IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

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

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

# Adversarial Defense on Harmony: Reverse Attack for Robust Models Against Adversarial Attacks

**YEBON KIM[1], (MEMBER, IEEE), JINHYO JUNG[2], HYUNJUN KIM[1],(MEMBER, IEEE), HWISOO SO[2], YOHAN KO[4],(MEMBER, IEEE), AVIRAL SHRIVASTAVA[3],(SENIOR MEMBER, IEEE), KYOUNGWOO LEE[2] AND UIWON HWANG[5].**

[1]Deparment of Computer Science, Yonsei University, Wonju, 26493, South Korea (e-mail: kyb0336@yonsei.ac.kr, hyunwns78@yonsei.ac.kr)
[2]Deparment of Computer Science, Yonsei University, Seoul, 03722, South Korea (e-mail: jinhyo.jung@yonsei.ac.kr, kyoungwoo.lee@yonsei.ac.kr,shs7719@yonsei.ac.kr)
[3]School of Computing and Augmented Intelligence, Arizona State University, TEMPE, AZ 85281 USA (e-mail: Aviral.Shrivastava@asu.edu)
[4]Deparment of Software, Yonsei University, Wonju, 26493, South Korea(e-mail: yohan.ko@yonsei.ac.kr)
[5]Division of Digital Healthcare, Yonsei University, Wonju, 26493, South Korea(e-mail: uiwon.hwang@yonsei.ac.kr)

Corresponding author: Yohan Ko (e-mail: yohan.ko@ yonsei.ac.kr).

'This work was supported in part by the U.S. Department of Commerce under Grant BS123456."

**ABSTRACT** Deep neural networks (DNNs) are crucial in safety-critical applications but vulnerable to adversarial attacks, where subtle perturbations cause misclassification. Existing defense mechanisms struggle with small perturbations and face accuracy-robustness trade-offs. This study introduces the "Reverse Attack" method to address these challenges. Our approach uniquely reconstructs and classifies images by applying perturbations opposite to the attack direction, using a complementary "Revenant" classifier to maintain original image accuracy. The proposed method significantly outperforms existing strategies, maintaining clean image accuracy with only a 2.92% decrease while achieving over 70% robust accuracy against all benchmarked adversarial attacks. This contrasts with current mechanisms, which typically suffer an 18% reduction in clean image accuracy and only 36% robustness against adversarial examples. We evaluate our method on the CIFAR-10 dataset using ResNet50, testing against various attacks including PGD and components of Auto Attack. Despite our approach incurs additional computational costs during reconstruction, our method represents a significant advancement in robust defenses against adversarial attacks while preserving clean input performance. This balanced approach paves the way for more reliable DNNs in critical applications. Future work will focus on optimization and exploring applicability to larger datasets and complex architectures.

**INDEX TERMS** Deep neural networks, adversarial attacks and defenses, security, and reliability

## I. INTRODUCTION

Deep neural networks (DNNs) are increasingly being utilized in various fields and have become particularly crucial in safety-critical applications such as malware detection [1], medical imaging [2], and autonomous driving [3]. Considering these applications directly impact human safety, even minor inaccuracies in deep learning models could lead to catastrophic outcomes. Despite their capabilities, deep learning models are often vulnerable to external attacks, primarily because these models depend heavily on the data on which they are trained. Moreover, unlike traditional programming, understanding how a complex deep learning model makes its decisions can be challenging. This opacity complicates the identification of vulnerabilities and the understanding of errors. Consequently, ensuring the robustness of the inference model and mitigating errors from external influences to maintain integrity are increasingly critical concerns.

