Privacy Preserving Defense For Black Box Classifiers Against On-Line Adversarial Attacks

Rajkumar Theagarajan, Student Member, IEEE, and Bir Bhanu, Life Fellow, IEEE

Abstract—Deep learning models have been shown to be vulnerable to adversarial attacks. Adversarial attacks are imperceptible perturbations added to an image such that the deep learning model misclassifies the image with a high confidence. Existing adversarial defenses validate their performance using only the classification accuracy. However, classification accuracy by itself is not a reliable metric to determine if the resulting image is “adversarial-free”. This is a foundational problem for online image recognition applications where the ground-truth of the incoming image is not known and hence we cannot compute the accuracy of the classifier or validate if the image is “adversarial-free” or not. This paper proposes a novel privacy preserving framework for defending Black box classifiers from adversarial attacks using an ensemble of iterative adversarial image purifiers whose performance is continuously validated in a loop using Bayesian uncertainties. The proposed approach can convert a single-step black box adversarial defense into an iterative defense and proposes three novel privacy preserving Knowledge Distillation (KD) approaches that use prior meta-information from various datasets to mimic the performance of the Black box classifier. Additionally, this paper proves the existence of an optimal distribution for the purified images that can reach a theoretical lower bound, beyond which the image can no longer be purified. Experimental results on six public benchmark datasets namely: 1) Fashion-MNIST, 2) CIFAR-10, 3) GTSRB, 4) MIO-TCD, 5) Tiny-ImageNet, and 6) MS-Celeb show that the proposed approach can consistently detect adversarial examples and purify or reject them against a variety of adversarial attacks.

Index Terms—Adversarial Defense, Bayesian uncertainties, Black box defense, Ensemble of defenses, Image purifiers, Knowledge distillation, Privacy preserving defense.

1 INTRODUCTION

Although deep learning has had astounding success on several image classification tasks, it has been shown to be vulnerable to adversarial attacks [1] - [3]. Adversarial attacks to an image are carefully crafted perturbations that are so subtle that a human observer does not even notice the modification at all, but can cause deep learning models to misclassify the input. Adversarial attacks can broadly be classified into two categories namely: 1) white box, and 2) black box attacks. White box adversarial attacks assume that the adversary has partial/full knowledge about the target model’s architecture, parameters, and training data [4], whereas in the black box attack, the adversary lacks this information [5]. In this paper we focus on defending against black box adversarial attacks.

Current defenses against adversarial attacks can be classified into four approaches: 1) modifying the training data, 2) modifying the model, 3) using auxiliary tools, and 4) detecting and rejecting adversarial examples. Modifying the training data involves augmenting the training dataset with adversarial examples and re-training the classifier [6] - [8] or performing N number of pre-selected image transformations in a random order [9] - [12]. Modifying the model involves pruning the architecture of the classifier [14] - [16] or adding pre/post-processing layers to it [21] - [23]. The category 1) and 2) approaches are not compatible to be used for defending Black box models because they require information about the target model’s architecture/training data which are not known.

Using auxiliary tools involves having an independent module that is able to process the input before it is passed to the classifier [26] - [28]. Detecting and rejecting adversarial examples involve using domain adaptation techniques and carrying out statistical analysis [30] - [33] to detect adversarial examples. The approaches in the latter two categories are the most suitable for defending Black box classifiers as they do not require any information about the target model’s architecture/training data. However, a drawback of these approaches is that they do not quantify how much adversarial component is left in the resulting purified image. Further, these approaches are single-step defenses which means that the defense can purify the image only once and the defense assumes that the purified image is void of all adversarial components. These defenses are trained on annotated datasets and after achieving a reasonable performance, they are deployed to defend real-world applications. Once deployed it is assumed that all of the purified images are no longer adversarial, but in reality this is not the case.

A foundational problem for online and safety-critical applications [35] - [39] is that it is not possible to know the annotations of all of the incoming input images, and these single-step defenses do not have the ability to determine by themselves whether the purified image is adversarial or not which could further cause disastrous results. To address this problem, this paper proposes a general framework for defending black box classifiers from adversarial attacks using an ensemble of iterative
TABLE 1: Summary of related work for white box adversarial defense

<table>
<thead>
<tr>
<th>Approach</th>
<th>Authors</th>
<th>Methods and Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adversarial training</td>
<td>Goodfellow et al. [7], Huang et al. [5] Tramèr et al. [7], Zhang et al. [8]</td>
<td>Adversarial examples are introduced into the dataset to improve the robustness of the training model with the legalized adversarial examples.</td>
</tr>
<tr>
<td>Gradient hiding</td>
<td>Tramèr et al. [7]</td>
<td>Hides the information about the gradient from adversaries. However, learning a Substitute model through knowledge distillation can penetrate this defense.</td>
</tr>
<tr>
<td>Data compression</td>
<td>Das et al. [9], Dziugaite et al. [10] Xu et al. [23]</td>
<td>Used JPEG compression on input images and color scale normalization as a pre-processing step to defend classifiers.</td>
</tr>
<tr>
<td>Data randomization</td>
<td>Xie et al. [24], Wang et al. [25]</td>
<td>Performed random resizing and addition of random noise to reduce the effect of the adversarial perturbations.</td>
</tr>
<tr>
<td>Image transformation</td>
<td>Gao et al. [11], Raff et al. [12]</td>
<td>Used hand-crafted image transformations in a random order to remove the effect of adversarial perturbations.</td>
</tr>
<tr>
<td>Blocking the transferability</td>
<td>Hosseini et al. [13]</td>
<td>Performed three step null labeling method that classifies adversarial examples into a “null” class.</td>
</tr>
<tr>
<td>Defensive distillation</td>
<td>Papernot et al. [15]</td>
<td>Defensive distillation is done by converting the softmax output logit probabilities to create soft labels.</td>
</tr>
<tr>
<td>Thermometer encoding</td>
<td>Buckman et al. [21]</td>
<td>Discretized the input by replacing pixel values with thermometer encoding.</td>
</tr>
<tr>
<td>Feature denoising</td>
<td>Liao et al. [14], Xie et al. [16] Mustafa et al. [17]</td>
<td>Used feature disentanglement to defend against adversarial attacks.</td>
</tr>
<tr>
<td>Architecture pruning</td>
<td>Liu et al. [18], Dhillon et al. [19] Ye et al. [20]</td>
<td>Reduced the effect of the adversarial perturbation by pruning the architecture and weights of the original classifier.</td>
</tr>
</tbody>
</table>

TABLE 2: Summary of related work for black box adversarial defense

<table>
<thead>
<tr>
<th>Approach</th>
<th>Authors</th>
<th>Methods and Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defense-GAN</td>
<td>Samangouei et al. [101]</td>
<td>Used Generative Adversarial Networks (GAN) to minimize the reconstruction error.</td>
</tr>
<tr>
<td>PixelDefend</td>
<td>Song et al. [27]</td>
<td>Used a PixelCNN to purify adversarial examples.</td>
</tr>
<tr>
<td>ShieldNets</td>
<td>Theagarajan et al. [28]</td>
<td>Used Probabilistic Adversarial Robustness (PAR).</td>
</tr>
<tr>
<td>MagNet</td>
<td>Meng et al. [26]</td>
<td>Used the reconstruction loss of an autoencoder to detect and purify adversarial attack.</td>
</tr>
<tr>
<td>SafetyNet</td>
<td>Lu et al. [29]</td>
<td>Detected if an image is adversarial using RGBD images.</td>
</tr>
<tr>
<td>Statistical testing</td>
<td>Grosse et al. [31]</td>
<td>Used kernel based two sample test to distinguish adversarial examples from the dataset.</td>
</tr>
<tr>
<td>Policy based detection</td>
<td>Lin et al. [32]</td>
<td>Compared action distributions from previous replays to detect policy targeting attacks.</td>
</tr>
<tr>
<td>Bayesian uncertainty</td>
<td>Rawat et al. [73]</td>
<td>Distinguished between adversarial and original images using Bayesian uncertainties.</td>
</tr>
<tr>
<td>Ensemble of iterative</td>
<td>Theagarajan and Bhanu [94]</td>
<td>Used an ensemble of adversarial defenses to defend black box facial recognition classifiers against adversarial attacks.</td>
</tr>
<tr>
<td>adversarial defenses</td>
<td>(This paper)</td>
<td>1) Proposed a novel model agnostic defense framework that uses an ensemble of adversarial defenses to iteratively purify adversarial images for various black box applications, 2) showed the relationship between the adversarial image and its corresponding purified image and, 3) proved the existence of a theoretical lower bound in the input space beyond which the image cannot be further purified.</td>
</tr>
</tbody>
</table>

adversarial defenses whose performance is continuously validated in a loop using Bayesian uncertainties. The paper proposes three novel knowledge distillation approaches for transferring the functionality of the black box classifier into our defense. The experimental results on six different datasets shows that our defense can be applied to defend various black box applications ranging from the general Fashion-MNIST [78] and CIFAR-10 [79] datasets to face biometrics and classification of vehicles and traffic signs for autonomous driving. In summary, the contributions of this paper are as follows:

- To the best of our knowledge, this is the first approach that defends against adversarial attacks using an ensemble of iterative adversarial defenses and can convert any single-step defense into an iterative adversarial defense.
- We prove theoretically and demonstrate empirically that there exists a lower bound in the input space on the amount of purification carried out on an image beyond which it can no longer be purified.
- The paper proposes three novel privacy preserving knowledge distillation approaches that exploit prior meta-information of the training dataset in order to transfer the functionality of the black box classifier into our defense and it does not require any information such as the logits probabilities [55], [56] or Teacher model architecture [63], [64].
- To the best of our knowledge there is no other work that defends black box models from adversarial attacks without any information of the model’s trained parameters or the dataset that are used for training the model prior to our work in [94].
- Exhaustive evaluation on six public benchmark datasets shows that our approach is able to consistently purify/reject adversarial examples. Various ablation studies show that it is computationally expensive to break the defense of our framework compared to stand-alone defenses.

