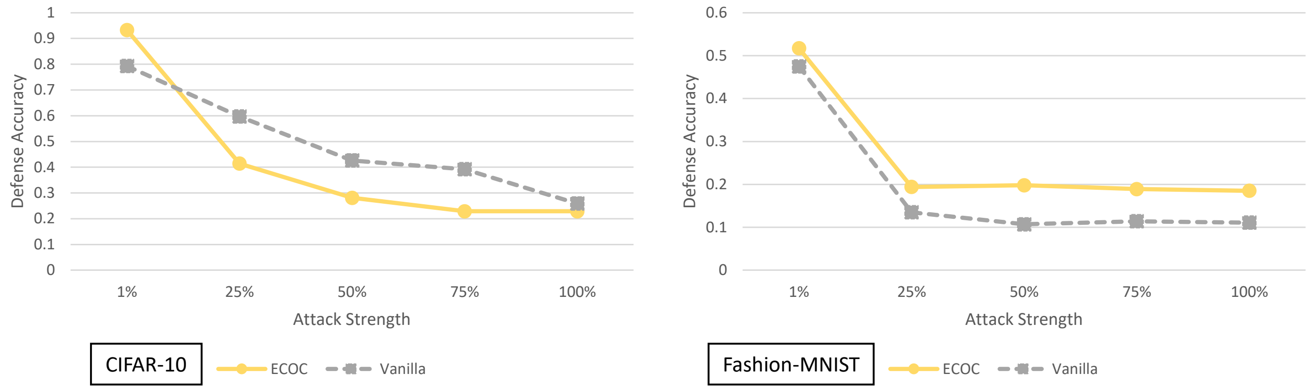


Figure 13. Defense accuracy of the error correcting output code defense on various strength adaptive black-box adversaries for CIFAR-10 and Fashion-MNIST. The defense accuracy in these graphs is measured on the adversarial samples generated from the untargeted MIM adaptive black-box attack. The % strength of the adversary corresponds to what percent of the original training dataset the adversary has access to. For full experimental numbers for CIFAR-10, see Table A5 through Table A9. For full experimental numbers for Fashion-MNIST, see Table A11 through Table A15.

5.8. Error Correcting Output Codes Analysis

In Figure 13, we show the ECOC defense for the adaptive black-box adversaries with varied strength. For CIFAR-10, ECOC performs worse than the vanilla defense in all cases except for the 1% strength adversary. For Fashion-MNIST, the ECOC defense performs only slightly better than the vanilla model. ECOC performs 6.82% greater in terms of defense accuracy on average when considering all the different strength adaptive black-box adversaries for Fashion-MNIST. In general, we don't see significant improvements (greater than 25% increases) in defense accuracy when implementing ECOC.

5.9. k-Winner-Take-All Analysis

The results for the adaptive black-box variable strength adversary for the k-WTA defense are given in Figure 6. We can see that the k-WTA defense performs approximately the same or slightly worse than the vanilla model in almost all cases.

The slightly worse performance on CIFAR-10 can be attributed to the fact that the clean accuracy of the k-WTA ResNet56 is slightly lower than the clean accuracy of the vanilla model. We go into detailed explanations about the lower accuracy in the Appendix A. In short, the k-WTA defense is implemented in PyTorch while the vanilla ResNet56 is implemented in Keras. The slightly lower accuracy is due to implementation differences between Keras and PyTorch. It is not necessarily a direct product of the defense.

Regardless of the slight clean accuracy discrepancies, we see that this defense does not offer any significant improvements over the vanilla defense. From a black-box attacker perspective, this makes sense. Replacing an activation function in the network while still making it have almost identical performance on clean images should not yield security. The only exception to this would be if the architecture change fundamentally alters the way the image is processed in the CNN. In the case of k-WTA, the experiments support the hypothesis that this is not the case.

5.10. On the Adaptability of the Adaptive Black-Box Attack

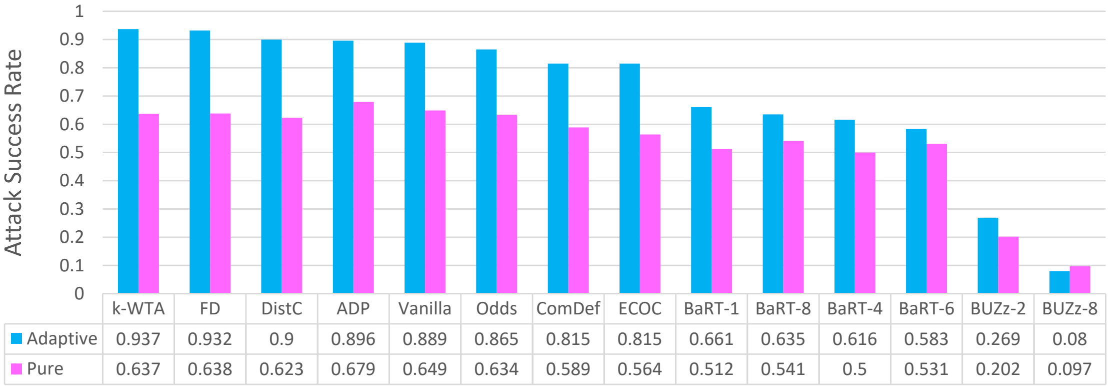
The adaptive black-box is aptly named because it *adapts* to the defense it is attacking. It does this by training the synthetic model on the output labels from the defense, as opposed to using the original training data labels. While this claim is intuitive in this subsection we give experimental proof to support our claim.

To show the advantage of the adaptive black-box attack, we compare it to the pure black-box attack (which is non-adaptive). The pure black-box attack is not considered adaptive because the adversarial examples generated in the pure black-box attack are defense agnostic. Specifically, this means the *same* set of adversarial examples are used, regardless of which defense is being attacked.

To compare attack results, we setup the following simple experiment: we use the Fashion-MNIST dataset, assuming an untargeted attack with respect to the l_∞ norm and maximum perturbation $\epsilon = 0.1$. We give both attacks access to 100% of the training data and we use the MIM method for generating adversarial examples once the synthetic model in each attack has been trained. For both the pure and adaptive black-box attack, we use the same synthetic model (see the Appendix A for further model details).

Having fixed the dataset, attack generation method, synthetic model and the amount of data available to the attacker, we report the attack success rate on vanilla classifiers and all 9 defenses in Figure 14. For each defense, we use 1000 clean examples and measure the percent of adversarial examples created from the clean examples that are misclassified. For almost every case, we can see that the adaptive black-box attack does better than the pure black-box attack, demonstrating the notion of adaptability. For example, the adaptive black-box attack has a 20% or greater improvement in attack success rate over the pure black-box attack on k-WTA, FD, DistC, ADP, Odds, ComDefend and ECOC. It should be worth noting the improvement is smaller but still there for all the BaRT defenses and every BUZZ defense except for BUZz-8. We conjecture this may be due to the fact the adaptive black-box attack does not train on null label data, something that the BUZz-8 defense outputs. Hence the lack of training data when attacking the BUZz-8 defense may cause the attack to be weaker. We discuss this notion of adaptive attacks on null label defenses in greater detail in the Appendix A.

Overall, our results in this subsection give strong experimental evidence to support the adaptability claim for the adaptive black-box attack. It can clearly be seen that in almost every case, the adaptive attack is able to make use of querying the defense to produce a higher attack success rate. When compared to a static black-box attack like the pure black-box attack, the adaptive black-box attack does better against the majority of the defenses analyzed in this work.



	k-WTA	FD	DistC	ADP	Vanilla	Odds	ComDef	ECOC	BaRT-1	BaRT-8	BaRT-4	BaRT-6	BUZz-2	BUZz-8
Adaptive	0.937	0.932	0.9	0.896	0.889	0.865	0.815	0.815	0.661	0.635	0.616	0.583	0.269	0.08
Pure	0.637	0.638	0.623	0.679	0.649	0.634	0.589	0.564	0.512	0.541	0.5	0.531	0.202	0.097

Figure 14. Adaptive black-box attack (100% strength) vs pure black-box attack on the vanilla classifier and all 9 defenses for Fashion-MNIST. It can clearly be seen that in almost every case, the adaptive black-box attack outperforms the pure black-box attack.

6. Conclusions

In this paper, we investigated and rigorously experimented with adaptive black-box attacks on recent defenses. Our paper's results span nine defenses, two adversarial models, six different attacks, and two datasets. From our vast set of experiments, we derive several principal results to advance the field of adversarial machine learning. We show that most defenses (7 out of 9 for each dataset) offer less than a 25% improvement in defense accuracy for an adaptive black-box adversary. We demonstrate that currently no defense gives significant black-box robustness without sustaining a drop in clean accuracy. While the defenses we cover generally provide marginal or less than marginal robustness, there are several common defense trends among the stronger defenses we analyzed. The common effective defense trends include using detection methods to mark suspicious samples as adversarial and using image transformations so large in magnitude that retraining of the classifier is required. Lastly, our experiments highlight the need for proper black-box attack testing. Simply building white-box defenses and only testing against white-box attacks can result in highly misleading claims about robustness. Overall, we complete the security picture for currently proposed defense with our experiments and give future defense designers insight and direction with our analyses.

Author Contributions: Conceptualization, investigation, experimentation, writing, K.M.; Experimentation, D.G.; Investigation, writing; M.v.D.; Experimentation, investigation: P.H.N. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The CIFAR-10 dataset can be found at: https://www.cs.toronto.edu/~kriz/cifar.html (accessed on 1 May 2020). The Fashion-MNIST dataset can be found at: https://github.com/zalandoresearch/fashion-mnist (accessed 1 May 2020).

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Appendix A.1. Black-Box Settings

We describe the detailed setup of our black box attacks in this paper. We strictly follow the setup of the black box attacks as described in [24]. This setup is carefully chosen by the authors to allow them to properly analyze the security of many defenses under the notion of pure black box attacks and adaptive black box attacks. For the sake of completeness, we re-introduce the setup used in [24].