As one of the deadliest techniques threatening the robustness of deep learning models, an adversarial attack involves injecting carefully crafted perturbations into input images before the inference phase, which decides the model's output, to confuse the model. These perturbations are imperceptible to the human eye but can be designed to cause the target model to misclassify the image with varying degrees of confidence, leading to catastrophic results. Several adversarial attacks are proposed based on having detailed information available and

access to the target model, including hyperparameters and gradients. With this information, attackers can manipulate the input images to induce misclassification using their in-depth model knowledge. The attacks that use model information are named white-box attacks. Examples of white-box attacks include Fast Gradient Sign Method (FGSM) [4], Projected Gradient Descent (PGD) [5] and various Auto Attack methods [6]. These attacks leverage the model's internal information to attack effectively deceive the target model. While highly potent, their reliance on detailed model knowledge can limit their applicability in real-world scenarios where such access may be restricted.

For these limitations, researchers have proposed effective attacks even without access to the model's internal information. These attacks are named black-box attacks. Such attacks involve the strategic insertion of numerous carefully designed inputs into the system to obtain a range of responses from the model. By closely analyzing the variations in model response to these inputs, attackers can gradually accumulate the information required for manipulating the model behavior. While generally less powerful than white-box attacks, advances in black-box methods have led to the design of many attacks that can degrade model performance as much as white-box in scenarios where access to the model internals is restricted. Although these attacks require this additional preliminary step of data gathering, they have achieved performance effectiveness similar to that of attacks that exploit and access all detailed information [7]–[9].

The recent surge in adversarial attacks has spurred significant research on robust defense mechanisms. However, existing techniques, such as adversarial training [5], encounter notable limitations. Although adversarial training improves robustness by incorporating adversarial examples subjected to adversarial attacks into the training alongside clean data, it often suffers from a fundamental tradeoff as the model learns to resist adversarial perturbations, its accuracy with clean data deteriorates. Mitigating this decrease in accuracy requires training with larger or more complex neural networks. However, such models are often impractical for deployment in resource-constrained devices, limiting the applicability of adversarial training in on-device or embedded AI settings.

Several studies proposed to prevent adversarial attacks from affecting AI models, especially in neural network model analysis. Adversarial training methods usually use gradient analysis and decision boundary analysis. Not only for the adversarial training but there are also mitigating methods using these analyses. One of the methods is adjusting regularization techniques in neural networks such as applying Jacobian regularization [10] and additional regularization methods for specific models [11]. Furthermore, some studies apply model ensemble techniques [11], [12]. These ensemble methods use several models that have diverse hyperparameters. By applying these methods, improve the robustness of prediction techniques using different models.

However, these methods require extensive training, including on original and perturbed images including the tradeoff of clean images and adversarial attacked images, denoising adversarial perturbation techniques have been proposed. These techniques exploit smoothing techniques (e.g., low-pass filtering) [13] and deep generative models to project perturbed images back onto a learned image manifold designed to eliminate adversarial noise [14]–[17]. Although these denoising schemes perform well for intense noise, they do not offer sufficient protection against small-scale noise attacks that resemble real-world attacks. In addition, these defense strategies can introduce significant computational overhead during inference [14]–[17] because they denoise all inputs regardless of the presence of attacks.

This paper proposes Reverse Attack, a novel mechanism that performs opposite noising to mitigate different types of adversarial attacks. Reverse attack leverages a fast gradient sign method (FGSM) attack to add noise in the direction opposite to that of the adversarial examples to create similarity in the adversarial example. The proposed method functions with the following three key attributes:

1) **Training a complementary classifier "Revenant"**: We co-train a complementary classifier "Revenant" alongside the clean one. This additional classifier leverages adversarial training with FGSM to add noise to the input images. Noise was added twice: once in the direction of the original FGSM and again in the opposite direction. This process aims to improve the generalization and robustness of Reverse Attack adversarial examples.
2) **Adaptive classification strategy**: We classified adversarial examples using the existing adversarial attack detection method, ML-LOO [18]. If no attack is detected, the image is classified using the original classifier, ensuring that the standard accuracy remains unaffected by the proposed method.
3) **Reverse Attack to reconstruct images**: If an attack is suspected, we perform a "Reverse Attack" using FGSM with the opposite noise direction for each class. The resulting altered image is then classified using the Revenant classifier established in the first stage.