2 RELATED WORK

In this section we describe state-of-the-art black box adversarial defenses and knowledge distillation approaches and contrast them with our approach. Tables 1 - 3 show a summary of the related work done for white box, black box adversarial defense, and knowledge distillation, respectively. In contrast to the related work, our work is significantly different in the following aspects:
TABLE 3: Summary of the related work for knowledge distillation

<table>
<thead>
<tr>
<th>Authors</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>Orekondy et al. [56]</td>
<td>Used the softmax logit probability scores and cross dataset images to distill knowledge.</td>
</tr>
<tr>
<td>Furlanello et al. [55]</td>
<td>Decomposed the predictions of the white box model into incorrect prediction and ground-truth information.</td>
</tr>
<tr>
<td>Frost and Hinton [63]</td>
<td>Distilled deep networks to decision trees in order to explain the predictions.</td>
</tr>
<tr>
<td>Ba and Caruana [64]</td>
<td>Distilled knowledge using the $L_2$ norm between the logits of the white box Teacher and Student model.</td>
</tr>
<tr>
<td>Shin et al. [66]</td>
<td>Used knowledge distillation for the purpose of minimizing forgetting in continuous learning</td>
</tr>
<tr>
<td>Tan et al. [66]</td>
<td>Performed distillation using the difference in prediction between a white box Student and Teacher classifier.</td>
</tr>
<tr>
<td>Wang et al. [67]</td>
<td>Explained the behavior of white box deep neural networks by transferring the functionality to a smaller model.</td>
</tr>
<tr>
<td>Song et al. [68]</td>
<td>Used a dual branch student neural network to learn the latent space which is guided by a white box teacher using Attentive Knowledge Distillation (AKD).</td>
</tr>
<tr>
<td>Tessler et al. [69]</td>
<td>Proposed a hierarchical learning framework that consists of multiple Deep Skill Networks (DSN) to perform knowledge transfer and life long learning.</td>
</tr>
<tr>
<td>Crowley et al. [70]</td>
<td>Proposed structural model distillation to reduce the memory cost using block convolutions that produces a student architecture that is a simple transformation of the white box teacher architecture.</td>
</tr>
<tr>
<td>Theagarajan and Bhanu</td>
<td>Used publicly available crowd sourced images to probe and distill the knowledge of a black box classifier.</td>
</tr>
<tr>
<td>Theagarajan and Bhanu</td>
<td>Proposed three knowledge distillation approaches that exploit the prior meta-information of the training datasets and showed that knowledge distillation can still be performed even when we do not have any knowledge such as the logit probabilities [58], [59] or architecture about the black box classifier [63], [64]</td>
</tr>
</tbody>
</table>

Fig. 1: Overall framework of our approach. Faces are masked to hide the identity of individuals.

- **Ensemble of Iterative Adversarial Attacks:** To the best of our knowledge, there is no other work that can convert single-step adversarial defenses into an iterative defense and also quantify the amount of adversarial component in the output of the defense after each iteration.
- **Relationship between Adversarial and Non-adversarial images:** This paper proves the existence of a lower bound in the input space beyond which an image can no longer be purified. We empirically verify this by showing the convergence of all the individual defenses across multiple iterations of purification.
- **Knowledge Distillation using Meta-information:** This is the first approach that exploits only the meta-information of a dataset to perform knowledge distillation on strict black box classifiers. Unlike the related work shown in Table 6 we assume we know neither the architecture of the classifier, the output logit probabilities, nor the data distribution used for training.

3 TECHNICAL APPROACH

Fig. 1 shows the overall framework of our approach. The input image ($X$) first passes through the Bayesian Convolutional Neural Network (CNN) and if the image is classified as an adversarial image, it is purified by the ensemble of independently trained iterative adversarial defenses. The purified image from each defense algorithm is averaged resulting in the average purified image ($X'_i$), where $i$ refers to the current iteration number. Next, $X'_i$ is passed as input back to the Bayesian CNN. If $X'_i$ is not adversarial, it is passed as input to the Black box classifier for final classification, else, it is passed again as input to the ensemble of adversarial defenses and this continues for $M$ iterations. After $M$ iterations, if $X'_M$ is still adversarial then the image is rejected.

In Fig. 1, we chose to take the average of the purified image of each individual defense as this further helps in removing the high frequency adversarial perturbations [40], [41] that may have not been purified by some of the individual defenses. Additionally, the number of iterations for purification, $M$, is chosen empirically depending on the training dataset used (see Section 3.4.1) and we prove the existence of a theoretical lower bound beyond which an image cannot be further purified (see Section 3.4.2).

3.1 Assumptions and Target Applications of our Defense

- **Assumptions of our Defense:** 1) The output of the Black box model is the top predicted class without any probabilities.
2) The architecture, parameters and entire training dataset of the Black box are not known to both the adversary and the defense algorithm. 3) The outputs of the Bayesian framework and ensemble of adversarial image purifiers are not shared with the adversary (i.e., the adversary can only see the final classification of the Black box). Although this is a strong assumption, we perform ablation studies to estimate the computational burden required to break the defense, when the adversary knows partial information about our defense strategy (see Section 4.6).

- **Reasons for our Assumptions:** When black box classifiers are made publicly available, they must neither reveal the sensitive details of their training data nor the features encoded within the classifier, but they must still be accurate and solve the underlying learning task. This kind of protection is also known as Security through Obscurity [42] and it is important that these models preserve the privacy of their sensitive training data. Advances in the field of Differential Privacy Learning have shown that it is possible to train black box classifiers without compromising the privacy and personnel information contained within the training dataset [43] - [46]. On the contrary, recent works have also shown that it is possible for an adversary to create adversarial attacks to determine if a specific data point was used to train the model and thus causing a leak in sensitive information [47] - [50]. There have also been similar news articles such as the ban on facial recognition technology in the state of California, USA [51] due to the misuse of information and privacy concerns. Once personally identifiable information is leaked, it is impossible to reverse it, hence it is imperative that adversarial defenses protect black box classifiers against adversarial attacks while at the same time maintain privacy. For these reasons we assume that our defense has little to no knowledge of the training data as well as the information on the learned features of the black box classifier. Our proposed defense approach is suited for black box applications that require security against adversarial attacks and preserve user privacy without giving away any sensitive information. These include a wide variety of machine intelligence applications such as human biometrics [94], [115], remote surveillance [52], autonomous driving [53], etc.

### 3.2 Functionality Transfer via Knowledge Distillation

According to the assumptions described in Section 3.1, we do not have any information about the target black box model’s architecture, parameters, and training dataset, hence we cannot directly use our defense algorithms. To overcome this problem, we transfer the functionality of the black box model to a substitute model using knowledge distillation [54] - [56]. In this paper we assume that the target black box classifier is deterministic (weights are fixed after training) and ignore approaches that continuously update the weights of the CNN (e.g., by using incremental [57] and reinforcement learning [58]). Based on the above discussion, we propose three novel Knowledge Distillation (KD) approaches namely: KD-1, KD-2, and KD-3 that exploit the meta-information related to the training datasets. Unlike the approaches proposed in Table 3, our knowledge distillation method is different because the approaches in Table 3 assume that they have prior knowledge about the training dataset or the logit probability of outputs and/or architecture of the target black box classifier. In our approach, we assume a strict black box classifier where we have no knowledge about the architecture or logit probabilities of the black box classifier and the only observable output is the single top predicted class. Fig. 2 shows illustrations of our three knowledge distillation approaches.

#### Assumptions of our KD approaches:

- **KD-1:** We know the total number of classes, the names of all the classes in the training dataset and Z% of images from each class.
- **KD-2:** We know only the total number of classes, the names of all the classes in the training dataset and the domain the dataset belongs to.
- **KD-3:** We know only the total number of classes in the training dataset and the domain the dataset belongs to.

In KD-2 and KD-3, domain refers to the application of the black box classifier, e.g., classifiers trained on the MS-
Celeb [59] and MIO-TCD [60] datasets generally belong to the domain of face recognition and vehicle classification, respectively.