Algorithm 1 describes the oracle based black-box attack from [4]. The oracle $\mathcal{O}$ represents black-box query access to the target model f and only returns the final class label $F(f(x))$ for a query x (and not the score vector $f(x)$). Initially, the adversary is given (a 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 $\mathcal{X}_0 \subseteq \mathcal{X}$.

Algorithm 1 Construction of synthetic network g in Papernot's oracle based black-box attack [24]

```
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 $S$
5:          training method M;
6:          constant $\lambda$;
7:          number $N$ of synthetic training epochs
8:  Output:
9:          synthetic model $s$ defined by parameters $\theta_s$
10:         ($s$ also has output function $F$ which selects the max confidence score;
11:         $s$ fits architecture $S$)
12:
13: Algorithm:
14: for $N$ iterations do
15:     $\mathcal{D} \leftarrow \{(x, \mathcal{O}(x)) : x \in \mathcal{X}_t\}$
16:     $\theta_s = \mathrm{M}(S, \mathcal{D})$
17:     $\mathcal{X}_{t+1} \leftarrow \{x + \lambda \cdot \mathrm{sgn}(J_{\theta_s}(x)[\mathcal{O}(x)]) : x \in \mathcal{X}_t\} \cup \mathcal{X}_t$
18: end for
19: Output $\theta_s$
```

Let S and θ_s be an a-priori synthetic architecture and the parameter of the synthetic network, respectively. θ_s is trained using Algorithm 1, i.e., the image-label pairs in $\mathcal{D}$ are used to train θ_s using a training method M (e.g., Adam [43]). The data augmentation method (i.e., Jacobian) is used to increase the samples in training dataset $\mathcal{X}_t$ as described in line 17. Algorithm 1 runs N iterations before outputting the final trained parameters θ_s.

Table A1. Training parameters used in the experiments [24].

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

Table A2. Adaptive black-box attack parameters [24].

	$\|\mathcal{X}_0\|$	N	λ
CIFAR-10	50,000	4	0.1
Fashion-MNIST	60,000	4	0.1

Table A3. Architectures of synthetic neural network s from [24,28].

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

Tables A1–A3 from [24] describe the setup of our experiments in this paper. Table A1 presents the setup of the optimization algorithm used for training in Algorithm 1. The architecture of the synthetic model is described in Table A3 and the main parameters for Algorithm 1 for CIFAR-10 and Fashion-MNIST are presented in Table A2.

Appendix A.2. The Adapative Black-Box Attack on Null Class Label Defenses

For the adaptive black-box attack, there is a special case to consider when applying this attack to defenses that have the option of outputting a *null class label*. We study two of these defenses, Odds and BUZz. Here we define the null class label l as a label the defense gives to an input x' when it considers the input to be manipulated by the adversary. This means for a 10 class problem like CIFAR-10, the defense actually has the option of outputting 11 class labels (with class label 11 being the adversarial label). In the context of the adaptive black-box attack, two changes occur. The first change is outside the control of the attacker, and is in regards to the definition of a successful attack. On a defense, that does not output a null class label, the attacker has to satisfy the following output condition: $\mathcal{O}(x') = y'$. We further specify $y' = y_t$ for a targeted attack or $y' \neq y$ for an untargeted attack. Also, we define $\mathcal{O}$ as the oracle in the defense, x' as the adversarial example, y as the original class label and y_t as the target class label. The above formulation only holds when the defense does not employ any detection method (such as adversarial labeling). When adversarial labeling is employed the conditions change slightly. Now a successful attack must be misclassified by the defense and not be the null class label. Formally, we can write this as: $\mathcal{O}(x') = y' \wedge \mathcal{O}(x') \neq l$. While this first change is straightforward, there is another major change in the attack which we describe next.

The second main change in the adaptive black-box attack on null class label defenses comes from training the synthetic model. In the main paper, we mention that training the synthetic model is done with data labeled from the defense $\mathcal{O}(x) = y$. However, we do not use data which has a null class label l, i.e. $\mathcal{O}(x) = l$. We ignore this type of data because this would require modifying the untargeted attack in an unnecessary way. The untargeted attack tries to find the malicious (wrong) label. If the synthetic network is outputting null labels, it is possible for the untargeted attack to produce an adversarial sample that will have a null label. In essence, the attack would fail under those circumstances. To prevent this, the objective function of every untargeted attack would need to be modified, such that the untargeted attack produces the malicious label and it is not the null label. To avoid needlessly complicating the attack, we simply do not use null labeled data. It is an open question of whether using null labeled data to train the synthetic network and the specialized untargeted attack we describe, would actually yield any meaningful performance gains.

Appendix A.3. Vanilla Model Implementation

CIFAR-10: We train a ResNet56 [44] for 200 epochs with ADAM. We accomplish this using Keras :https://github.com/keras-team/keras (accessed on 1 May 2020) and the ResNet56 version 2 implementation: https://keras.io/examples/cifar10_resnet/ (accessed on 1 May 2020). In terms of the dataset, we use 50,000 samples for training and 10,000 samples for testing. All images are normalized in the range [0,1] with a shift of −0.5 so that they are in the range [−0.5, 0.5]. We also use the built in data augmentation technique provided by Keras during training. With this setup our vanilla network achieves a testing accuracy of 92.78%.

Fashion-MNIST: We train a VGG16 network [2] for 100 epochs using ADAM. We use 60,000 samples for training and 10,000 samples for testing. All images are normalized in the range [0,1] with a shift of −0.5 so that they are in the range [−0.5, 0.5]. For this dataset we do not use any augmentation techniques. However, our VGG16 network has a built in resizing layer that transforms the images from 28 × 28 to 32 × 32. We found this process slightly boosts the clean accuracy of the network. On testing data we achieve an accuracy of 93.56%.

Appendix A.4. Barrage of Random Transforms Implementation

The authors of BaRT [14] do not provide source code for their defense. We contacted the authors and followed their recommendations as closely as possible to re-implement their defense. However, some implementation changes had to be made. For the sake of the reproducibility of our results, we enumerate the changes made here.

Image transformations: In the appendix for BaRT, they provide code snippets which are configured to work with scikit image package version 14.0.0. However, due to compatibility issues, the closest version we could implement with our other existing packages was scikit image 14.4.0. Due to the different scikit version, two parts of the defense had to be modified. The original denoising wavelet transformation code in the BaRT appendix had invalid syntax for version 14.4.0, so we had to modify it and run it with different less random parameters.

The second defense change we made was due to error handling. In extremely rarely cases, certain sequences of image transformations return images with NAN values. When contacting the authors they acknowledged that their code failed when using newer versions of sci-kit. As a result, in sci-kit 14.4.0 when we encounter this error, we randomly pick a new sequence of random transformations for the image. We experimentally verified that this has a negligible impact on the entropy of the defense. For example, in CIFAR-10 for the 5 transformation defense, we encounter this error 47 times when running all 50,000 training samples. That means roughly only 0.094% of the possible transformations sequences cannot be used in sci-kit 14.4.0.

It is worth noting one other change we made to the Fashion-MNIST version of this defense. The original BaRT defense was only implemented for ImageNet, a three color channel (RGB) dataset. Fashion-MNIST is a single color channel (grayscale) dataset. As a results two transformation groups are not usable for the Fashion-MNIST BaRT defense (the color space change group and grayscale transformation group).

Training BaRT: In [14] the authors start with a ResNet model pre-trained on ImageNet and further train it on transformed data for 50 epochs using ADAM. The transformed data is created by transforming samples in the training set. Each sample is transformed T times, where T is randomly chosen from distribution $U(0,5)$. Since the authors did not experiment with CIFAR-10 and Fashion-MNIST, we tried two approaches to maximize the accuracy of the BaRT defense. First, we followed the author's approach and started with a ResNet56 pre-trained for 200 epochs on CIFAR-10 with data-augmentation. We then further trained this model on transformed data for 50 epochs using ADAM. For CIFAR-10, we were able to achieve an accuracy of 98.87% on the training dataset and a testing accuracy of 62.65%. Likewise, we tried the same approach for training the defense on the Fashion-MNIST dataset. We started with a VGG16 model that had already been trained with the standard Fashion-MNIST dataset for 100 epochs using ADAM. We then generated the transformed data and trained it for an additional 50 epochs using ADAM. We were able to achieve a 98.84% training accuracy and a 77.80% testing accuracy. Due to the relatively low testing accuracy on the two datasets, we tried a second way to train the defense.

In our second approach we tried training the defense on the randomized data using untrained models. For CIFAR-10 we trained ResNet56 from scratch with the transformed data and data augmentation provided by Keras for 200 epochs. We found the second approach yielded a higher testing accuracy of 70.53%. Likewise for Fashion-MNIST, we trained a VGG16 network from scratch on the transformed data and obtained a testing accuracy of 80.41%. Due to the better performance on both datasets, we built the defense using models trained using the second approach.

Appendix A.5. Improving Adversarial Robustness via Promoting Ensemble Diversity Implementation

The original source code for the ADP defense [11] on MNIST and CIFAR-10 datasets was provided on the author's Github page: https://github.com/P2333/Adaptive-Diversity-Promoting (accessed on 1 May 2020). We used the same ADP training code the authors provided, but trained on our own architecture. For CIFAR-10, we used the ResNet56 model mentioned in subsection Appendix A.3 and for Fashion-MNIST, we used the VGG16 model mentioned in Appendix A.3. We used K = 3 networks for ensemble model. We followed the original paper for the selection of the hyperparameters, which are $\alpha = 2$ and $\beta = 0.5$ for the adaptive diversity promoting (ADP) regularizer. In order to train the model for CIFAR-10, we trained using the 50,000 training images for 200 epochs with a batch size of 64. We trained the network using ADAM optimizer with Keras data augmentation. For Fashion-MNIST, we trained the model for 100 epochs with a batch size of 64 on the 60,000 training images. For this dataset, we again used ADAM as the optimizer but did not use any data augmentation.