We conducted a comprehensive set of experiments on image classification models to evaluate the proposed method. We specifically focused on ResNet50 [19] trained using the CIFAR-10 dataset [20], which consists of images that resemble everyday objects. We evaluated the robustness of our method against five well-established attacks, the white-box attacks: Projected Gradient Descent (PGD) [5] and four representative attacks auto attacks [6], including APGD-CE, APGD-T, FAB-T, and the black-box attack: Square.

Our findings demonstrate that the proposed approach consistently restores the model accuracy to over 70% on average under all adversarial attacks, contrasting starkly with existing defense mechanisms, which only achieve an average accuracy of approximately 36% under the same conditions. Notably, unlike adversarial training, the proposed approach maintains the accuracy of clean images, resulting in a reduc-

tion of 2.92% in the misclassification rate with the ML-LOO detector. By contrast, adversarial training schemes typically suffer from an 18% decrease in the accuracy of clean images.

## II. BACKGROUND AND RELATED WORKS

For brevity, we discuss some representative adversarial attack methods in Section II-A. Section II-B presents the attacks and defense mechanisms designed to counter adversarial attacks. In Section II-C, we describe the leave-one-out (ML-LOO) [18] detection method, a leave-one-out based approach for adversarial attack detection.

### A. ADVERSARIAL ATTACKS

**FGSM** [4], a cornerstone adversarial attack, manipulates input data during the inference stage. The FGSM operates under the assumption that the attacker possesses in-depth knowledge of the deep learning model, including its gradient and loss functions. By leveraging this access, the FGSM computes the gradients for each pixel of the target input image and then applies a one-step perturbation, adding a minute amount of carefully crafted noise in a direction that maximizes the loss function. This manipulation ultimately deceives the model by misclassifying images.

Equation ( 1) describes the input perturbation in the FGSM scheme. In Equation ( 1), $x$ is the clean input image, and $\tilde{x}$ is the victim image after the perturbation $\delta$. In Equation ( 2), $\bigtriangledown_x L(\theta, x, y)$ represents the loss function of the target model, and $\alpha$ is the sign of the loss function. $\epsilon$ is the epsilon value, a constant set by an attacker to adjust the noise ratio. Based on this idea, adversarial attacks always insert noise to maximize the loss of the target model.

$$\tilde{x} = x + \delta \tag{1}$$

$$\delta = \epsilon \cdot \alpha(\nabla_x L(\theta, x, y)) \tag{2}$$

Building on the core concept of the FGSM, the **Projected Gradient Descent** (PGD) [5] attack is one of the most potent iterative methods for generating adversarial examples. The PGD iteratively modifies the input image by adding small perturbations, calculated using the loss function and gradient of the model. Crucially, PGD leverages the gradient of the modified image in each iteration, thereby making the attack progressively stronger. However, to ensure that the perturbations remain within a bounded range (often referred to as the L-infinity norm), PGD performs orthogonal projection for any value exceeding this limit to return them back to the valid range. This iterative refinement and clipping process enables the PGD to achieve misclassification with surprisingly low noise levels.

With the increasing sophistication of adversarial attack methods, new attacks, such as Auto Attacks, have been developed to evaluate adversarial defense mechanisms. The **Auto Projected Gradient Descent on the Cross-Entropy loss** (APGD-CE) [6] represents a variant of the PGD attack that utilizes cross-entropy loss and automatically adjusts the step size instead of, employing a fixed step size. This attack dynamically alters the size of the steps during execution. Introducing randomness to the initial value, aiming to perform as many iterations as possible within a given epsilon boundary rather than adhering to a set number of iterations. This approach generates effective adversarial examples by repeatedly executing attacks.

Several studies have addressed the challenge of adversarial attacks on image classification models. Notably, **Auto Projected Gradient Descent with Target class** (APGD-T) [6] employs a targeted attack strategy that leverages the difference in the logit ratio as its loss function. This approach allows attackers to specify a target class, thereby enabling the creation of adversarial examples that are deliberately misclassified into specific classes. Another targeted attack, the **Fast Adaptive Boundary Attack with Target class** (FAB-T) [6], [21], utilizes information regarding model gradients and decision boundaries to identify the most influential pixels for manipulating the classification towards a chosen target class. Through an iterative process, FAB-T refines the perturbation to generate adversarial examples close to the decision boundary.