3.2.1 Knowledge Distillation-1 (KD-1)

For this case we assume that we have access to Z% of images of each class from the training dataset used to train the black box classifier as shown in Fig. 2(a). For simplicity we refer to the black box classifier as the Teacher model and our substitute classifier as the Student model. Next, we randomly select Z% of images from each class of the dataset and use these images to train a Deep Convolutional Generative Adversarial Network (DCGAN) [72]. After training the DCGAN, we probe the Teacher model with the images generated using DCGAN and label them as the predicted class. We then augment these labeled images to the Z% amount of original images to create a pseudo-labeled dataset. In order to have an equal data distribution between the original and pseudo-labeled dataset, we made the number of images per class in the pseudo-labeled dataset to be the same as the original dataset used for training the Teacher model. Finally, the pseudo-labeled dataset is used for training the Student model (i.e., our substitute classifier).

In KD-1, since we randomly select Z% of images from each class to train the DCGAN and create the pseudo-labeled dataset, there is no guarantee that the selected images will statistically represent an accurate distribution of each class. To address this issue we perform k-fold cross validation by selecting different Z% amount of images (see Section 4.5). In our approach we set the value of Z to be 25% and 50% and use the Fashion-MNIST [78], CIFAR-10 [79], and GTSRB [80] datasets to evaluate KD-1.

3.2.2 Knowledge Distillation-2 (KD-2)

In KD-2, we do not have any knowledge about the images or their data distribution, but we know the names of all the classes and the total number of classes in the dataset as shown in Fig. 2(b). Hence, we search for images belonging to those classes from publicly available datasets. First, we create a pseudo-labeled dataset by probing the Teacher model with the images from the public domain and label these images with the predicted class. It should be noted that since we already know the names of the class the image belongs to, we do not need to further probe the Teacher model and relabel the images, but prior work done by [81] - [83] shows that when we train the Student model with images that were manually annotated by humans, the classification accuracy with respect to the Teacher model is lower compared to training the Student model with images that were annotated entirely by the Teacher model. The reason for this is that images that are misclassified by the Teacher model add a regularizing effect while training the Student model, thus, resulting in efficient functionality transfer [83].

Finally, after creating the pseudo-labeled dataset, we use it for training the Student model. In this paper we use the MIO-TCD [60] and Tiny ImageNet [84] dataset to evaluate the KD-2 approach.

3.2.3 Knowledge Distillation-3 (KD-3)

In this setting we know only the total number of classes in the training dataset and the domain of the dataset as shown in Fig. 2(c). We neither have any knowledge of the images in the training dataset nor we know the names of the classes. This scenario occurs in large scale identification applications such as face recognition and pedestrian re-identification. Based on this we scour for images that belong to this domain from publicly available datasets. Similar to KD-2 we probe the black box classifier with these images in order to create the pseudo-labeled dataset and then use the pseudo-labeled dataset to train the substitute classifier. We evaluate the KD-3 approach using the MS-Celeb dataset [59] that consists of facial images of over 99,000 celebrities. It should be noted that in this paper we evaluate the KD-2 and KD-3 approaches in two different settings: i) when there is no overlap and ii) when there is 50% overlap between the black box training dataset and the pseudo-labeled dataset (see Table 6).

Comments: In both KD-1 and KD-2, although the Student model is trained on a dataset that is entirely different from the dataset used for training the Teacher model, we are still able to distill some of the learned features from the Teacher model. The reason for this is that the Teacher model is assumed to be a deterministic model meaning that, after training, the features learned are fixed and do not change over time. Hence, when we probe a given image X, the resulting prediction f(X) = Y will never change and with a considerably large and diverse pseudo-labeled dataset, the student model is able to distill the learned features of the Teacher model and achieve good classification accuracy. This observation is particularly advantageous to black box applications that have an abundance of crowd-sourced images from various publicly available datasets such as the field of face biometrics and vehicle classification. Note that prior to [94], this observation had not been addressed and can effectively be used for distilling the knowledge of robust Black box classifiers into edge devices that usually have compressed and shallow CNN architectures [95] - [98].

3.3 Uncertainty Prediction via Bayesian Learning

In the domain of adversarial defense, it is very important to know the amount of adversarial perturbation that still remains in the output image after it passes through any defense algorithm. Bayesian methods offer a principled way to represent these uncertainties in a model and can be utilized to quantify a model’s confidence in its prediction [73]. Deep learning models f(·) consists of a set of weights w that are optimized on a labeled dataset D = \{x_i, y_i\}_{i=1}^N, where x_i and y_i are the input data and corresponding ground-truth, respectively. Bayesian inference involves learning a posterior distribution over the weights p(w|D) which is used for predicting unseen observations:

\[ p(y|x, D) = \int p(y|x, w) p(w|D)dw \] (1)

The above integral is intractable because of the sheer number of parameters in deep learning models. To overcome this, in our approach we design the Bayesian CNN using Bayes by Backprop [75]. Bayes by Backprop is a variational
inference to Bayesian neural networks where the posterior is assumed to be a diagonal Gaussian distribution which assumes independence among the variables. The Gaussian posterior \( q_\theta(\omega|D) \) is defined to be as similar as possible to the original posterior \( p(\omega|D) \) when measured by the KL divergence [76]. Based on this the optimal parameters are defined as:

\[
\theta_{\text{opt}} = \arg \min_{\theta} KL(q_\theta(w|D) || p(w)) - \mathbb{E}_{q(\theta|w)}(\log p(D|\theta) + \log p(D))
\]  

(2)

After learning the approximate posterior distribution we compute two uncertainty metrics namely: 1) Aleatoric, and 2) Epistemic uncertainties [77] which are given by:

\[
\text{Aleatoric Uncertainty} = \frac{1}{S} \sum_{i=1}^{S} \text{diag}(\hat{g}_i) - \hat{g}_i \hat{g}_i^T
\]

(3)

\[
\text{Epistemic Uncertainty} = \frac{1}{S} \sum_{i=1}^{S} (\hat{g}_i - \hat{g})(\hat{g}_i - \hat{g})^T
\]

(4)

where, \( S \) is the number of samples drawn from the posterior distribution, \( \hat{g} = \frac{1}{S} \sum_{i=1}^{S} \hat{g}_i \) and \( \hat{g}_i = f_{w_i}(x) \). It should be noted that we trained the Bayesian CNN using the same pseudo-labeled dataset used for training the Substitute model as described in Section 4.4 and Table 6.

**Aleatoric Uncertainty**: It is a measure for the variation of data. This value increases if certain classes are heavily unbalanced or lack data. Adversarial images have been shown to lie in the high frequency and low probability density regions [27], [28], which is similar to highly unbalanced and long-tailed datasets. This is also a reason why adversarial training [3] is an effective white box defense.

**Epistemic Uncertainty**: It is caused by the model itself. It is the ability of a model to learn robust and representative features which depends on its architecture and parameters. This value increases in the presence of adversarial attacks.

### 3.3.1 When is an image adversarial?

After learning the posterior distribution \( q_\theta(\omega|D) \), we find the minimum uncertainty for an adversarial image. For this we generated adversarial images with the smallest perturbation (i.e. \( \epsilon = 1/255 \)) for the Substitute model using three well known attacks: IFGSM [3], BIM [34], and PGD [100]. Next, these adversarial images are transferred to the Bayesian CNN and we compute the average (\( \mu \)) and standard deviation (\( \sigma \)) of the Epistemic and Aleatoric uncertainties. Finally, we set two thresholds \( T_1 \) and \( T_2 \) given by:

\[
T_1 = \mu(\text{Aleatoric}) - 3\sigma(\text{Aleatoric})
\]

(5)

\[
T_2 = \mu(\text{Epistemic}) - 3\sigma(\text{Epistemic})
\]

(6)

For a given image if at least one uncertainty is greater than its corresponding threshold, we classify it as an adversarial image and pass it as input to our ensemble of iterative defenses. We chose the threshold values for \( T_1 \) and \( T_2 \) to be as shown in Eq. (5) and (6) because, although \( \mu - \sigma \) is the least amount of uncertainty corresponding to annotated training dataset \( D \), there could be unseen attacks in the real-world where the uncertainty value is below \( \mu - \sigma \). To accommodate these unseen adversarial images we set \( T_1 \) and \( T_2 \) to be two standard deviations lower than \( \mu - \sigma \) as shown in Eq. (5) and (6).

### 3.4 Ensemble of Iterative Adversarial Defenses

The adversarial defenses used in this paper are auxiliary generative networks that can be used in conjunction with any classifier as a pre-processing step without modifying the structure of the classifier. These approaches do not assume any classifier model and are model agnostic. In this paper we chose to use MagNet [26], PixelDefend [27], Shield-Nets [28], and Defense-GAN [101] in our ensemble because they achieve state-of-the-art results for white and black box defense. It should be noted that the above mentioned defenses are all single-step defenses and cannot quantify if a purified image is adversarial or not. However, using our defense framework, we are able to convert these single-step defenses into iterative defenses and quantify the amount of adversarial component remaining after each iteration of purification. Additionally, each of these individual defenses are trained independently and this provides flexibility to alter the structure of the ensemble, i.e., add/remove individual defenses without affecting the entire framework (see Section 4.5.1). Unlike the related work done for adversarial defense as shown in Table 1 and 2, we chose to use an ensemble of defenses for two reasons: 1) we show that an ensemble of defenses is able to statistically defend against adversarial attacks much better compared to individual defenses (See Section 4.5) and, 2) the output of the ensemble (the average purified image) acts as a momentum which helps prevent individual deterministic defenses from getting stuck at a local minimum [103].