We constructed a wrapper for the ADP defense where the inputs are predicted by the ensemble model and the accuracy is evaluated. For CIFAR-10, we used 10,000 clean test images and obtained an accuracy of 94.3%. We observed no drop in clean accuracy with the ensemble model, but rather observed a slight increase from 92.78% which is the original accuracy of the vanilla model. For Fashion-MNIST, we tested the model with 10,000 clean test images and obtained an accuracy of 94.86%. Again for this dataset we observed no drop in accuracy after training with the ADP method.

Appendix A.6. Error Correcting Output Codes Implementation

The training and testing code for ECOC defense [12] on CIFAR-10 and MNIST datasets was provided on the Github page of the authors: https://github.com/Gunjan108/robust-ecoc/ (accessed on 1 May 2020). We employed their "TanhEns32" method which uses 32 output codes and the hyperbolic tangent function as sigmoid function with an ensemble model. We choose this model because it yields better accuracy with clean and adversarial images for both CIFAR-10 and MNIST than the other ECOC models they tested, as reported in the original paper.

For CIFAR-10, we used the original training code provided by the authors. Unlike the other defenses, we did not use a ResNet network for this defense because the models used in their ensemble predict individual bits of the error code. As a result these models are much less complex than ResNet56 (fewer trainable parameters). Due to the lower model complexity of each individual model in the ensemble, we used the default CNN structure the authors provided instead of our own. We did this to avoid over parameterization of the ensemble. We used 4 individual networks for the ensemble model and trained the network with 50,000 clean images for 400 epochs with a batch size of 200. We used data augmentation (with Keras) and batch normalization during training.

We used the original MNIST training code to train Fashion-MNIST by simply changing the dataset. Similarly, to avoid over parameterization, we again used the CNNs the authors used with lower complexity instead of using our VGG16 architecture. We trained the ensemble model with 4 networks for 150 epochs and with a batch size of 200. We did not use data augmentation for this dataset.

For our implementation, we constructed our own wrapper class where the input images are predicted and evaluated using the TanhEns32 model. We tested the defense with 10,000 clean testing images for both CIFAR-10 and Fashion-MNIST, and obtained 89.08% and 92.13% accuracy, respectively.

Appendix A.7. Distribution Classifier Implementation

For the distribution classifier defense [16], we used random resize and pad (RRP) [38] and a DRN [45] as distribution classifier. The authors did not provide a public code for their complete working defense. However, the DRN implementation by the same author was previously released on Github: https://github.com/koukl/drn (accessed on 1 May 2020). We also contacted the authors, followed their recommendations for the training parameters and used the DRN implementation they sent to us as a blueprint.

In order to implement RRP, we followed the resize ranges the paper suggested, specifically for IFGSM attack. Therefore, we chose the resize range as 19 pixels to 25 pixels for CIFAR-10 and 22 pixels to 28 pixels for Fashion-MNIST and used these parameters for all of our experiments.

As for the distribution classifier, the DRN consists of fully connected layers and each node encodes a distribution. We use one hidden layer of 10 nodes. For the final layer, there are 10 nodes (representing each class) and there are two bins representing the logit output for each class. In this type of network the output from the layers are 2D. For the final classification, we convert from 2D to 1D by taking the output from the hidden layer and simply discarding the second bin each time. The distribution classifier then performs the final classification and outputs the class label.

Training: We followed the parameters the paper suggested to prepare training data. First, we collected 1000 correctly classified training clean images for Fashion-MNIST and 10,000 correctly classified clean images for CIFAR-10. Therefore, with no transformation, the accuracy of the networks is 100%. For Fashion-MNIST, we used N = 100 transformation samples and for CIFAR-10, we used N = 50 samples, as suggested in the original paper. After collecting N samples from the RRP, we fed them into our main classifier network and collected the softmax probabilities for each class. Finally, for each class, we made an approximation by computing the marginal distributions using kernel density estimation with a Gaussian kernel (kernel width = 0.05). We used 100 discretization bins to discretize the distribution. For each image, we obtain 100 distribution samples per class. For further details of this distribution, we refer the readers to [16].

We trained the model with the previously collected distribution of 1000 correctly classified Fashion-MNIST images for 10 epochs as the authors suggested. For CIFAR-10, we trained the model with the distributions collected from 10,000 correctly classified images for 50 epochs. For both of the datasets, we used a learning rate of 0.1 and a batch size of 16. The cost function is the cross entropy loss on the logits and the distribution classifier is optimized using backpropagation with ADAM.

Testing: We first tested the RRP defense alone with 10,000 clean test images for both CIFAR-10 and Fashion-MNIST to see the drop in clean accuracy. We observed that this defense resulted in approximately 71% for CIFAR-10 and 82% for Fashion-MNIST. Compared to the clean accuracies we obtain without the defense (93.56% for Fashion-MNIST and 92.78% for CIFAR-10), we observe drops in accuracy after random resizing and padding.

We tested the full implementation with RRP and DRN. In order to compare our results with the paper, we collected 5000 correctly classified clean images for both datasets and collected distributions after transforming images using RRP (N = 50 for Fashion-MNIST and N = 100 for CIFAR-10) like we did for training. We observed a clean test accuracy of 87.48% for CIFAR-10 and 97.76% Fashion-MNIST, which is consistent with the results reported by the original paper. Clearly, if we test all of the clean testing data (10,000 images), we obtain lower accuracy (approximately 83% for CIFAR-10 and 92% for Fashion-MNIST) since there is also some drop in accuracy caused by the CNN. On the other hand, it can be seen that there is a smaller drop in clean accuracy as compared to the basic RRP implementation.

Appendix A.8. Feature Distillation Implementation

Background: The human visual system (HVS) is more sensitive to high frequency parts of the image and less sensitive to the low frequency parts. The standard JPEG compression is based on this understanding, so the standard JPEG quantization table compresses less sensitive frequency parts of the image (i.e. low frequency components) more than other parts. In order to defend against images, a higher compression rate is needed. However, since the CNNs work differently than the HVS, the testing accuracy and defense accuracy both suffer if a higher compression rate is used across all frequencies. In the Feature Distillation defense, as mentioned in Section 3, a crafted quantization technique is used as a solution to this problem. A large quantization step (QS) can reduce adversarial perturbations but also cause more classification errors. Therefore, the proper selection of QS is needed. In the crafted quantization technique, the frequency components are separated as Accuracy Sensitive (AS) band and Malicious Defense (MD) band. A higher quantization step (QS_1) is applied to the MD band to mitigate adversarial perturbations while a lower quantization step (QS_2) is used for AS band to enhance clean accuracy. For more details of this technique, we refer the readers to [18].

Implementation: The implementation of the defense can be found on the author's Github page: https://github.com/zihaoliu123 (accessed on 1 May 2020). However, this defense has only been implemented and tested for the ImageNet dataset by the authors. In order to fairly compare our results with the other defenses, we implemented and tested this defense for CIFAR-10 and Fashion-MNIST datasets.

This defenses uses two different methods: A one-pass process and a two-pass process. The one-pass process uses the proposed quantization/dequantization only in the decompression of the image. The two-pass process, on the other hand, uses the proposed quantization/dequantization in compression followed by one-pass process. In our experiments, we use the two-pass method as it has better defense accuracy than the one-pass process [18].

In the original paper, experiments were performed in order to find a proper selection of (QS_1) and (QS_2) for the AS and MD bands. At the end of these experiments, they set ($QS_1 = 30$) and ($QS_2 = 20$). However, these experiments were performed on ImageNet images where the images are much larger than CIFAR-10 and Fashion-MNIST images. Therefore, we performed experiments in order to properly select QS_1 and QS_2 for the Fashion-MNIST and CIFAR-10 datasets. For each dataset we start with the vanilla classifier (see Appendix A.3). For each vanilla CNN we first do a one-pass and then generate 500 adversarial samples using untargeted FGSM. For CIFAR-10 we use $\epsilon = 0.05$ and for Fashion-MNIST we use $\epsilon = 0.15$. Here we use FGSM to do the hyperparameter selection for the defense because this is how the authors designed the original defense for ImageNet.

After generating the adversarial examples for each QS combination, we do a grid search over the possible hyperparameters QS_1 and QS_2. Specifically, we test 100 defense combinations by varying QS_1 from 10 to 100 and varying QS_2 from 10 to 100. For every possible combination of QS_1 and QS_2 we measure the accuracy on the clean test set and on the adversarial examples. The results of these experiments are shown in Figure A1.

In Figure A1 for the CIFAR-10 dataset, there is an intersection where both the green dots and red dots overlap. This region represents a defense with both higher clean accuracy and higher defense accuracy (the idealized case). There are multiple different combinations of QS_1 and QS_2 that we could choose that give a decent trade-off. Here we arbitrarily select from among these better combinations $QS_1 = 70$ and $QS_2 = 40$ which gives a clean score of 71.4% and a defense accuracy of 21.2%.

In Figure A1 for the Fashion-MNIST dataset, there is no region in which both the clean accuracy and defense accuracy are high. This may show a limitation in the use of feature distillation as a defense for some datasets, as here no ideal trade-off exists. We pick $QS_1 = 70$ and $QS_2 = 40$ which gives a clean score of 89.34% and a defense accuracy of 9%. We picked these values because this combination gave the highest defense accuracy out of all possible hyperparameter choices.

Appendix A.9. End-to-End Image Compression Models Implementation

The original source code for defenses on Fashion-MNIST and ImageNet were provided by the authors of ComDefend [13] on their Github page: https://github.com/jiaxiaojunQAQ/Comdefend (accessed on 1 May 2020). In addition, they included their trained compression and reconstruction models for Fashion-MNIST and CIFAR-10 separately.

Since this defense is a pre-processing module, it does not require modifications to the classifier network [13]. Therefore, in order to perform the classification, we used our own models as described in Section A.3 and we combined them with this pre-processing module.