Deep learning models are increasingly deployed in security-sensitive applications, raising concerns about their vulnerability to adversarial attacks. However, the effectiveness of many traditional attacks depends on possessing intricate knowledge of the target model, such as its gradients and loss functions. This dependency on model access presents a critical limitation in real-world scenarios, particularly as advances in model privacy and access control have become generalized. The field of adversarial protection has seen significant progress in developing attacks that can manipulate model predictions without requiring detailed internal information. These "black-box" attacks pose a significant threat, as they can potentially target deployed models in real-world settings. A **Square Attack** [9], a prominent black-box adversarial attack, initializes an adversarial example with a pattern of vertical stripes. During an attack, small square regions within the image are iteratively selected and the pixel values within these regions are adjusted to manipulate model prediction. These modifications are strategically chosen to be near the decision boundaries, thereby minimizing the number of changes required. This approach allows a Square Attack to balance runtime efficiency with attack effectiveness, effectively disrupting the model performance without requiring detailed knowledge of its internal workings. Although less potent than white-box attacks (those with full model access), the Square Attack demonstrates the capability of black-box methods to bypass access restrictions and degrade model performance in real-world scenarios.

### B. DEFENSE AGAINST ADVERSARIAL ATTACK

Since the emergence of adversarial attacks, myriad defense and recovery mechanisms have been introduced. Notably, adversarial training has been recognized as a foundational technique for counteracting such attacks during the training phase of models. This method integrates standard images

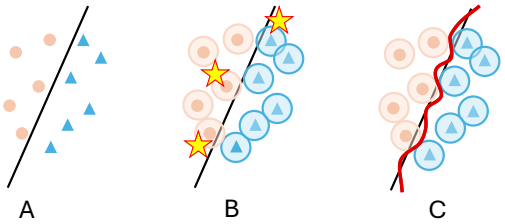



FIGURE 1: Adjusting the decision boundary via adversarial training. (A) The baseline configuration of the target model, where circles and triangles represent distinct labels accurately classified within their respective decision boundaries. (B) Adversarial examples concerning the target model demonstrate how stars located close to the initial decision boundary can be misclassified from circles to triangles or vice versa; shaded circles indicate the variance of the image within the same noise. (C) Effect of adversarial training on the target model, demonstrating that stars from (B) are now correctly classified owing to the formation of a new decision boundary, depicted as a bold line.

with a specialized loss function specifically crafted to neutralize the effects of adversarial examples [5], [22]–[24]. This procedure intentionally generates adversarial examples and evaluates the divergence between these manipulated instances and their original forms, aiming to formulate a decision boundary that effectively reduces the discrepancy between model predictions and authentic images [22], as depicted in Figure 1.

Figure 1 (A) illustrates the model under normal conditions, where the circles and triangles denote distinct categories, neatly separated by the decision boundary to ensure precise classification. Figure 1 (B) depicts an adversarial example designed explicitly for the model. The shaded circles represent the variance in the images caused by identical noise levels. Here, stars (meant to be classified as circles) are positioned close to the decision boundary. This proximity increases the risk of misclassification as triangles or vice versa, especially because the shaded circles cross the original decision boundary. Figure 1 (C) illustrates the effectiveness of adversarial training in the model, incorporating previously misclassified stars from Figure 1 (B), now correctly identified as circles or triangles, including the shaded circles. This process results in an adjustment of the decision boundary of the model, highlighted in bold. This recalibration sharpens the boundary, significantly improving model accuracy in classifying stars as belonging to the circle or triangle categories.