#### 3.4.1 Empirical Determination of the Number of Iterations for Purification

In Fig. 1, \( M \) is the maximum number of iterations an image can be purified before being (a) passed as input to the black box CNN or (b) rejected. The reason for this is that after each iteration, the amount of purification done decreases and after \( M \) iterations the ensemble is not able to further purify the image. This situation arises when the adversarial perturbation (\( \epsilon \)) is very high causing the adversarial noise to dominate the image, which makes it very difficult to purify the image. This is a potential threat that an adversary could use to lock our defense in a state of an infinite loop of purification, thus crashing the defense framework. To eliminate this threat, we set a threshold (\( M \)) on the maximum number of iterations of purification before rejecting an image. In order to empirically determine the value of \( M \), we attacked the Substitute CNN using the IFGSM [3], BIM [34], and PGD [100] attacks with \( \epsilon = 0.05, 0.1, \) and 0.2. We chose the values of \( \epsilon \) within the range of 0.05 - 0.2 because this is the range an adversarial attack is likely to fool a human observer and adversarial images with \( \epsilon > 0.2 \) make the resulting images more discernible to the human eye [28], [94]. The resulting adversarial images are then passed as input to our ensemble of image purifiers for six iterations of purification. From this we quantify the amount of purification done by measuring the \( L_2 \) distance between the input and output images at every iteration. Fig. 3 shows the plots for the amount of purification, Aleatoric, and Epistemic uncertainties Vs. the number of iterations of purification for the GTSRB [80], MIO-TCD [60], Tiny ImageNet [84], and MS-Celeb [59] datasets with \( \epsilon = 0.05 \).
Fig. 3: (a) - (d) show the average amount of purification Vs. the number of iterations of purification, (e) - (h) show the average Aleatoric uncertainties and, (i) - (l) show the average Epistemic uncertainties for the GTSRB [80], MIO-TCD [60], Tiny-ImageNet [84], and MS-Celeb [59] datasets, respectively.
and 0.1. From Fig. 3(a) - (d) we can see that after 3 iterations, the amount of purification does not significantly change for the GTSRB, and Tiny ImageNet datasets. Hence, we set the value of M = 3 for these datasets. Similarly, for the MIO-TCD and MS-Celeb datasets we set M = 4. It is also interesting to note that, as the dimension of the input space increases (see Table 5), the value of M also increases.

Note: Since the Bayesian uncertainties do not significantly change after M iterations (see Fig. 3), in order to save computational resources, we can purify all input images for M iterations rather than verifying after each iteration if the given image is still adversarial or not. By doing so, we also risk the possibility of purifying non-adversarial images and, this could affect the final classification (refer to Section 4.5.1). Additionally, without the adversarial detector we further invite the possibility of brute force attacks (i.e., attacks with severe adversarial perturbation). Prior work done in [94] shows that such images cannot be purified within M iterations and cause misclassifications, which in turn could potentially compromise safety critical automated applications. Ideally, such adversarial images would be rejected by the adversarial detector after M iterations of purification (see Fig. 2b in the supplementary material).

Hence after determining the value of M for a given dataset, applications that do not pose significant risk may not need the adversarial detector because the damage caused by brute force adversarial attacks and misclassifying non-adversarial images is negligible compared to safety critical automated applications such as remote surveillance [52] and human biometrics [115], where it is critical to have the adversarial detector.

3.4.2 Theoretical Lower Bound for the Amount of Purification in the Input Space

For a given image X ∈ R^P×Q from dataset D, where P×Q is the number of pixels in the image, an ε-bounded adversarial image is denoted as X + δ, where δ belongs to the l_p bounded neighborhood Δ = {δ ∈ R^P×Q | ∥δ∥_p ≤ ε} to X. The individual adversarial defenses π_ω(X′|X + δ) are expected to map the adversarial images from adversarial regions back to a space within Δ, where ω are the trainable parameters of the defense and X′ is the output of the defense. Adversarial attacks on any classification task with a loss function of L(X′, Y; θ), where, θ are the parameters of the black box classifier, can be achieved by optimizing,

$$\arg \max_{\delta \in \Delta} \int_{\Delta} \pi_\omega(X′|X + \delta) L(X′, Y; \theta) dX′.$$  

(7)

The loss function of the adversarial defense (L_{Def}) can be expressed as the marginalized expectation:

$$L_{Def} = \mathbb{E}_{(X,Y) \sim D} \int_{\Delta} \mathbb{E}_{X′ \sim \pi_\omega(X′|X + \delta)} [L(X′, Y; \theta)] p(\delta) d\delta$$  

(8)

where p(δ) represents the distribution of adversarial samples in Δ. The theoretical possibility of the adversarial defense to neutralize the adversarial images is supported by the following theorem:

**Theorem 1.** Assume L(X′, Y; θ) is continuous in X + Δ and π_ω(X′|X + δ) is supported on X + Δ, there exists a lower bound for L_{Def} in space Δ. If π_ω(X′|X + δ) = δ_{Dirac}(X′ - X - \beta_0), L_{Def} reaches the lower bound, where \beta_0 = \arg \min_{\beta \in \Delta} L(X + \beta, Y; \theta).

**Proof of Theorem 1:** Since L(X + \beta, Y; \theta) is continuous and Δ is compact, \beta_0 = \arg \min_{\beta \in \Delta} L(X + \beta, Y; \theta), where \beta_0 is the lower bound space within Δ beyond which an image cannot be further purified. Assume p(δ) is a distribution that only supports in Δ and π_ω supports in X + Δ,

$$L_{Def} = \mathbb{E}_{(X,Y) \sim D} \int_{\Delta} \mathbb{E}_{X′ \sim \pi_\omega(X′|X + \delta)} [L(X′, Y; \theta)] p(\delta) d\delta$$  

(9)

$$= \mathbb{E}_{(X,Y) \sim D} \int_{\Delta} p(\delta) d\delta \int_{X + \Delta} dX′ \pi_\omega(X′ | X + \delta) L(X′, Y; \theta)$$  

(10)

$$\geq \mathbb{E}_{(X,Y) \sim D} \int_{\Delta} p(\delta) d\delta \int_{X + \Delta} dX′ \pi_\omega(X′ | X + \delta) \left( \min_{X′ \sim \Delta} L(X′, Y; \theta) \right)$$  

(11)

$$= \mathbb{E}_{(X,Y) \sim D} \int_{\Delta} p(\delta) d\delta \int_{X + \Delta} dX′ \pi_\omega(X′ | X + \delta) \left( L(X + \beta_0, Y; \theta) \right)$$  

(12)

$$= \mathbb{E}_{(X,Y) \sim D} \int_{\Delta} p(\delta) d\delta L(X + \beta_0, Y; \theta)$$  

(13)

$$= \mathbb{E}_{(X,Y) \sim D} L(X + \beta_0, Y; \theta)$$  

(14)

The equality is satisfied when π_ω(X′|X + δ) = δ_{Dirac}(X′ - X - \beta_0). Theorem 1 is further empirically supported by Fig. 3. In Fig. 3(a) - (d) it can be seen that after M iterations the amount of purification significantly decreases and the image can no longer be purified. This also means that after M iterations, the adversarial image X_{adv} is transformed/purified to X′ ∈ \beta_0, beyond which it cannot be further purified. Additionally, from Fig. 3 it can be seen that as the amount of purification decreases after each iteration (Fig. 3(a) - (d)), the aleatoric and epistemic uncertainties also gradually decrease (Fig. 3(e) - (l)) and after M iterations they do not significantly vary because the amount of purification does not change. This further emphasizes that the resulting purified image X′ has been reached the lower bound within the input space \beta_0 ∈ Δ and cannot be further purified.

4 Experimental Results

4.1 CNN Architectures

Table 4 shows the architectures of the CNN used in our approach for adversarial defense. For fair comparisons, the CNN architectures in Table 4 are the same as those reported in [28] and [94]. We evaluate our black box defense by creating an adversarial Substitute CNN and transfer the adversarial images generated for the adversary’s Substitute as input to our defense [3], [99]. In Table 4 we used the same CNN architecture and training data for our defense’s Substitute as well as the adversary’s Substitute. By doing so
TABLE 4: CNN architectures

<table>
<thead>
<tr>
<th>CNN Architectures</th>
<th>ResNet</th>
<th>VGG</th>
</tr>
</thead>
<tbody>
<tr>
<td>Target Black Box CNN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Defense Substitute CNN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Bayesian CNN</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Adversary’s Substitute CNN</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

we are giving the adversary an equal amount of knowledge as to our defense Substitute model in order to have a fair evaluation of our defense. Although we chose the architecture of the black box classifier to be ResNet [105], it should be noted that our framework in general can be adopted to other classifier as well. In fact, our prior work done in [94], shows results for a black box classifier with VGG architecture [106].

4.2 Datasets

We evaluate our defense on six public benchmark datasets namely: Fashion-MNIST [78], CIFAR-10 [79], GTSRB [80], MIO-TCD [60], Tiny ImageNet [84], and MS-Celeb [59]. Table 5 shows a summary of all the datasets and Fig. 4 shows examples of images from the six datasets used in this paper.