According to the authors of ComDefend, ComCNN and RecCNN were trained on 50,000 clean (not perturbed) images from the CIFAR-10 dataset for 30 epochs using a batch size of 50. In order to use their pre-trained models, we had to install the canton package v0.1.22 for Python. However, we had incompatibility issues with canton and the other Python packages installed in our system. Therefore, instead of installing this package directly, we downloaded the source code of the canton library from its Github page and added it to our defense code separately. We constructed a wrapper for ComDefend, where the type of dataset (Fashion-MNIST or CIFAR-10) is indicated as input so that the corresponding classifier can be used (either ResNet56 or VGG16). We tested the defense with the testin data of CIFAR-10 and Fashion-MNIST and we were able to achieve an accuracy of 88% and 93% respectively.

Appendix A.10. The Odds Are Odd Implementation

Mathematical background: Here we give a detailed description of the defense based on the statistical test derived from the logits layer. For given image x, we denote $\phi(x)$ as the logits layer (i.e., the input to the softmax layer) of a classifier, $f_y = \langle w_y, \phi(x) \rangle$ where w_y is the weight vector for the class $y, y \in \{1, \cdots, K\}$. The class label is determined by $F(x) = \text{argmax}_y f_y(x)$. We define pair-wise log-odds between class y and z as

$$f_{y,z}(x) = f_z(x) - f_y(x) = \langle w_z - w_y, \phi(x) \rangle. \tag{A1}$$

We denote $f_{y,z}(x+\eta)$ the noise-perturbed log-odds where the noise η is sampled from a distribution $\mathcal{D}$. Moreover, we define the following formulas for a pair (y,z):

$$\begin{aligned} g_{y,z} &:= f_{y,z}(x+\eta) - f_{y,z}(x) \\ \mu_{y,z} &:= \text{E}_{x|y}\text{E}_{\eta}[g_{y,z}(x,\eta)] \\ \sigma_{y,z} &:= \text{E}_{x|y}\text{E}_{\eta}[(g_{y,z}(x,\eta) - \mu_{y,z})^2] \\ \bar{g}_{y,z}(x,\eta) &:= [g_{y,z}(x,\eta) - \mu_{y,z}]/\sigma_{y,z} \end{aligned} \tag{A2}$$

For the original training data set, we compute $\mu_{y,z}$ and $\sigma_{y,z}$ for all (y,z). We apply the untargeted white-box attack (PGD [27]) to generate the adversarial dataset. After that, we compute $\mu^{adv}_{y,z}$ and $\sigma^{adv}_{y,z}$ using the adversarial dataset. We denote $\tau_{y,z}$ as the threshold to control the false positive rate (FPR) and it is computed based on $\mu^{adv}_{y,z}$ and $\sigma^{adv}_{y,z}$. The distribution of clean data and the distribution of adversarial data are represented by (μ, σ) and $(\mu^{adv}, \sigma^{adv})$, respectively. These distributions are supposed to be separated and τ is used to control the FPR.

For a given image x, the statistical test is done as follows. First, we calculate the expected perturbed log-odds $\bar{g}_{y,z}(x) = \mathrm{E}_{\eta}[\bar{g}_{y,z}(x,\eta)]$ where y is the predicted class label of image x given by the vanilla classifier. The test will determine the image x with the label $y = F(x)$ as adversarial (malicious) if

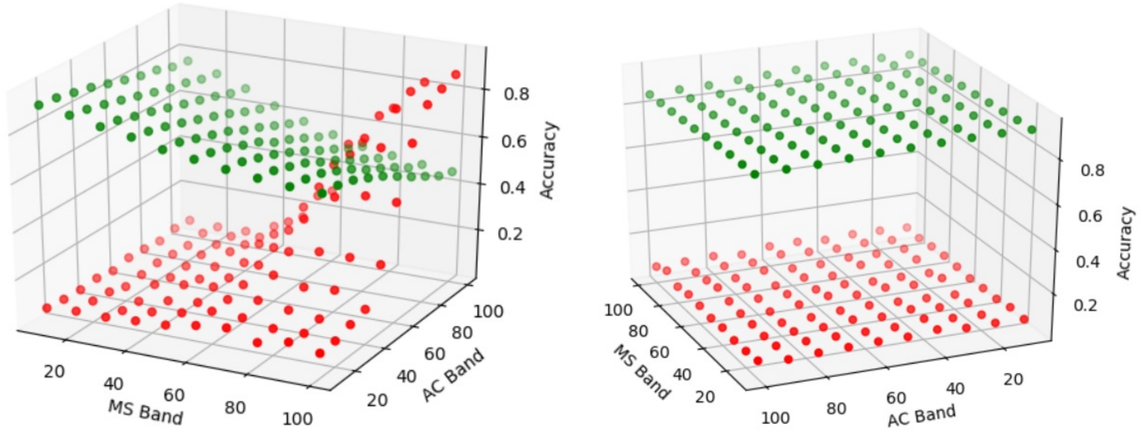
$$\max_{z \neq y}\{\bar{g}_{y,z}(x) - \tau_{y,z}\} \geq 0.$$

Otherwise, the input will be considered benign. In case the test recognizes the image as malicious one, the "corrected" class label z is defined as

$$\max_{z}\{\bar{g}_{y,z}(x) - \tau_{y,z}\}.$$

Implementation details: The original source code for the Odds defense [17] on CIFAR-10 and ImageNet was provided by the authors: https://github.com/yk/icml19_public (accessed on 1 May 2020). We use their code as a guideline for our own defense implementation. We develop the defense for the CIFAR-10 and Fashion-MNIST and datasets. For each dataset, we apply the untargeted 10-iteration PGD attack on the vanilla classifier that will be used in the defense. Note this is a white-box attack. The parameters for the PGD attack are $\epsilon = 0.005$ for CIFAR-10 and $\epsilon = 0.015$ for Fashion-MNIST respectively. By applying the white-box PGD attack we can create the adversarial datasets for the defense. We choose these attack parameters because they yield adversarial examples with small noise. In [17], the authors assume that the adversarial examples are created by adding small noise. Hence, they are not robust against adding the white noises. For a given image, it is normalized first to be in the range $[-0.5, 0.5]$. For each pixel, we generate a noise from $\mathcal{N}(0, 0.05)$ and add it to the pixel.

For CIFAR-10, we create 50,000 adversarial examples. For Fashion-MNIST, we create 60,000 adversarial examples. We calculate μ, σ and τ for each data set for FPR = 1%, 10%, 20%, 30%, 40%, 50% and 80% as described in the mathematical background. For each image, we evaluate it 256 times to compute $\bar{g}_{y,z}(x)$. Table A16 shows the prediction accuracy of the defense for the clean (non-adversarial) dataset for CIFAR-10 and Fashion-MNIST. To compute the clean prediction accuracy, we use 1000 samples from the test dataset of CIFAR-10 and Fashion-MNIST.



(**a**) CIFAR-10

(**b**) Fashion-MNIST

Figure A1. Feature distillation experiments to determine the hyperparameters for the defense. The x and y axis of the grid correspond to the specific hyperparameters for the defense. The Accuracy Sensitive band (denoted as AC in the figure) is the same as QS_1. The Malicious Defense band (denoted as MS in the figure) is the same as QS_2. On the z-axis the accuracy is measured. For every point in this grid two accuracy measurements are taken. The green dot corresponds to the clean accuracy using the QS values specified by the x-y coordinates. The red dot corresponds to the defense accuracy using the QS values specified by the x-y coordinates.

Table A4. CIFAR-10 pure black-box attack. Note the defense numbers in the table are the defense accuracy minus the vanilla defense accuracy. This means they are relative accuracies. The very last row is the actual defense accuracy of the vanilla network.

	FGSM-T	**IFGSM-T**	**MIM-T**	**PGD-T**	**CW-T**	**EAD-T**	**FGSM-U**	**IFGSM-U**	**MIM-U**	**PGD-U**	**CW-U**	**EAD-U**	**Acc**
ADP	0.003	0.016	0.004	0.011	0.003	0.003	0.044	0.022	0.001	0.03	0.009	0.013	0.0152
BaRT-1	0.007	0.026	0.027	0.032	−0.005	−0.005	0.151	0.135	0.089	0.153	−0.07	−0.066	−0.0707
BaRT-10	0.001	−0.001	0.045	−0.012	−0.052	−0.053	−0.039	−0.086	−0.019	−0.041	−0.457	−0.456	−0.4409
BaRT-4	0.006	0.035	0.024	0.009	−0.005	−0.021	0.098	0.099	0.061	0.101	−0.186	−0.175	−0.1765
BaRT-7	0.009	0.014	0.037	−0.001	−0.032	−0.036	0.036	0.025	−0.009	0.021	−0.337	−0.353	−0.3164
BUZz-2	0.053	0.057	0.099	0.05	0.011	0.011	0.352	0.306	0.315	0.338	0.047	0.049	−0.0771
BUZz-8	0.083	0.076	0.131	0.07	0.014	0.014	0.504	0.479	0.508	0.473	0.075	0.078	−0.1713
ComDef	−0.005	0.022	0.004	0.013	−0.014	−0.015	0.078	0.033	−0.02	0.043	−0.054	−0.059	−0.043
DistC	0.009	0.027	0.014	0.024	−0.008	−0.014	0.022	0.041	0.016	0.051	−0.104	−0.11	−0.0955
ECOC	−0.006	0.047	−0.007	0.042	0.004	0.001	0.131	0.103	−0.003	0.099	−0.029	−0.033	−0.0369
FD	−0.018	0.001	0.018	0	−0.025	−0.035	−0.017	0.032	0.01	0.014	−0.248	−0.252	−0.2147
k-WTA	0.003	0.02	0.001	0.021	−0.002	−0.003	0.07	0.019	0.001	0.028	−0.07	−0.08	−0.0529
Odds	0.054	0.041	0.071	0.041	0.003	0.002	0.314	0.207	0.233	0.22	0.011	0.011	−0.2137
Vanilla	0.902	0.917	0.853	0.924	0.984	0.984	0.443	0.453	0.384	0.455	0.923	0.919	0.9278

Table A5. CIFAR-10 adaptive black-box attack 1%. Note the defense numbers in the table are the defense accuracy minus the vanilla defense accuracy. This means they are relative accuracies. The very last row is the actual defense accuracy of the vanilla network.