Traditionally, defense methods have focused on correctly classified examples for loss calculations. However, a novel and more advanced approach is emerging, fortifying the decision boundary by incorporating misclassified images (e.g., Method Against Robust Training [23]). This innovative technique enhances the capacity of the model to distinguish between legitimate and adversarial data, marking a significant leap in defense strategies. Recent advances have incorporated diffusion models [24]–[27] to enhance defense capabilities. These models progressively transform images and add noise during learning to remove and retain clean data. This gradual transformation and traditional training techniques promote diverse and practical learning of the target model.

Leading edge methods use diffusion model architectures to generate adversarial images more efficiently with improved algorithms to add noise processes [24]. They also delve into conditional perturbations tailored for specific classes or attributes, further enhancing the models' performance. These techniques promise to correct labels for adversarial examples by strategically manipulating the decision boundaries, as shown in Figure 1. However, a fundamental limitation of almost all these methods is the use of a single classifier for clean and adversarial examples. This mixed classifier can lead to a tradeoff: adversarial examples become more accurately classified at the expense of the accuracy of the clean image. Furthermore, these methods may not provide comprehensive protection against a wide range of attacks with varying epsilon values (maximum allowed perturbation).

Denoising techniques aim to defend against adversarial attacks by removing adversarial noise from the input images. Simple methods like low-pass [13] and Gaussian [28], [29] filtering can be effective for high-frequency noise levels but struggle with other perturbations. More advanced methods [17] add additional noise to cancel out adversarial noise and then train a network to remove it. These methods then train a separate network to remove the combined noise, essentially denoising adversarial perturbation. Although these methods are more effective than simple filtering, they are computationally expensive.

In response to these challenges, ongoing research is dedicated to advancing defense mechanisms against adversarial attacks to overcome these drawbacks, aiming to devise strategies capable of addressing a broad spectrum of attack types and epsilon values, including subtle modifications, while preserving the classification accuracy for genuine images.

### C. ML-LOO FOR ADVERSARIAL ATTACK DETECTION

Adversarial attacks are perturbative noise invisible to the human eye to disrupt the model, so effective detection methods are crucial. However, progress in this area of research is hampered by constantly evolving attacks [30]. Indeed, many detection methods only have experimented with a few gradient-based attacks including state-of-the-art detection methods [31]–[33]. In some cases, advances in attacks have even rendered past methods invalid [34]. The notable exception is ML-LOO [18], which stands out as the most effective detection method against various attacks. ML-LOO detects adversarial attacks by checking the dispersion of feature attributions used by the neural network to determine the outcome. Thus, this study utilizes the ML-LOO [18] approach for detecting adversarial examples before image reconstruction for label correction.

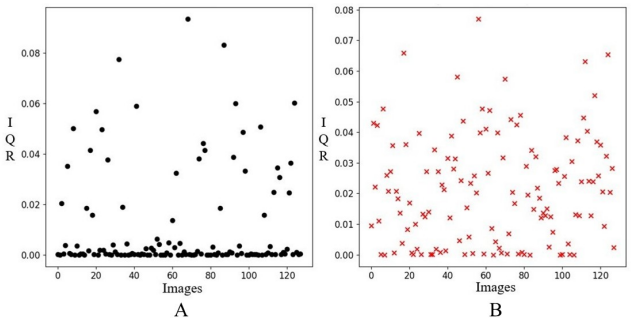



FIGURE 2: Differences in feature map attribution between clean images (A) and adversarial examples (B). X-axis denotes the image index, while Y-axis represents the interquartile range (IQR) value of feature attribution for each image, illustrating the variance in utilized feature numbers during the classification of a single image.

Capitalizing on this observation, ML-LOO extends the method to encompass multilayer feature attribution, enabling the computation of attribution scores for middle layers without requiring additional model queries to identify adversarial examples.

ML-LOO, which leverages the Leave-One-Out (LOO) approach [35] in conjunction with feature attribution to identify adversarial examples, analyzes the impact of removing individual pixels from the input image on the model output, both before and after adversarial perturbation. Although the visual changes caused by the perturbation may be imperceptible, the corresponding feature attribution undergoes significant variations. This difference in feature attribution allows the ML-LOO to distinguish between clean and adversarial examples. [18] found that the interquartile range (IQR) of this feature attribution variation served as a particularly effective metric for detection. Finally, these feature discrepancies are used to train regression models, such as XGBoost [36] or Support Vector Machines (SVM) [37] to detect attacks.