- **Fashion-MNIST:** Fashion MNIST was designed to be a much more difficult and drop-in replacement for the MNIST dataset [87]. The dataset consists of 60,000 training and 10,000 testing gray-scale images of size 28x28 distributed evenly into 10 different classes.

- **CIFAR-10:** CIFAR-10 is another widely used dataset that consists of 50,000 training and 10,000 testing RGB images of size 32x32 distributed evenly into 10 different classes.

- **GTSRB:** The GTSRB dataset consists of 51,883 images of traffic sign in Germany. The dataset is split into 43 different classes with 39,295 training and 12,631 testing images.

- **MIO-TCD:** The MIO-TCD dataset consists of two sub-datasets namely: MIO-TCD Classification and MIO-TCD Localization. The dataset consists of images of vehicles captured from various traffic cameras installed on the roads all over the USA and Canada. The dataset is split into 11 classes with 519,164 training and 129,795 testing images for the classification subset. The localization subset is split into 11 classes with 110,000 training and 27,743 testing images.

- **Tiny-ImageNet:** The Tiny-Imagenet consists of 200 classes with 500 training and 50 testing images per class. The 200 classes were collected from the Synsets of the WordNet [86] hierarchy and is similar to the ImageNet dataset [85].

- **MS-Celeb:** The MS-Celeb dataset consists of approximately 9.5M images for 99,892 celebrities. It has been shown that this dataset is extremely noisy with many incorrect annotations [88] - [91]. In order to reduce the noise due to incorrect annotation, we followed the approach proposed by Jin et al. [93] which uses the community detection algorithm [92]. Based on this, the authors provided a list of correctly annotated images and showed that approximately 97.3% of images in the dataset are correctly labeled. This results in a total of approximately 6.5M images for 94,682 celebrities.

4.3 Threat Models

In this sub-section we define the adversarial attacks that are used for evaluating our defense. For a given test image-label pair \((X,Y)\), adversarial attacks find a perturbation \(\delta\) with \(\|\delta\|_\infty \leq \epsilon\) such that a deep learning classifier \(f(\cdot)\) results in \(f(X+\delta) \neq Y\). \(\epsilon\) is a hyper-parameter that sets the perturbation limit for each pixel in \(X\) on the color scale.

- **Basic Iterative Method (BIM)** [34]: This attack extends the FGSM attack [3] by iterating it multiple times with a small step size.

\[
X_{n+1}^{adv} = \text{Clip}_\epsilon(X_n + \alpha \cdot \text{sign}(\nabla_X L(X_n,Y)))
\]  

(16)

- **Projected Gradient Descent** [100]: This attack computes the gradient in the direction of the highest loss and projects it back to the \(l_p\) norm around the sample.

\[
X_{n+1}^{adv} = \prod^\epsilon(X_n + \alpha \cdot \text{sign}(\nabla_X L(X_n,y)))
\]  

(17)

In eq. (15) \(-\) (17), \(X_{n+1}^{adv}\) is the resulting adversarial image, \(\nabla_X L(X,Y)\) is the loss function used to train the CNN, \(\alpha\) is the iterative step size, \(\text{Clip} (\cdot)\) and \(\prod (\cdot)\) are the clipping and projection functions, respectively.

4.4 Performance Evaluation of the Proposed KD Approaches

In this sub-section we evaluate the KD-1 approach using the Fashion-MNIST [78], CIFAR-10 [79], and GTSRB [80] datasets. We evaluate KD-2 using the MIO-TCD [60] and Tiny-ImageNet [84] datasets and finally, we evaluate KD-3 using the MS-Celeb [59] dataset.

4.4.1 Knowledge Distillation on the MIO-TCD Dataset

This dataset consists of two parts: 1) classification dataset, and 2) localization dataset. We trained the Teacher model using the MIO-TCD classification dataset and used the localization dataset to create our pseudo-labeled dataset for training the Student model. It should be noted that in the MIO-TCD classification dataset, we ignored the class “Background” because this class does not belong in the
localization dataset [71]. Based on this we probed the Teacher model and created the pseudo-labeled dataset such that each class had at least 1,713 images.

4.4.2 Knowledge Distillation on the Tiny ImageNet Dataset

The Tiny ImageNet dataset consists of 200 classes and this dataset is used for training the Teacher model and the pseudo-labeled dataset is created using the ImageNet dataset [85]. It should be noted that we use only those corresponding 200 classes in the ImageNet dataset to create the pseudo-labeled dataset. Based on this we probed the Teacher model and created the pseudo-labeled dataset such that each class had at least 487 images.

4.4.3 Knowledge Distillation on the MS-Celeb Dataset

In order to train the Teacher model, we manually selected 100 celebrities that had at least 100 images after discarding images that had extremely skewed poses and celebrities wearing sunglasses. A complete list of all of the selected facial identities used for training the Teacher model is provided in the supplementary material. We denote this dataset as $Q_{1:1:100}$ ($Q_i$ is the identity of the celebrity) and it is used for training the Teacher model. In order to train the Student model we first create a pseudo-labeled dataset by probing the Teacher model with images of celebrities that do not belong in $Q_{1:1:100}$ and labeled the images with the predicted class. We denote this pseudo-labeled dataset as $Q_{1:1:100:∞}$ and it should be noted that the dataset $Q_{1:1:100}$ and $Q_{1:1:100:∞}$ contain images of different celebrities and their data distributions do not overlap (ignoring the noise due to incorrect annotations). Based on this we augmented 3,000 pseudo-labeled images per class in the $Q_{1:1:1:∞}$ dataset. In the MIO-TCD and MS-Celeb datasets we chose the number of augmented images in the pseudo-labeled dataset to be at least 1,713 and 487 images, per class, respectively, because this is the maximum number of images possible for the class with the least number of images. Table 6 shows the baseline performance comparison between the Student model (defense substitute classifier) and the Teacher model (black box classifier). In Table 6 although the Teacher model outperforms the Student model, it can be seen that as the overlap between the Teacher model’s training dataset and the student model’s pseudo-labeled dataset increases, the performance of the Student model also increases. For the following experimental results in Section 4.5, we train our defense framework to defend the Student model in Table 6 and then deploy the trained defense framework to defend the black box classifier.

4.5 Performance Evaluation of Adversarial Detection using Bayesian Uncertainties

Table 7 shows the performance evaluation of our adversarial detection using Bayesian uncertainties. For the results in Table 7 we created adversarial images using the IFGSM attack [3] with $\epsilon = 0.05$ and 0.2 and passed both the adversarial and non-adversarial images as inputs to the Bayesian CNN. From Table 7 when $\epsilon = 0.05$, we can see that our approach achieves at least 69.05% accuracy on the Tiny-ImageNet dataset [84] and at most 93.51% on the Fashion-MNIST dataset [78]. The reason for this is that the Fashion-MNIST dataset contains images with one foreground object in a

---

**TABLE 5:** Summary of the datasets used in this paper.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Image size</th>
<th>Domain</th>
<th>Grayscale/RGB</th>
<th>Number of classes</th>
<th>Balanced classes?</th>
<th>Training data</th>
<th>Testing data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fashion-MNIST</td>
<td>28 x 28</td>
<td>Fashion</td>
<td>Grayscale</td>
<td>10</td>
<td>Yes (6,000)</td>
<td>60,000</td>
<td>10,000</td>
</tr>
<tr>
<td>CIFAR-10</td>
<td>32 x 32</td>
<td>General</td>
<td>RGB</td>
<td>10</td>
<td>Yes (5,000)</td>
<td>50,000</td>
<td>10,000</td>
</tr>
<tr>
<td>GTSRB</td>
<td>64 x 64*</td>
<td>German traffic signs</td>
<td>RGB</td>
<td>43</td>
<td>No</td>
<td>39,252</td>
<td>12,630</td>
</tr>
<tr>
<td>Tiny ImageNet</td>
<td>64 x 64</td>
<td>General</td>
<td>RGB</td>
<td>200</td>
<td>Yes (500)</td>
<td>100,000</td>
<td>10,000</td>
</tr>
<tr>
<td>MS-Celeb</td>
<td>128 x 128*</td>
<td>Celebrity faces</td>
<td>RGB</td>
<td>100</td>
<td>No</td>
<td>8,933</td>
<td>2,177</td>
</tr>
<tr>
<td>MIO-TCD</td>
<td>224 x 224**</td>
<td>USA &amp; Canada vehicles</td>
<td>RGB</td>
<td>10</td>
<td>No</td>
<td>399,164</td>
<td>129,796</td>
</tr>
</tbody>
</table>

*Images are resized to the specified size. ** Shorter side of the image is resized to 256 while maintaining the aspect ratio, and then center is cropped to the size of 224 x 224. † The Number in parentheses indicates the number of images per class.

**TABLE 6:** Performance evaluation and comparison of our KD approaches with respect to the Teacher (black box) classifier.