	FGSM-T	**IFGSM-T**	**MIM-T**	**PGD-T**	**FGSM-U**	**IFGSM-U**	**MIM-U**	**PGD-U**	**CW-T**	**CW-U**	**EAD-T**	**EAD-U**	**Acc**
ADP	−0.015	−0.001	−0.017	−0.013	−0.055	0.002	−0.062	−0.007	0.008	−0.001	0.002	−0.003	0.0152
BaRT-1	0.011	0.017	0.013	0.013	0.125	0.128	0.103	0.121	0.009	−0.061	0.009	−0.061	−0.0695
BaRT-10	−0.05	−0.044	−0.071	−0.042	−0.287	−0.277	−0.325	−0.28	−0.058	−0.439	−0.053	−0.394	−0.4408
BaRT-4	−0.003	0.001	−0.014	−0.011	−0.019	0.002	−0.06	−0.016	−0.008	−0.213	−0.01	−0.185	−0.1834
BaRT-7	−0.035	−0.023	−0.016	−0.017	−0.151	−0.125	−0.208	−0.149	−0.026	−0.307	−0.024	−0.284	−0.319
BUZz-2	0.002	0.017	0.012	0.015	0.148	0.149	0.103	0.148	0.015	0.005	0.016	0.004	−0.0771
BUZz-8	0.026	0.027	0.024	0.024	0.234	0.228	0.2	0.227	0.017	0.005	0.017	0.006	−0.1713
ComDef	0.014	0.016	0.012	0.016	0.13	0.137	0.109	0.131	0.01	−0.007	0.004	−0.004	−0.0424
DistC	−0.003	0.003	0.001	0.01	0.043	0.067	0.007	0.076	0.004	−0.033	0.004	−0.029	−0.0933
ECOC	0.017	0.022	0.014	0.022	0.186	0.192	0.14	0.194	0.008	0.002	0.005	0.001	−0.0369
FD	−0.014	0.006	−0.009	0.007	−0.026	0.012	−0.036	0.001	0.003	−0.012	−0.006	−0.01	−0.2147
k-WTA	−0.022	−0.004	−0.019	−0.006	−0.023	0.04	−0.018	0.042	0.003	−0.008	−0.01	−0.011	−0.0529
Odds	0.009	0.014	0.005	0.004	0.135	0.124	0.104	0.125	0.014	−0.002	0.013	0.001	−0.214
Vanilla	0.973	0.973	0.976	0.974	0.751	0.766	0.793	0.764	0.983	0.995	0.982	0.994	0.9278

Table A6. CIFAR-10 adaptive black-box attack 25%. Note the defense numbers in the table are the defense accuracy minus the vanilla defense accuracy. This means they are relative accuracies. The very last row is the actual defense accuracy of the vanilla network.

	FGSM-T	IFGSM-T	MIM-T	PGD-T	FGSM-U	IFGSM-U	MIM-U	PGD-U	CW-T	CW-U	EAD-T	EAD-U	Acc
ADP	−0.024	−0.049	−0.087	−0.047	−0.074	−0.12	−0.185	−0.131	−0.012	−0.007	−0.008	−0.006	0.0152
BaRT-1	0.015	0.032	0.018	0.029	0.184	0.099	0.089	0.11	−0.012	−0.057	0.001	−0.04	−0.0724
BaRT-10	−0.025	0.006	−0.012	0.015	−0.142	−0.179	−0.208	−0.182	−0.046	−0.398	−0.053	−0.425	−0.4384
BaRT-4	0.003	0.003	0	0.019	0.004	−0.015	−0.072	−0.047	−0.036	−0.196	−0.026	−0.187	−0.1764
BaRT-7	−0.014	−0.013	−0.022	−0.007	−0.106	−0.125	−0.201	−0.15	−0.055	−0.302	−0.04	−0.316	−0.3089
BUZz-2	0.032	0.053	0.051	0.05	0.274	0.228	0.231	0.232	0.003	0.007	0.003	0.011	−0.0771
BUZz-8	0.069	0.07	0.084	0.078	0.419	0.336	0.374	0.335	0.003	0.009	0.005	0.015	−0.1713
ComDef	0.031	0.041	0.029	0.039	0.137	0.126	0.078	0.111	−0.004	−0.009	−0.005	−0.004	−0.0421
DistC	−0.044	−0.007	−0.049	−0.011	−0.022	−0.019	−0.112	−0.02	−0.01	−0.036	−0.011	−0.032	−0.0944
ECOC	−0.044	−0.056	−0.119	−0.041	0.004	−0.073	−0.183	−0.091	−0.004	−0.006	−0.011	−0.009	−0.0369
FD	−0.045	−0.023	−0.045	−0.014	−0.062	−0.05	−0.146	−0.055	−0.011	−0.035	−0.014	−0.031	−0.2147
k-WTA	−0.052	−0.068	−0.112	−0.066	−0.074	−0.174	−0.258	−0.2	−0.008	−0.021	−0.009	−0.019	−0.0529
Odds	0.045	0.047	0.048	0.051	0.237	0.182	0.161	0.16	−0.003	−0.001	−0.003	0	−0.2132
Vanilla	0.924	0.924	0.91	0.921	0.551	0.638	0.597	0.644	0.997	0.991	0.994	0.985	0.9278

Table A7. CIFAR-10 adaptive black-box attack 50%. Note the defense numbers in the table are the defense accuracy minus the vanilla defense accuracy. This means they are relative accuracies. The very last row is the actual defense accuracy of the vanilla network.

	FGSM-T	IFGSM-T	MIM-T	PGD-T	FGSM-U	IFGSM-U	MIM-U	PGD-U	CW-T	CW-U	EAD-T	EAD-U	Acc
ADP	−0.036	−0.116	−0.137	−0.107	−0.077	−0.21	−0.199	−0.229	−0.002	0.001	0.006	0.001	0.0152
BaRT-1	0.034	0.011	0.021	0.028	0.148	0.071	0.051	0.071	−0.009	−0.062	−0.013	−0.064	−0.0753
BaRT-10	0.036	0.046	0.086	0.037	−0.044	−0.092	−0.008	−0.104	−0.043	−0.414	−0.034	−0.433	−0.4399
BaRT-4	0.036	0.02	0.055	0.058	0.075	0.043	0.036	0.02	−0.024	−0.173	−0.039	−0.183	−0.1772
BaRT-7	0.03	0.016	0.055	0.048	0.026	−0.034	0	−0.025	−0.045	−0.297	−0.046	−0.306	−0.3181
BUZz-2	0.088	0.08	0.11	0.093	0.367	0.289	0.316	0.293	0.007	0.012	0.01	0.011	−0.0771
BUZz-8	0.124	0.106	0.162	0.12	0.542	0.428	0.521	0.434	0.007	0.013	0.01	0.014	−0.1713
ComDef	0.01	−0.033	−0.039	−0.014	0.03	−0.015	−0.067	−0.036	−0.005	−0.012	−0.002	−0.016	−0.0411
DistC	−0.021	−0.036	−0.059	−0.014	−0.041	−0.065	−0.117	−0.059	−0.014	−0.042	−0.012	−0.045	−0.0922
ECOC	−0.025	−0.045	−0.11	−0.035	0.02	−0.079	−0.145	−0.09	0.001	−0.004	0.001	−0.019	−0.0369
FD	0.013	0.002	0.008	0.029	0.018	0.021	−0.01	0.015	−0.014	−0.035	−0.014	−0.038	−0.2147
k-WTA	−0.002	−0.139	−0.171	−0.131	−0.064	−0.226	−0.241	−0.248	−0.002	−0.022	−0.005	−0.029	−0.0529
Odds	0.073	0.064	0.098	0.074	0.283	0.181	0.185	0.181	−0.002	−0.002	−0.006	−0.005	−0.2133
Vanilla	0.87	0.886	0.826	0.872	0.423	0.529	0.426	0.531	0.993	0.987	0.99	0.986	0.9278

Table A8. CIFAR-10 adaptive black-box attack 75%. Note the defense numbers in the table are the defense accuracy minus the vanilla defense accuracy. This means they are relative accuracies. The very last row is the actual defense accuracy of the vanilla network.

	FGSM-T	IFGSM-T	MIM-T	PGD-T	FGSM-U	IFGSM-U	MIM-U	PGD-U	CW-T	CW-U	EAD-T	EAD-U	Acc
ADP	−0.038	−0.167	−0.201	−0.167	−0.092	−0.231	−0.216	−0.234	−0.009	−0.006	−0.001	0	0.0152
BaRT-1	0.002	0.027	0.007	0.015	0.117	0.072	0.029	0.069	−0.013	−0.067	−0.006	−0.069	−0.0706
BaRT-10	0.018	0.052	0.061	0.03	−0.055	−0.036	−0.001	−0.03	−0.065	−0.428	−0.058	−0.417	−0.4349
BaRT-4	0.014	0.034	0.045	0.038	0.083	0.088	0.065	0.066	−0.035	−0.2	−0.031	−0.197	−0.1829
BaRT-7	0.016	0.057	0.072	0.05	0.048	0.03	0.001	0.014	−0.035	−0.3	−0.036	−0.334	−0.308
BUZz-2	0.074	0.094	0.104	0.086	0.332	0.328	0.344	0.324	0.007	0.011	0.007	0.014	−0.0771
BUZz-8	0.105	0.126	0.159	0.114	0.541	0.484	0.55	0.464	0.007	0.011	0.008	0.015	−0.1713
ComDef	−0.013	0.003	−0.034	−0.008	0.014	−0.013	−0.063	−0.007	−0.001	−0.019	−0.001	−0.017	−0.0434
DistC	−0.051	−0.042	−0.083	−0.042	−0.078	−0.073	−0.122	−0.077	−0.012	−0.055	−0.016	−0.059	−0.0913
ECOC	−0.06	−0.049	−0.143	−0.054	−0.008	−0.086	−0.163	−0.099	0.004	−0.009	0.002	−0.008	−0.0369
FD	−0.013	0.055	0.004	0.024	0.006	0.097	0.007	0.048	−0.01	−0.032	−0.007	−0.02	−0.2147
k-WTA	−0.036	−0.157	−0.254	−0.162	−0.094	−0.252	−0.243	−0.283	−0.007	−0.031	−0.014	−0.044	−0.0529
Odds	0.05	0.07	0.088	0.05	0.246	0.19	0.196	0.179	−0.002	−0.009	−0.003	−0.014	−0.2133
Vanilla	0.887	0.864	0.822	0.875	0.425	0.478	0.392	0.496	0.993	0.989	0.992	0.984	0.9278

Table A9. CIFAR-10 adaptive black-box attack 100%. Note the defense numbers in the table are the defense accuracy minus the vanilla defense accuracy. This means they are relative accuracies. The very last row is the actual defense accuracy of the vanilla network.