Figure 2 illustrates the feature map differences between clean images (A) and adversarial examples (B). The X-axis, representing the image index, indicates the analysis of a set of images. The Y-axis representing the interquartile range (IQR) of feature attribution for each image shows the variation in the utilized features during the classification process for each image (potentially comparing original vs. adversarial examples). High IQR values on the Y-axis may indicate images in which removing individual pixels significantly affects feature attributions, potentially suggesting the presence of adversarial manipulation.

## III. OUR APPROACH: REVERSE ATTACK

This paper introduces a novel defense mechanism termed "Reverse Attack" to counteract adversarial threats, as illustrated in Figure 3. Our proposed method consists of three main steps: (i) We detect adversarial examples using the ML-LOO method, which utilizes information from the neural network. (ii) If the detector determines that the image is normal, we input the image into the original classifier. (iii) If an attack is detected, we reconstruct the image using our Reverse Attack method and then reclassify it using a new classifier called Revenant.

The proposed approach involves training two distinct classifiers: the original classifier, tasked with handling clean images, and the Revenant classifier, specifically designed to address reconstructed adversarial examples. If the detector indicates that the image is clean, it is classified by the original classifier, thus preserving the accuracy of the non-adversarial inputs. Conversely, upon detecting an attack, the Revenant classifier reconstructs the image in the opposite direction of the attack, allowing the adversarial example to be correctly classified. Notably, the Revenant classifier is tailored to reconstructed adversarial examples, whereas the original classifier is based on any classifier suitable for clean images.

### A. REVERSE ATTACK NEW CLASSIFIER REVENANT

A novel Reverse Attack technique was used to foster accurate label identification under diverse adversarial attacks. Notably, a Reverse Attack promotes consistent similarity between adversarial examples generated by various attack methods for the same original image. The mathematical formulation of the Reverse Attack, detailed in Equation ( 3), closely resembles the FGSM attack [4]. Therefore, we introduce key implementation differences that distinguish our Reverse Attack from FGSM.

$$-\alpha\epsilon\left(\bigtriangledown_x L(\theta, x, y)\right) \tag{3}$$

The first difference is $\alpha$, which denotes the sign or direction of the attack. In the FGSM method, the parameter is adjusted to increase the model loss. By contrast, a Reverse Attack has the opposite sign, meaning that the attack actually decreases the loss of the model. Regardless of the type of attack, the basic concept of adversarial attacks involves perturbations in the direction that increases loss. Therefore, the proposed methodology can be used to generate a particular similarity through a reverse FGSM attack in a direction that reduces model loss, ensuring it can be classified. The second factor is the image gradient, represented by $x$. During inference, the ground-truth-label for an adversarial example is typically unknown. Consequently, our Reverse Attack employs gradients for all possible class labels. This approach generated a set of additional images, one for each potential class. In contrast, FGSM leverages only the gradient associated with the target class, crafting an adversarial example specifically designed to be misclassified as that target class.

Finally, the variable $\epsilon$, a hyperparameter that describes the noise ratio or the intensity of an adversarial attack, should be considered. Through empirical evidence, we found 0.03 to be an appropriate value for creating similarities between data. We leverage the Reverse Attack for training the Revenant classifier and image reconstruction during the inference. To improve the adaptability to typical adversarial attacks on