<table>
<thead>
<tr>
<th>Training dataset (Teacher classifier)</th>
<th>Testing dataset (Teacher &amp; student classifier)</th>
<th>Teacher accuracy (%)</th>
<th>KD algorithm</th>
<th>Pseudo-labeled training dataset (student classifier)</th>
<th>Student accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fashion-MNIST training dataset</td>
<td>Fashion-MNIST testing dataset</td>
<td>93.51</td>
<td>KD-1 ($Z = 25%$)</td>
<td>$Z$</td>
<td>$77.16 \pm 1.35$</td>
</tr>
<tr>
<td>CIFAR-10 training dataset</td>
<td>CIFAR-10 testing dataset</td>
<td>95.31</td>
<td>KD-2 ($Z = 25%$)</td>
<td>$Z$</td>
<td>$89.67 \pm 1.32$</td>
</tr>
<tr>
<td>GTSRB training dataset</td>
<td>GTSRB testing dataset</td>
<td>96.45</td>
<td>KD-2 ($Z = 25%$)</td>
<td>$Z$</td>
<td>$88.42 \pm 1.26$</td>
</tr>
<tr>
<td>MIO-TCD classification training dataset</td>
<td>MIO-TCD classification testing dataset</td>
<td>94.68</td>
<td>KD-2</td>
<td>MIO-TCD classification</td>
<td>$84.51$</td>
</tr>
<tr>
<td>Tiny ImageNet training dataset</td>
<td>Tiny ImageNet testing dataset</td>
<td>Top 1: 46.79 Top 5: 72.30</td>
<td>KD-2</td>
<td>200 classes from ImageNet training dataset</td>
<td>Top 1: 40.80 Top 5: 64.37</td>
</tr>
<tr>
<td>MS-Celeb 75% of 100 celebrities for training</td>
<td></td>
<td>90.32 ± 1.56</td>
<td>KD-3</td>
<td>Every other celebrity</td>
<td>$74.35 \pm 3.72$</td>
</tr>
</tbody>
</table>
TABLE 7: Performance evaluation of adversarial detection using Bayesian uncertainties.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>IFGSM attack with $\epsilon = 0.05$</th>
<th>IFGSM attack with $\epsilon = 0.2$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy (%)</td>
<td>False Positive (%)</td>
</tr>
<tr>
<td>F-MNIST</td>
<td>93.51</td>
<td>3.95</td>
</tr>
<tr>
<td>CIFAR-10</td>
<td>79.34</td>
<td>15.26</td>
</tr>
<tr>
<td>GTSRB</td>
<td>91.06</td>
<td>6.36</td>
</tr>
<tr>
<td>MIO-TCD</td>
<td>87.74</td>
<td>9.77</td>
</tr>
<tr>
<td>Tiny-ImageNet</td>
<td>69.05</td>
<td>23.94</td>
</tr>
<tr>
<td>MS-Celeb</td>
<td>72.93</td>
<td>20.15</td>
</tr>
</tbody>
</table>

completely black background compared to Tiny-ImageNet which has a more complex background (please refer to the supplementary material for visual comparisons on all of the datasets). Furthermore, in Table 7 when $\epsilon = 0.05$, the false positive is significantly higher than the false negative, meaning that more non-adversarial images were predicted as adversarial images than vice-versa. In the context of adversarial detection it is better to have a higher false positive rate compared to false negative rate because the damage caused by misclassifying an adversarial image is more than misclassifying a non-adversarial image [112] - [114]. Moreover, as the adversarial perturbation ($\epsilon$) increases from 0.05 to 0.2, the Signal-to-Noise ratio (SNR) of the image starts to decrease making the adversarial noise more easily visible to the human eye as well as easily detected using the Bayesian CNN [94]. As shown in Fig. 3(a) - 3(i), after $M$ iterations of purification, these images would automatically be rejected by the system if the predicted uncertainty values are above the corresponding thresholds.

4.5.1 How does purifying non-adversarial images affect the classification?

In this sub-section we perform experiments to observe how purifying the non-adversarial images affect the final classification. Table 8 shows the total number of non-adversarial images (i.e., images that failed to fool the black box classifier) using the IFGSM attack with $\epsilon = 0.05$ and the total number of images predicted correctly after one iteration of purification.

TABLE 8: Performance evaluation of purifying non-adversarial images.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Total No. of non-adversarial images</th>
<th>Images correctly classified after purification*</th>
</tr>
</thead>
<tbody>
<tr>
<td>F-MNIST</td>
<td>3920</td>
<td>3871 (98.8%)</td>
</tr>
<tr>
<td>CIFAR-10</td>
<td>2617</td>
<td>2588 (98.9%)</td>
</tr>
<tr>
<td>GTSRB</td>
<td>4042</td>
<td>3965 (98.1%)</td>
</tr>
<tr>
<td>Tiny-ImageNet</td>
<td>1276</td>
<td>1257 (98.5%)</td>
</tr>
<tr>
<td>MIO-TCD</td>
<td>26439</td>
<td>26305 (99.5%)</td>
</tr>
<tr>
<td>MS-Celeb</td>
<td>562</td>
<td>547 (97.4%)</td>
</tr>
</tbody>
</table>

*corresponds to images correctly classified with respect to the predictions of the black box before and after purification.

In Table 8, a correctly classified image is defined as: for a given image $X$, the predicted label of black box $f$ is $Y$, (i.e. $f(X) = y$). The corresponding purified image $X'$ has a black box predicted label $Y'$, (i.e. $f(X') = Y'$). If $Y = Y'$ then the image is correctly classified, but it should be noted that $Y$ and $Y'$ do not necessarily correspond to the human annotated ground-truth. From Table 8 we can see that the least possible accuracy is 97.4% which indicates that even after one iteration of purification, the non-adversarial images are not significantly altered to cause severe misclassification.

4.6 Performance Evaluation and Comparison of our Defense Against Adversarial Attacks

Tables 9 - 14 show the performance and comparison of our approach against the state-of-the-art on the GTSRB [80], Tiny-ImageNet [84], MIO-TCD [60], and MS-Celeb [59] datasets described in Section 4.2. Tables 1 - 4 in the supplementary material show the performance and comparison of our approach on the Fashion-MNIST [78] and CIFAR-10 [79] datasets. Note that in Tables 9 - 14 all of the single-step adversarial defenses (MagNet [26], PixelDefend [27], ShieldNets [56], and Defense-GAN [101]) have been converted into iterative defenses using our framework. From Tables 9 - 14 it can be seen that our ensemble of defenses outperforms all the stand-alone defenses. Although, as the perturbation limit ($\epsilon$) of the adversarial attack increases, the performance of all the approaches in Tables 9 - 14 gradually decrease, but there is also a gradual increase in the Bayesian uncertainty metrics. This means that even if an adversary tries to break our defense by significantly increasing the value of ($\epsilon$), the resulting adversarial image would still be rejected because the uncertainty values of the image are beyond the threshold limits $T_1$ and $T_2$. By increasing the value of ($\epsilon$) in order to break a defense may seem trivial, however, it is still a foundational problem in online applications where there is no human in the loop [38], [39]. In Table 10, we report the Top 5 classification accuracy for the Tiny ImageNet as this is the metric that is used for evaluating the dataset. Additionally, the adversarial images used in Table 10 were generated such that the top 5 predictions for an adversarial image do not contain the ground-truth label ($Y$), i.e., $Y \notin \{Y_i| i \leq 5\}$.

4.6.1 Ablation Study for Evaluating Different Combinations of Ensembles of Iterative Defenses

In this sub-section we perform an ablation study to evaluate different combinations of ensembles of adversarial defenses and compare their performance. For this purpose we chose to use the Fashion-MNIST [78] and CIFAR-10 [79] datasets and the following adversarial defenses: MagNet [26], PixelDefend [27], ShieldNets [56], and Defense-GAN [101]. Table 15 shows the comparisons of different ensembles.

From Table 15 it can be seen that using the ensemble MPSD and MSD achieves the best classification accuracy on the Fashion-MNIST dataset against the IFGSM and BIM attack, respectively. Similarly, using the ensemble MPS achieves the best classification accuracy on the CIFAR-10 dataset against the IFGSM and BIM attacks. It is also interesting to note that although some of the combinations of ensembles such as MS on the CIFAR-10 dataset and MSD on the Fashion-MNIST dataset had lower aleatoric and epistemic uncertainties compared to MPS and MPSD, respectively, their classification accuracy was lower by at
TABLE 9: Performance comparison of our Defense on the GTSRB dataset using the KD-1 approach with Z = 25% and 50%.

<table>
<thead>
<tr>
<th>Attack</th>
<th>Defense</th>
<th>GTSRB KD 1 (Z = 25%)</th>
<th>GTSRB KD 1 (Z = 50%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy (%)</td>
<td>Average Aleatoric uncertainty</td>
<td>Average Epistemic uncertainty</td>
</tr>
<tr>
<td></td>
<td>T₁ = 0.0541</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>T₂ = 0.0583</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

- **No Defense**
  - Accuracy: 28.93
  - Average Aleatoric uncertainty: -
  - Average Epistemic uncertainty: -

- **IFGSM**
  - MagNet: 83.50 ± 3.47
  - ShieldNets: 86.19 ± 3.20
  - PixelDefend: 76.27 ± 2.68
  - Ensemble: 89.26 ± 3.07

- **BIM**
  - MagNet: 81.79 ± 3.28
  - ShieldNets: 84.25 ± 3.79
  - PixelDefend: 79.03 ± 3.14
  - Ensemble: 86.18 ± 3.44

- **PGD**
  - MagNet: 82.83 ± 3.81
  - ShieldNets: 86.42 ± 3.28
  - PixelDefend: 80.43 ± 3.50
  - Ensemble: 88.41 ± 3.28

TABLE 10: Performance comparison of our Defense on the Tiny ImageNet dataset using the KD-2 approach.