	FGSM-T	IFGSM-T	MIM-T	PGD-T	FGSM-U	IFGSM-U	MIM-U	PGD-U	CW-T	CW-U	EAD-T	EAD-U	Acc
ADP	−0.023	−0.163	−0.172	−0.136	−0.002	−0.163	−0.112	−0.148	0.004	0.006	−0.004	−0.002	0.0152
BaRT-1	0.02	0.011	0.022	0.027	0.173	0.166	0.139	0.169	−0.016	−0.069	−0.018	−0.054	−0.0707
BaRT-10	0.044	0.05	0.126	0.073	0.057	0.078	0.123	0.118	−0.063	−0.405	−0.047	−0.404	−0.4409
BaRT-4	0.044	0.053	0.089	0.053	0.148	0.167	0.184	0.203	−0.02	−0.183	−0.017	−0.199	−0.1765
BaRT-7	0.038	0.06	0.1	0.069	0.113	0.147	0.167	0.161	−0.028	−0.282	−0.045	−0.309	−0.3164
BUZz-2	0.103	0.11	0.168	0.123	0.473	0.426	0.493	0.451	0.009	0.014	0.008	0.01	−0.0771
BUZz-8	0.127	0.13	0.203	0.145	0.628	0.568	0.667	0.576	0.009	0.014	0.009	0.013	−0.1713
ComDef	−0.008	0.005	−0.01	0.007	0.133	0.097	0.084	0.102	−0.003	−0.019	−0.007	−0.022	−0.043
DistC	−0.011	−0.017	−0.041	−0.002	0.005	0.018	−0.01	0.026	0.004	−0.025	0.004	−0.031	−0.0955
ECOC	−0.04	−0.056	−0.105	−0.054	0.091	0.012	−0.03	0.033	0.008	0.002	0.003	−0.008	−0.0369
FD	0.007	0.048	0.062	0.074	0.105	0.181	0.138	0.194	0.002	−0.02	−0.004	−0.029	−0.2147
k-WTA	−0.019	−0.136	−0.174	−0.129	−0.015	−0.157	−0.124	−0.138	−0.003	−0.029	−0.009	−0.034	−0.0529
Odds	0.075	0.077	0.134	0.082	0.312	0.277	0.319	0.299	0.006	−0.012	0	−0.01	−0.2137
Vanilla	0.866	0.861	0.777	0.848	0.334	0.387	0.259	0.374	0.991	0.986	0.991	0.987	0.9278

Table A10. Fashion-MNIST pure black-box attack. Note the defense numbers in the table are the defense accuracy minus the vanilla defense accuracy. This means they are relative accuracies. The very last row is the actual defense accuracy of the vanilla network.

	FGSM-T	**IFGSM-T**	**MIM-T**	**PGD-T**	**CW-T**	**EAD-T**	**FGSM-U**	**IFGSM-U**	**MIM-U**	**PGD-U**	**CW-U**	**EAD-U**	**Acc**
ADP	0.01	−0.05	−0.035	−0.018	−0.003	−0.002	−0.031	−0.006	−0.03	−0.024	0.012	0.018	0.013
BaRT-1	0.041	0.055	0.089	0.062	−0.002	0	0.119	0.173	0.137	0.159	−0.021	−0.017	−0.0317
BaRT-4	0.042	0.05	0.079	0.057	−0.019	−0.015	0.118	0.172	0.149	0.15	−0.132	−0.106	−0.1062
BaRT-6	0.049	0.038	0.084	0.052	−0.029	−0.038	0.091	0.139	0.118	0.133	−0.174	−0.18	−0.1539
BaRT-8	0.055	0.018	0.083	0.059	−0.036	−0.047	0.041	0.09	0.108	0.092	−0.239	−0.222	−0.2212
BUZz-2	0.108	0.1	0.161	0.109	0.003	0.006	0.362	0.482	0.447	0.469	0.04	0.051	−0.0819
BUZz-8	0.128	0.109	0.176	0.12	0.005	0.008	0.47	0.566	0.552	0.563	0.069	0.078	−0.1577
ComDef	0.008	0.053	0.048	0.066	−0.003	0.001	−0.005	0.089	0.06	0.09	0.001	0.011	−0.0053
DistC	0.007	0.018	0.027	0.031	−0.004	−0.003	0.005	0.038	0.026	0.027	−0.008	−0.001	−0.0093
ECOC	0.012	0.056	0.079	0.073	0	0.003	0.043	0.113	0.085	0.108	0.001	0.006	−0.0141
FD	−0.002	0.006	0.022	0.014	−0.017	−0.019	−0.046	0.057	0.011	0.035	−0.094	−0.098	−0.0823
k-WTA	−0.006	0.002	0.013	0.014	−0.001	0	−0.064	0.044	0.012	0.028	0.001	−0.001	−0.0053
Odds	0	0.002	0.004	0.003	0.003	0.001	0.001	0.028	0.015	0.023	0.026	0.021	−0.1809
Vanilla	0.865	0.889	0.817	0.879	0.995	0.992	0.429	0.363	0.351	0.374	0.914	0.905	0.9356

Table A11. Fashion-MNIST adaptive black-box attack 1%. Note the defense numbers in the table are the defense accuracy minus the vanilla defense accuracy. This means they are relative accuracies. The very last row is the actual defense accuracy of the vanilla network.

	FGSM-T	**IFGSM-T**	**MIM-T**	**PGD-T**	**FGSM-U**	**IFGSM-U**	**MIM-U**	**PGD-U**	**CW-T**	**CW-U**	**EAD-T**	**EAD-U**	**Acc**
ADP	0.03	−0.018	−0.009	−0.004	0.051	−0.029	0.008	−0.027	−0.018	0.023	0.005	0.031	0.013
BaRT-1	0.083	0.063	0.05	0.061	0.229	0.137	0.175	0.171	0.041	−0.022	0.009	−0.026	−0.0308
BaRT-4	0.069	0.04	0.034	0.049	0.153	0.056	0.067	0.055	−0.035	−0.182	−0.032	−0.165	−0.0999
BaRT-6	0.046	0.033	−0.006	0.013	0.113	0.008	0.026	0.036	−0.081	−0.274	−0.062	−0.215	−0.1615
BaRT-8	0.046	0.012	−0.028	0.018	0.048	−0.027	−0.03	−0.053	−0.098	−0.326	−0.111	−0.296	−0.2258
BUZz-2	0.155	0.122	0.111	0.117	0.529	0.425	0.436	0.436	0.061	0.079	0.022	0.039	−0.0819
BUZz-8	0.187	0.136	0.123	0.126	0.679	0.488	0.515	0.504	0.064	0.086	0.026	0.05	−0.1577
ComDef	0.032	0.055	0.02	0.025	0.086	0.114	0.123	0.13	0.038	0.042	0.011	0.013	−0.0058
DistC	−0.021	−0.033	−0.029	−0.039	−0.024	−0.057	−0.025	−0.029	0.007	0.029	−0.002	0.008	−0.0093
ECOC	−0.01	0.03	0.008	0.019	0.061	0.038	0.042	0.051	−0.033	−0.1	−0.026	−0.08	−0.0141
FD	−0.073	−0.08	−0.099	−0.06	−0.099	−0.168	−0.15	−0.136	−0.043	−0.1	−0.028	−0.097	−0.0823
k-WTA	0.035	0.036	0.027	0.044	0.072	0.044	0.049	0.068	0.02	0.05	0.016	0.035	−0.0053
Odds	0.031	0.043	0.019	0.038	0.064	0.051	0.065	0.085	0.021	0.033	0.006	0.017	−0.1833
Vanilla	0.807	0.864	0.876	0.873	0.29	0.503	0.475	0.486	0.935	0.91	0.972	0.947	0.9356

Table A12. Fashion-MNIST adaptive black-box attack 25%. Note the defense numbers in the table are the defense accuracy minus the vanilla defense accuracy. This means they are relative accuracies. The very last row is the actual defense accuracy of the vanilla network.