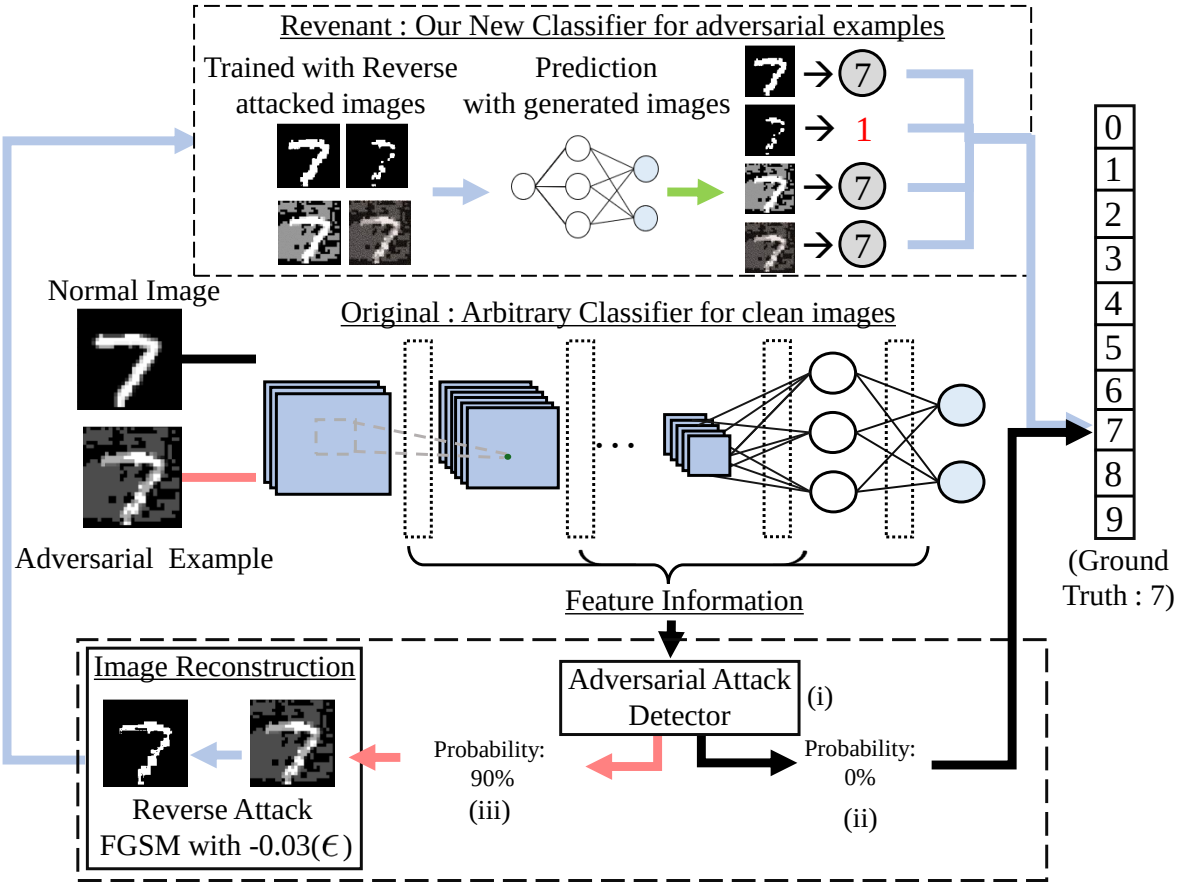



FIGURE 3: Overall workflow of the Reverse Attack method. (i) The attack detector analyzes the features extracted from the original classifier to determine the presence of adversarial attacks. (ii) If no attack is detected, the image is classified directly by the original classifier. (iii) Upon attack detection, our method employs the Reverse Attack reconstruction process followed by reclassification using the Revenant classifier.

**Algorithm 1** Algorithm for generating training data for the Revenant classifier using the Reverse Attack scheme

**Require:** Training data: $D$; Size of dataset: $N$; Number of classes : $C$; Gradient of ground-truth-label: $G$; Gradient of label $C$: $G_C$; $\epsilon$: Epsilon value 0.03; Loss function: $\epsilon\left(\nabla_x L(\theta, G_C, y)\right)$;