<table>
<thead>
<tr>
<th>Attack</th>
<th>Defense</th>
<th>Tiny ImageNet - Black box training data</th>
<th>Tiny ImageNet + 50% Pseudo-labeled training data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Top 5 Accuracy (%)</td>
<td>Average Aleatoric uncertainty</td>
<td>Average Epistemic uncertainty</td>
</tr>
<tr>
<td></td>
<td>T₁ = 0.0685</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>T₂ = 0.0687</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

- **No Defense**
  - Accuracy: 12.76
  - Average Aleatoric uncertainty: -
  - Average Epistemic uncertainty: -

- **IFGSM**
  - MagNet: 42.66
  - ShieldNets: 45.83
  - PixelDefend: 38.29
  - Ensemble: 46.34

- **BIM**
  - MagNet: 45.29
  - ShieldNets: 47.93
  - PixelDefend: 41.75
  - Ensemble: 49.02

- **PGD**
  - MagNet: 40.37
  - ShieldNets: 41.28
  - PixelDefend: 35.16
  - Ensemble: 42.86

TABLE 11: Performance comparison of our Defense on the MIO-TCD classification dataset using the KD-2 approach with no overlap between the black box and pseudo-labeled training dataset.

<table>
<thead>
<tr>
<th>Attack</th>
<th>Defense</th>
<th>MIO-TCD Classification</th>
<th>MIO-TCD Localization</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Accuracy (%)</td>
<td>Average Aleatoric uncertainty</td>
<td>Average Epistemic uncertainty</td>
</tr>
<tr>
<td></td>
<td>T₁ = 0.0604</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>T₂ = 0.0689</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

- **No Defense**
  - Accuracy: 28.37
  - Average Aleatoric uncertainty: -
  - Average Epistemic uncertainty: -

- **IFGSM**
  - MagNet: 83.65
  - ShieldNets: 84.51
  - PixelDefend: 75.06
  - Ensemble: 85.80

- **BIM**
  - MagNet: 82.79
  - ShieldNets: 81.08
  - PixelDefend: 76.92
  - Ensemble: 84.16

- **PGD**
  - MagNet: 80.35
  - ShieldNets: 81.25
  - PixelDefend: 73.39
  - Ensemble: 82.48
TABLE 12: Performance comparison of our Defense on the MIO-TCD classification dataset using the KD-2 approach with 50% overlap between the black box and pseudo-labeled training dataset.

<table>
<thead>
<tr>
<th>Attack</th>
<th>Defense</th>
<th>$\epsilon$ = 0.05 (13/255)</th>
<th>$\epsilon$ = 0.1 (26/255)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T1 = 0.0580</td>
<td>T2 = 0.0656</td>
<td>T1 = 0.0580</td>
</tr>
<tr>
<td>No Defense</td>
<td>20.57</td>
<td>47.12</td>
<td>15.52</td>
</tr>
<tr>
<td>IFGSM</td>
<td>84.25 ± 2.18</td>
<td>0.0520 ± 0.0081</td>
<td>0.0509 ± 0.0081</td>
</tr>
<tr>
<td>ShieldNets</td>
<td>85.02 ± 1.43</td>
<td>0.0521 ± 0.0082</td>
<td>0.0562 ± 0.0082</td>
</tr>
<tr>
<td>PixelDefend</td>
<td>76.81 ± 2.06</td>
<td>0.0594 ± 0.0076</td>
<td>0.0592 ± 0.0082</td>
</tr>
<tr>
<td>Ensemble</td>
<td>86.36 ± 1.59</td>
<td>0.0513 ± 0.0072</td>
<td>0.0520 ± 0.0081</td>
</tr>
<tr>
<td>No Defense</td>
<td>18.36</td>
<td>-</td>
<td>14.21</td>
</tr>
<tr>
<td>BIM</td>
<td>84.58 ± 1.87</td>
<td>0.0501 ± 0.0082</td>
<td>0.0543 ± 0.0085</td>
</tr>
<tr>
<td>ShieldNets</td>
<td>84.27 ± 2.05</td>
<td>0.0519 ± 0.0083</td>
<td>0.0536 ± 0.0079</td>
</tr>
<tr>
<td>PixelDefend</td>
<td>77.14 ± 1.91</td>
<td>0.0585 ± 0.0081</td>
<td>0.0585 ± 0.0081</td>
</tr>
<tr>
<td>Ensemble</td>
<td>86.58 ± 1.54</td>
<td>0.0494 ± 0.0078</td>
<td>0.0511 ± 0.0082</td>
</tr>
<tr>
<td>No Defense</td>
<td>15.89</td>
<td>-</td>
<td>12.76</td>
</tr>
<tr>
<td>PGD</td>
<td>81.76 ± 1.51</td>
<td>0.0533 ± 0.0091</td>
<td>0.0556 ± 0.00771</td>
</tr>
<tr>
<td>ShieldNets</td>
<td>82.66 ± 1.38</td>
<td>0.0530 ± 0.0072</td>
<td>0.0539 ± 0.0082</td>
</tr>
<tr>
<td>PixelDefend</td>
<td>75.02 ± 1.76</td>
<td>0.0560 ± 0.0080</td>
<td>0.0574 ± 0.0082</td>
</tr>
<tr>
<td>Ensemble</td>
<td>83.92 ± 1.53</td>
<td>0.0527 ± 0.0094</td>
<td>0.0565 ± 0.0087</td>
</tr>
</tbody>
</table>

TABLE 13: Performance comparison of our Defense on the MS-Celeb dataset using the KD-3 approach with no overlap between the black box and pseudo-labeled training dataset.

<table>
<thead>
<tr>
<th>Attack</th>
<th>Defense</th>
<th>$\epsilon$ = 0.05 (13/255)</th>
<th>$\epsilon$ = 0.1 (26/255)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>T1 = 0.0572</td>
<td>T2 = 0.0620</td>
<td>T1 = 0.0572</td>
</tr>
<tr>
<td>IFGSM</td>
<td>25.81 ± 3.78</td>
<td>-</td>
<td>25.81 ± 3.78</td>
</tr>
<tr>
<td>ShieldNets</td>
<td>62.79 ± 3.77</td>
<td>0.0495 ± 0.0093</td>
<td>0.0544 ± 0.0087</td>
</tr>
<tr>
<td>PixelDefend</td>
<td>60.03 ± 2.19</td>
<td>0.0529 ± 0.0079</td>
<td>0.0579 ± 0.0074</td>
</tr>
<tr>
<td>Ensemble</td>
<td>70.38 ± 2.64</td>
<td>0.0416 ± 0.0072</td>
<td>0.0472 ± 0.0084</td>
</tr>
<tr>
<td>BIM</td>
<td>29.55 ± 4.16</td>
<td>-</td>
<td>22.43 ± 3.97</td>
</tr>
<tr>
<td>ShieldNets</td>
<td>63.84 ± 3.54</td>
<td>0.0547 ± 0.0075</td>
<td>0.0536 ± 0.0083</td>
</tr>
<tr>
<td>PixelDefend</td>
<td>64.43 ± 3.59</td>
<td>0.0468 ± 0.0077</td>
<td>0.0439 ± 0.0089</td>
</tr>
<tr>
<td>Ensemble</td>
<td>68.91 ± 3.17</td>
<td>0.0459 ± 0.0082</td>
<td>0.0498 ± 0.0064</td>
</tr>
<tr>
<td>No Defense</td>
<td>22.03 ± 4.52</td>
<td>-</td>
<td>17.94 ± 4.25</td>
</tr>
<tr>
<td>PGD</td>
<td>61.93 ± 3.48</td>
<td>0.0526 ± 0.0091</td>
<td>0.0562 ± 0.0085</td>
</tr>
<tr>
<td>ShieldNets</td>
<td>64.05 ± 2.27</td>
<td>0.0486 ± 0.0075</td>
<td>0.0510 ± 0.0084</td>
</tr>
<tr>
<td>PixelDefend</td>
<td>59.10 ± 2.12</td>
<td>0.0549 ± 0.0089</td>
<td>0.0570 ± 0.0080</td>
</tr>
<tr>
<td>Ensemble</td>
<td>65.78 ± 2.91</td>
<td>0.0491 ± 0.0073</td>
<td>0.0523 ± 0.0078</td>
</tr>
</tbody>
</table>

TABLE 14: Performance comparison of our Defense on the MS-Celeb dataset using the KD-3 approach with 50% overlap between the black box and pseudo-labeled training dataset.
least 1%. This observation indicates that although some purified images have slightly above average uncertainty metrics does not necessarily mean that these images are adversarial. A possible reason for this phenomenon is that since the Bayesian CNN is trained independently with almost no knowledge of the black box classifier’s weights or its training dataset, the features learned by these two CNNs could be slightly different from each other. Hence, an image that is adversarial to the Bayesian CNN may not necessarily be adversarial to the black box classifier. It should also be noted that this kind of situation is statistically more likely to happen only to the images that are close to the classification boundary in the CNN’s feature space and it is a very small percentage of the total dataset.