	FGSM-T	IFGSM-T	MIM-T	PGD-T	FGSM-U	IFGSM-U	MIM-U	PGD-U	CW-T	CW-U	EAD-T	EAD-U	Acc
ADP	0.089	0.025	0.036	0.004	0.035	−0.038	−0.008	−0.038	−0.04	−0.092	−0.013	−0.032	0.013
BaRT-1	0.11	0.238	0.195	0.217	0.191	0.165	0.159	0.145	−0.038	−0.134	−0.019	−0.097	−0.0314
BaRT-4	0.141	0.293	0.268	0.256	0.215	0.246	0.224	0.256	−0.067	−0.216	−0.041	−0.218	−0.1018
BaRT-6	0.113	0.285	0.261	0.273	0.209	0.224	0.208	0.206	−0.065	−0.28	−0.056	−0.217	−0.1627
BaRT-8	0.133	0.29	0.294	0.285	0.198	0.195	0.21	0.197	−0.091	−0.341	−0.055	−0.278	−0.221
BUZz-2	0.224	0.42	0.411	0.415	0.542	0.603	0.572	0.601	0.033	0.067	0.034	0.073	−0.0819
BUZz-8	0.288	0.465	0.452	0.454	0.783	0.818	0.808	0.815	0.034	0.073	0.035	0.083	−0.1577
ComDef	0.003	0.17	0.08	0.159	0.043	0.112	0.089	0.105	0.022	0.016	0.015	−0.004	−0.0048
DistC	−0.052	−0.034	−0.062	−0.044	0.013	−0.043	−0.037	−0.054	−0.006	−0.058	−0.001	−0.037	−0.0096
ECOC	0.047	0.188	0.169	0.175	0.014	0.063	0.059	0.06	−0.07	−0.282	−0.067	−0.242	−0.0141
FD	−0.086	−0.012	−0.037	−0.025	−0.048	−0.05	−0.066	−0.072	−0.01	−0.063	−0.036	−0.088	−0.0823
k-WTA	0.012	0.017	−0.001	−0.014	−0.043	−0.029	−0.026	−0.031	−0.279	−0.411	−0.437	−0.402	−0.8516
Odds	−0.064	0.02	0.017	0.024	−0.007	0.042	0.025	0.025	−0.017	−0.037	−0.022	−0.022	−0.1807
Vanilla	0.696	0.53	0.538	0.539	0.108	0.141	0.135	0.149	0.966	0.927	0.965	0.917	0.9356

Table A13. Fashion-MNIST adaptive black-box attack 50%. Note the defense numbers in the table are the defense accuracy minus the vanilla defense accuracy. This means they are relative accuracies. The very last row is the actual defense accuracy of the vanilla network.

	FGSM-T	IFGSM-T	MIM-T	PGD-T	FGSM-U	IFGSM-U	MIM-U	PGD-U	CW-T	CW-U	EAD-T	EAD-U	Acc
ADP	0.115	−0.015	0.021	−0.024	0.01	−0.022	−0.013	−0.027	−0.019	−0.065	−0.011	−0.044	0.013
BaRT-1	0.143	0.227	0.263	0.24	0.183	0.205	0.197	0.195	−0.047	−0.114	−0.02	−0.1	−0.0312
BaRT-4	0.179	0.325	0.327	0.314	0.241	0.246	0.258	0.224	−0.059	−0.216	−0.024	−0.184	−0.1
BaRT-6	0.175	0.331	0.357	0.336	0.251	0.278	0.256	0.268	−0.045	−0.248	−0.03	−0.243	−0.1563
BaRT-8	0.188	0.324	0.342	0.325	0.201	0.24	0.264	0.235	−0.064	−0.296	−0.046	−0.258	−0.2174
BUZz-2	0.264	0.446	0.473	0.444	0.534	0.627	0.611	0.625	0.017	0.057	0.019	0.06	−0.0819
BUZz-8	0.321	0.482	0.514	0.482	0.766	0.835	0.823	0.826	0.018	0.061	0.02	0.067	−0.1577
ComDef	0.044	0.143	0.123	0.158	0.016	0.083	0.08	0.084	0.003	−0.01	−0.004	−0.015	−0.0067
DistC	0.029	−0.024	−0.009	−0.029	0.038	−0.019	−0.007	−0.035	−0.006	−0.038	−0.012	−0.054	−0.0094
ECOC	0.097	0.23	0.238	0.235	0.013	0.075	0.091	0.072	−0.049	−0.133	−0.05	−0.129	−0.0141
FD	−0.019	0	0.009	0	−0.055	−0.019	−0.02	−0.043	−0.02	−0.049	−0.024	−0.069	−0.0823
k-WTA	0.057	−0.006	−0.018	−0.013	−0.037	−0.042	−0.035	−0.058	−0.012	−0.028	−0.032	−0.049	−0.0053
Odds	−0.012	0.027	0	0.016	−0.024	0.013	0.005	0.006	−0.005	0.011	−0.011	0.004	−0.1828
Vanilla	0.666	0.516	0.479	0.516	0.132	0.127	0.107	0.132	0.982	0.939	0.98	0.933	0.9356

Table A14. Fashion-MNIST adaptive black-box attack 75%. Note the defense numbers in the table are the defense accuracy minus the vanilla defense accuracy. This means they are relative accuracies. The very last row is the actual defense accuracy of the vanilla network.

	FGSM-T	IFGSM-T	MIM-T	PGD-T	FGSM-U	IFGSM-U	MIM-U	PGD-U	CW-T	CW-U	EAD-T	EAD-U	Acc
ADP	0.089	0.024	0.103	0.018	−0.033	−0.028	−0.026	−0.016	−0.016	−0.054	−0.006	−0.054	0.013
BaRT-1	0.144	0.299	0.336	0.297	0.151	0.239	0.195	0.254	−0.027	−0.112	−0.002	−0.078	−0.0316
BaRT-4	0.173	0.347	0.41	0.345	0.196	0.304	0.269	0.36	−0.046	−0.17	−0.022	−0.167	−0.107
BaRT-6	0.175	0.372	0.437	0.354	0.202	0.309	0.29	0.327	−0.043	−0.23	−0.027	−0.183	−0.1503
BaRT-8	0.148	0.368	0.422	0.35	0.159	0.303	0.267	0.297	−0.063	−0.309	−0.035	−0.281	−0.2154
BUZz-2	0.232	0.471	0.522	0.478	0.5	0.626	0.594	0.636	0.01	0.05	0.02	0.055	−0.0819
BUZz-8	0.281	0.501	0.563	0.504	0.715	0.838	0.809	0.857	0.01	0.051	0.021	0.061	−0.1577
ComDef	0.029	0.226	0.192	0.221	−0.044	0.127	0.076	0.145	0.002	−0.006	0.009	−0.015	−0.0052
DistC	−0.01	−0.049	−0.007	−0.03	−0.004	−0.025	−0.013	−0.002	−0.016	−0.043	−0.013	−0.056	−0.0096
ECOC	0.04	0.218	0.275	0.232	−0.033	0.075	0.075	0.099	−0.063	−0.156	−0.043	−0.151	−0.0141
FD	−0.087	0.003	0.026	0.004	−0.099	−0.03	−0.039	−0.01	−0.025	−0.039	−0.013	−0.054	−0.0823
k-WTA	0.009	−0.042	−0.007	−0.036	−0.126	−0.035	−0.056	−0.018	−0.002	−0.011	−0.006	−0.024	−0.0053
Odds	−0.043	0.063	0.064	0.049	−0.054	0.049	−0.003	0.068	−0.002	0	0.004	−0.01	−0.1807
Vanilla	0.698	0.494	0.423	0.49	0.195	0.116	0.114	0.096	0.99	0.949	0.979	0.939	0.9356

Table A15. Fashion-MNIST adaptive black-box attack 100%. Note the defense numbers in the table are the defense accuracy minus the vanilla defense accuracy. This means they are relative accuracies. The very last row is the actual defense accuracy of the vanilla network.

	FGSM-T	IFGSM-T	MIM-T	PGD-T	FGSM-U	IFGSM-U	MIM-U	PGD-U	CW-T	CW-U	EAD-T	EAD-U	Acc
ADP	0.086	−0.014	0.039	−0.012	−0.093	−0.038	−0.007	−0.029	−0.006	−0.027	−0.005	−0.03	0.013
BaRT-1	0.129	0.278	0.304	0.274	0.125	0.26	0.228	0.258	−0.015	−0.1	−0.015	−0.062	−0.0317
BaRT-4	0.165	0.319	0.383	0.317	0.176	0.276	0.273	0.288	−0.052	−0.182	−0.032	−0.148	−0.1062
BaRT-6	0.175	0.347	0.397	0.346	0.136	0.314	0.306	0.293	−0.058	−0.237	−0.044	−0.213	−0.1539
BaRT-8	0.159	0.344	0.389	0.327	0.166	0.287	0.254	0.274	−0.051	−0.255	−0.033	−0.243	−0.2212
BUZz-2	0.227	0.432	0.489	0.43	0.462	0.657	0.62	0.653	0.006	0.037	0.007	0.057	−0.0819
BUZz-8	0.279	0.469	0.535	0.466	0.672	0.818	0.809	0.835	0.007	0.039	0.009	0.061	−0.1577
ComDef	0.014	0.131	0.103	0.136	−0.029	0.088	0.074	0.093	−0.003	−0.018	−0.006	−0.012	−0.0053
DistC	0.014	−0.015	−0.001	−0.012	−0.047	−0.035	−0.011	−0.029	−0.011	−0.034	−0.021	−0.026	−0.0093
ECOC	0.057	0.193	0.233	0.176	−0.081	0.068	0.074	0.088	−0.026	−0.083	−0.026	−0.076	−0.0141
FD	−0.094	−0.038	−0.006	−0.041	−0.158	−0.037	−0.043	−0.037	−0.026	−0.071	−0.032	−0.064	−0.0823
k-WTA	0.047	−0.032	−0.024	−0.013	−0.138	−0.045	−0.048	−0.04	−0.008	−0.018	−0.026	−0.041	−0.0053
Odds	−0.036	0.011	0.009	0.017	−0.01	0.03	0.024	0.036	−0.002	−0.017	−0.004	−0.002	−0.1809
Vanilla	0.707	0.529	0.46	0.531	0.234	0.123	0.111	0.118	0.993	0.961	0.991	0.939	0.9356

Table A16. Clean prediction accuracy of the Odds defense on Fashion-MNIST and CIFAR-10 with different FPRs.