4.6.2 Performance against Black box Adversarial Attacks

In this sub-section we evaluate our defense against six black box adversarial attacks SimBA [102], NES [107], \( \mathcal{A} \)-attack [108], SPSA [109], Boundary [110], and ZOO [111]. NES [107] and SPSA [109] attacks use query limited attacks where the attacker has a query budget, within which the adversarial attack needs to be successful. The authors replace the gradient of the loss function with an estimate of the gradient, which is approximated by querying the classifier rather than computed by auto-differentiation. SimBA [102] is similar to NES attack [107] but rather than traversing in a specific direction of the gradient, they pick a random direction from a pre-specified set of orthogonal search directions, use the probability logit scores to check if it is pointing towards or away from the decision boundary, and perturb the image by adding the vector from the image. ZOO [111] estimates the gradient at each coordinate by finite differences and adopts C&W [4] for attacks based on the estimated gradient. \( \mathcal{A} \)-attack [108] does not estimate the gradient but learns a Gaussian distribution centered around the input such that a sample drawn from it is likely adversarial. A major difference between our defense and SimBA [102] is that our approach does not need any probability logit output and requires only the final class prediction, but for the sake of comparison we provide the logit probabilities as input.

Table 16 shows the performance of our defense against the six black box attacks on the GTSRB [80], MS-Celeb [59], and CIFAR-10 [79] datasets. In Table 16 we set the query limit \( L \) for all query based attacks to 1,000 queries per image and adversarial perturbation limit \( \epsilon = 1 \). We created the black box adversarial attacks by initially attacking our black box classifier without any defense. After creating the adversarial attacks we then transfer these images to the ensemble of adversarial defenses for a maximum of \( M \) iterations as shown in Fig. 1 and compute the accuracy. From Table 16 we can see that without our defense the defense the lowest possible accuracy is 16.84%, 11.35%, and 15.57% on the GTSRB, MS-Celeb and CIFAR-10 datasets, respectively. With our defense we can significantly improve the performance to at least 74.60%, 55.72%, and 56.73% accuracy on the GTSRB, MS-Celeb and CIFAR-10 datasets, respectively which is an improvement of at least 3.5x.

4.7 Robustness of Defense - an Adversary’s Point of View

In Section 3.1, we assumed that the adversary has no knowledge about our defense and can only see the input and final classification output of the black box classifier. Although this may seem to be a strong assumption, in this subsection we relax this assumption by allowing the adversary to have partial amounts of information about our defense. We evaluated the CIFAR-10 dataset against the IFGSM and PGD attack with \( \epsilon = 0.1 \) in order to compare our defense with MagNet [26], PixelDefend [27] and ShieldNets [28]. We quantify the robustness of our approach by computing the time taken for the adversary to create 50 successful adversarial attacks against our defense using 2 TITAN X GPUs. Table 17 shows the computational time required to break our defense when the adversary has varying amounts of information about our defense framework.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>GTSRB</td>
<td>22.77/63.07</td>
<td>22.02/80.18</td>
<td>16.84/74.60</td>
<td>23.77/85.04</td>
<td>17.47/78.92</td>
<td>23.80/80.29</td>
</tr>
<tr>
<td>MS-Celeb</td>
<td>17.08/62.91</td>
<td>16.54/60.29</td>
<td>11.35/55.72</td>
<td>19.18/59.67</td>
<td>15.11/58.09</td>
<td>19.35/63.90</td>
</tr>
<tr>
<td>CIFAR-10</td>
<td>20.47/64.80</td>
<td>24.35/66.19</td>
<td>15.57/58.25</td>
<td>21.27/61.06</td>
<td>18.71/56.73</td>
<td>23.89/64.33</td>
</tr>
</tbody>
</table>

Each cell is represented as \( x/y \) where \( x \) and \( y \) are the accuracy before and after applying our ensemble of defenses (i.e., MagNet + ShieldNets + PixelDefend).
TABLE 17: Computation time required for breaking our defense framework on the CIFAR-10 dataset using the IFGSM and PGD attack with $\epsilon = 0.1$.

<table>
<thead>
<tr>
<th>Bayesian CNN</th>
<th>MagNet</th>
<th>Pixel Defend</th>
<th>ShieldNets</th>
<th>Time to create 50 attacks*</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>38/34 hours</td>
</tr>
<tr>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>19/17 hours</td>
</tr>
<tr>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>28/25 hours</td>
</tr>
<tr>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>14/14 hours</td>
</tr>
<tr>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>9.55/6.30 hours</td>
</tr>
<tr>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>8.70/8.25 hours</td>
</tr>
<tr>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>6.65/5.40 hours</td>
</tr>
<tr>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>17/10 minutes</td>
</tr>
<tr>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>34/31 hours</td>
</tr>
<tr>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>$\times$</td>
<td>7.5/6 minutes</td>
</tr>
</tbody>
</table>

* is the part of the defense the adversary has knowledge of, whereas $\bar{x}$ is the part of the defense protected from the adversary.

The table shows that for breaking our defense framework, it takes 38 hours to create 50 successful IFGSM adversarial attacks. To put this into perspective, it took 10 hours for Song et al. [27] to create 100 IFGSM attacks against their defense using 1 TITAN X GPU and 27 hours for Theagarajan et al. [28] to create 50 IFGSM attacks against their defense using 2 TITAN X GPUs.

### 4.7.1 Attacking the Bayesian CNN

In our defense the Bayesian CNN decides if an incoming image is adversarial or not before (i) passing it to the Black box classifier or (ii) rejecting the image. Hence, a natural target for an adversary to beat our defense would be to adversarially attack the Bayesian CNN. The optimization function for creating adversarial examples against the Bayesian CNN is shown in Eq. (18).

$$\arg \max \ |\delta| < \epsilon \quad (18a)$$

$$\text{S.T. } Y^I \neq Y \quad (18b)$$

$$\text{Aleatoric uncertainty } < T_1 \quad (18c)$$

$$\text{Epistemic uncertainty } < T_2 \quad (18d)$$

As shown in Table 17 attacking the Bayesian CNN is by far the weakest point in our defense which can be exploited by an adversary provided the adversary has this information. But, optimizing Eq. (18) makes the adversarial images to be very close to the boundary of the original images in the input space thus resulting in a weakly perturbed adversarial image. We noticed that when we pass these weakly perturbed adversarial images through our ensemble of defense the resulting image is purified in just one iteration of purification. Hence, in Table 17 when we force all input images (regardless if they are adversarial or not) to at least one iteration of purification, these weakly perturbed adversarial examples are no longer adversarial and are correctly classified by the black box classifier. By doing so, even if the adversary has full knowledge about the Bayesian CNN but no information about our ensemble of defenses, it takes approximately 34 hours to break our defense compared to just 17 minutes when there is no forced purification. Moreover, if the adversary has full knowledge of our defense it takes only 6 minutes to create 50 successful PGD attacks, but doing so would violate our assumptions described in Section 3.1 (i.e. security through obscurity). This finding is corroborated by the works done in Ilyas et al. [107], where it is shown that once the adversary has information about the defense (such as model architecture, parameters, gradients, etc.), it is relatively easy to break the defense.

### 4.7.2 Attacking the Ensemble of Iterative Adversarial Defenses

From Table 17 we can see that when the adversary has partial information about our ensemble of defenses, the time taken to break the defense ranges from 28 to 6.65 hours. The reason for this is that with the inclusion of probabilistic generative networks, such as PixelDefend [27] and ShieldNets [28], the ensemble changes from being a deterministic system to a probabilistic system and in order to attack a probabilistic system, one needs to solve the stochastic gradient descent. The convergence rate for solving this is of the order of $O(1/\lambda)$, where $\lambda$ is the convergence error and this is exponentially slower than the deterministic case. Additionally, the individual defenses are all trained independently and have independent parameters, hence a perturbation in a certain direction leading to a misclassification against a particular defense does not necessarily lead to a similar perturbation in the same direction for the other defenses within the ensemble. Moreover, breaking our defense requires a significant amount of probing and querying from the adversary’s side and this can be limited by setting a threshold beyond which the adversary cannot probe the defense for a certain amount of time [107], hence further increasing the computation overhead.

### 5 Conclusions

This paper presented a novel privacy preserving adversarial defense framework for defending black box classifiers. The proposed framework has the ability to convert an existing single-step black box defense into an iterative defense and experimental results showed that using an ensemble of defenses outperforms the state-of-the-art. We demonstrated the relationship between an adversarial image and its corresponding purified image and proved the existence of a lower bound in the input space beyond which an image cannot be further purified. This paper proposed three novel knowledge distillation approaches that exploit prior meta-information of the training datasets, while preserving the privacy of the black box classifiers. Experimental results showed that crowd sourced images that are available in the public domain can be used to effectively distill the knowledge of the Black box classifier and still achieve reasonable performance in defending against adversarial attacks. Most importantly, neither our defense required any information about the parameters of the black box classifiers nor its training data, thus it preserved the privacy and significantly increased its adversarial robustness.

### REFERENCES