FPR	1%	10%	20%	30%	40%	50%	80%
FashionMNIST	78.6%	79.6%	78.5%	79.5%	78.6%	78.8%	79.1%
CIFAR-10	0.3%	27.8%	43.2%	61.1%	75.2%	86.2%	99.3%

References

1. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems 25 (NIPS 2012), Lake Tahoe, NV, USA, 3–8 December 2012; pp. 1097–1105.
2. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition . *arXiv* **2014**, arXiv:1409.1556. Available online: https://arxiv.org/abs/1409.1556 (accessed on 16 September 2021).

3. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. *arXiv* **2014**, arXiv:1412.6572. Available online: https://arxiv.org/abs/1412.6572 (accessed on 16 September 2021).
4. Papernot, N.; McDaniel, P.D.; Goodfellow, I.J.; Jha, S.; Celik, Z.B.; Swami, A. Practical Black-Box Attacks against Machine Learning. *ACM Asia CCS* **2017**, *2017*, 506–519.
5. Chen, P.Y.; Zhang, H.; Sharma, Y.; Yi, J.; Hsieh, C.J. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, Dallas, TX, USA, 3 November 2017; pp. 15–26.
6. Chen, J.; Jordan, M.I. Boundary Attack++: Query-Efficient Decision-Based Adversarial Attack. *arXiv* **2014**, arXiv:1904.02144v1. Available online: https://gaokeji.info/abs/1904.02144v1 (accessed on 16 September 2021).
7. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.J.; Fergus, R. Intriguing properties of neural networks. *arXiv* **2013**, arXiv:1312.6199. Available online: https://arxiv.org/abs/1312.6199 (accessed on 16 September 2021).
8. Papernot, N.; McDaniel, P.; Goodfellow, I. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv* **2016**, arXiv:1605.07277. Available online: https://arxiv.org/abs/1605.07277 (accessed on 16 September 2021).
9. Athalye, A.; Carlini, N.; Wagner, D. Obfuscated Gradients Give a False Sense of Security: Circumventing Defenses to Adversarial Examples. In Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 274–283.
10. Liu, Y.; Chen, X.; Liu, C.; Song, D. Delving into Transferable Adversarial Examples and Black-box Attacks. *arXiv* **2017**, arXiv:1611.02770. Available online: https://arxiv.org/abs/1611.02770 (accessed on 16 September 2021).
11. Pang, T.; Xu, K.; Du, C.; Chen, N.; Zhu, J. Improving Adversarial Robustness via Promoting Ensemble Diversity. *Int. Conf. Mach. Learn.* **2019**, *97*, 4970–4979 .
12. Verma, G.; Swami, A. Error Correcting Output Codes Improve Probability Estimation and Adversarial Robustness of Deep Neural Networks. In Proceedings of the 33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, BC, Canada, 8–14 December 2019.
13. Jia, X.; Wei, X.; Cao, X.; Foroosh, H. Comdefend: An efficient image compression model to defend adversarial examples. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 6084–6092.
14. Raff, E.; Sylvester, J.; Forsyth, S.; McLean, M. Barrage of random transforms for adversarially robust defense. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 6528–6537.
15. Xiao, C.; Zhong, P.; Zheng, C. Enhancing Adversarial Defense by k-Winners-Take-All. In Proceedings of the International Conference on Learning Representations, Addis Ababa, Ethiopia, 26–30 April 2020.
16. Kou, C.; Lee, H.K.; Chang, E.C.; Ng, T.K. Enhancing transformation-based defenses against adversarial attacks with a distribution classifier. In Proceedings of the International Conference on Learning Representations, New Orleans, LA, USA, 6–9 May 2019.
17. Roth, K.; Kilcher, Y.; Hofmann, T. The Odds are Odd: A Statistical Test for Detecting Adversarial Examples. In Proceedings of the International Conference on Machine Learning, Long Beach, CA, USA, 10–15 June 2019; pp. 5498–5507.
18. Liu, Z.; Liu, Q.; Liu, T.; Xu, N.; Lin, X.; Wang, Y.; Wen, W. Feature distillation: Dnn-oriented jpeg compression against adversarial examples. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 10–15 June 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 860–868.
19. Carlini, N.; Athalye, A.; Papernot, N.; Brendel, W.; Rauber, J.; Tsipras, D.; Goodfellow, I.; Madry, A.; Kurakin, A. On evaluating adversarial robustness. *arXiv* **2019**, arXiv:1902.06705. Available online: https://arxiv.org/abs/1902.06705 (accessed on 16 September 2021).
20. Carlini, N.; Wagner, D. Adversarial examples are not easily detected: Bypassing ten detection methods. In Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security, Dallas, TX, USA, 3 November 2017; pp. 3–14.
21. He, W.; Wei, J.; Chen, X.; Carlini, N.; Song, D. Adversarial example defense: Ensembles of weak defenses are not strong. In Proceedings of the 11th {USENIX} Workshop on Offensive Technologies ({WOOT} 17), Vancouver, BC, Canada, 14–15 August 2017.
22. Tramer, F.; Carlini, N.; Brendel, W.; Madry, A. On adaptive attacks to adversarial example defenses. *arXiv* **2020**, arXiv:2002.08347. Available online: https://arxiv.org/abs/2002.08347 (accessed on 16 September 2021).
23. Dong, Y.; Fu, Q.A.; Yang, X.; Pang, T.; Su, H.; Xiao, Z.; Zhu, J. Benchmarking Adversarial Robustness on Image Classification. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Seattle, WA, USA, 13–19 June 2020.
24. Mahmood, K.; Nguyen, P.H.; Nguyen, L.M.; Nguyen, T.; van Dijk, M. BUZz: BUffer Zones for defending adversarial examples in image classification. *arXiv* **2019**, arXiv:1910.02785. Available online: https://arxiv.org/abs/1910.02785 (accessed on 16 September 2021).
25. Yuan, X.; He, P.; Zhu, Q.; Li, X. Adversarial Examples: Attacks and Defenses for Deep Learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *30*, 2805–2824. [CrossRef] [PubMed]
26. Brendel, W.; Rauber, J.; Bethge, M. Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 30 April 30–3 May 2018.

27. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards deep learning models resistant to adversarial attacks. *arXiv* **2021**. arXiv:1706.06083. Available online: https://arxiv.org/abs/1706.06083 (accessed on 16 September 2021).
28. Carlini, N.; Wagner, D. Towards evaluating the robustness of neural networks. In Proceedings of the 2017 IEEE Symposium on Security and Privacy (sp), San Jose, CA, USA, 22–26 May 2017; IEEE: Piscataway, NJ, USA, 2017; pp. 39–57.
29. Guo, C.; Gardner, J.R.; You, Y.; Wilson, A.G.; Weinberger, K.Q. Simple black-box adversarial attacks. In Proceedings of the 36th International Conference on Machine Learning, Long Beach, CA, USA, 9–15 June 2019.
30. Chen, J.; Gu, Q. Rays: A ray searching method for hard-label adversarial attack. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event, 6–10 July 2020; pp. 1739–1747.
31. Kurakin, A.; Goodfellow, I.; Bengio, S. Adversarial examples in the physical world. *arXiv* **2017**. arXiv:1607.02533.
32. Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; Li, J. Boosting adversarial attacks with momentum. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 9185–9193.
33. Chen, P.Y.; Sharma, Y.; Zhang, H.; Yi, J.; Hsieh, C.J. Ead: Elastic-net attacks to deep neural networks via adversarial examples. In Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018.
34. Chen, J.; Jordan, M.I.; Wainwright, M.J. Hopskipjumpattack: A query-efficient decision-based attack. In Proceedings of the 2020 IEEE Symposium on Security and Privacy (sp), San Francisco, CA, USA, 17–21 May 2020; IEEE: Piscataway, NJ, USA, 2020, pp. 1277–1294.
35. Byun, J.; Go, H.; Kim, C. Small Input Noise is Enough to Defend Against Query-based Black-box Attacks. *arXiv* **2021**, arXiv:2101.04829. Available online: https://arxiv.org/abs/2101.04829 (accessed on 16 September 2021)
36. Athalye, A.; Engstrom, L.; Ilyas, A.; Kwok, K. Synthesizing robust adversarial examples. In Proceedings of the International Conference on Machine Learning, Stockholm, Sweden, 10–15 July 2018; pp. 284–293.
37. Carlini, N.; Wagner, D. MagNet and "Efficient Defenses against Adversarial Attacks" Are Not Robust to Adversarial Examples. *arXiv* **2017**, arXiv:cs.LG/1711.08478. Available online: https://arxiv.org/abs/1711.08478 (accessed on 16 September 2021).
38. Xie, C.; Wang, J.; Zhang, Z.; Ren, Z.; Yuille, A. Mitigating adversarial effects through randomization. *arXiv* **2018**, arXiv:1711.01991. Available online: https://arxiv.org/abs/1711.01991 (accessed on 16 September 2021).
39. Krizhevsky, A.; Nair, V.; Hinton, G. Learning Multiple Layers of Features from Tiny Images. Available online: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.222.9220&rep=rep1&type=pdf (accessed on 16 September 2021)
40. Xiao, H.; Rasul, K.; Vollgraf, R. Fashion-MNIST: A Novel Image Dataset for Benchmarking Machine Learning Algorithms. *arXiv* **2017**. arXiv:1708.07747. Available online: https://arxiv.org/abs/1708.07747 (accessed on 16 September 2021).
41. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
42. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Fei-Fei, L. Imagenet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
43. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980. Available online: https://arxiv.org/abs/1412.6980 (accessed on 16 September 2021).
44. He, K.; Zhang, X.; Ren, S.; Sun, J. Identity Mappings in Deep Residual Networks. *Lect. Notes Comput. Sci.* **2016**, 630–645. [CrossRef]
45. Kou, C.K.L.; Lee, H.K.; Ng, T.K. A compact network learning model for distribution regression. *Neural Netw.* **2019**, *110*, 199–212. [CrossRef] [PubMed]