$T \leftarrow \{\}$
**for** $i = 1$ to $N$ **do**
  $D' \leftarrow D_i + \epsilon\left(\nabla_x L(\theta, G, y)\right)$
  **for** $j = 1$ to $C$ **do**
    $R \leftarrow D' - \epsilon\left(\nabla_x L(\theta, G_C, y)\right)$
    $T \leftarrow T \cup \{R\}$
  **end for**
**end for**
**return** $T$

**Algorithm 2** Algorithm for the inference of a single image with our Reverse Attack

**Require:** Test data $D$; Number of classes: $C$; Gradient of ground-truth-label: $G$; Gradient of label $C$: $G_C$; $\epsilon$: Epsilon value 0.03; The loss function with gradient $G_C$ is $\epsilon\left(\nabla_x L(\theta, G_C, y)\right)$; Original classifier $O$; Revenant classifier $R$.

**if** $D$ is adversarial example **then**
  $S \leftarrow \{\}$
  **for** $i := 1$ to $C$ **do**
    $D' \leftarrow D - \epsilon\left(\nabla_x L\left(\theta, G_C, y\right)\right)$
    $S \leftarrow S \cup \{R(D')\}$
  **end for**
  **return** vote($S$)
**else**
  **return** $O(D)$
**end if**
End Prediction

images during training, we applied a standard FGSM attack followed by a Reverse Attack. During inference, we perform a Reverse Attack on an attacked image and input it to our Revenant classifier. This combined approach enables the Revenant classifier to handle a broader range of epsilon values and enhances its adaptability to diverse adversarial attacks.

Algorithm 1 describes the process of generating perturbated images to train our Revenant classifier. First, an FGSM attack is applied to images in the original training data to generate adversarial examples (Line 3). Then, each of the generated images undergoes Reverse Attacks, one attack for each possible target class (Line 5). The resulting images are collected to be used as the training set (Line 9). By training solely on this newly generated dataset, the Revenant classifier achieves successful classification of reverse-attacked adversarial examples, returning them to their original labels.

### B. ROBUST CLASSIFICATION

Algorithm 2 outlines the inference phase of the reverse-attack method. Initially, the detector determines whether an input image is an adversarial example (Line 1). If classified as clean, the original classifier assigns a final class label (Line 9). However, if adversarial attacks are detected, the proposed method employs a reconstruction step that uses gradients for all possible class labels (line 4). This Reverse Attack generates additional images, one for each potential class—subsequently fed into a dedicated Revenant classifier (Line 5). The Revenant classifier produces a set of predictions for each reverse-attacked image. Finally, a voting mechanism was applied to the predicted labels to determine the final classification (Line 7). This crucial step enhances the adversarial robustness of the model without compromising the traditional trade-off between clean image accuracy and perturbation robustness. This innovation represents a significant leap forward in maintaining a high-classification Algorithm 2 for inferring a single image with our Reverse Attack accuracy while ensuring a robust defense against adversarial manipulations.

## IV. EXPERIMENTAL SETUP

To evaluate the proposed method comprehensively, we conducted a series of experiments encompassing five adversarial attack methods, 12 models, and four attack scenarios. The twelve models comprised: one baseline model without protections, two models trained using the TRADES method with epsilon values 0.02 and 0.03[1], and three ResNet50 models trained with our Reverse Attack. The accuracy of each model was evaluated under four attack scenarios: no attack (clean images) and three attacks with varying epsilon strengths (0.01, 0.02, and 0.03), resulting in 240 experiments (five attack methods $\times$ 12 models $\times$ four attack scenarios). The details of the experimental setup are as follows:

**Dataset:** To achieve generalizability of the experimental setup, we utilized the CIFAR-10 [20] dataset, which includes images with complex pixel structures and is usually used to demonstrate real-world applicability. The main reason for selecting CIFAR-10 for our experiment is that it is one of the most widely used datasets for adversarial training [38]–[43].

**Neural network model:** We utilized ResNet-50 [19] since it is the most widely recognized benchmark architecture for evaluating protective strategies against adversarial threats. In addition, ResNet-50 has recently become one of the most common and versatile models applicable in many areas. These include medical imaging [2], autonomous vehicle object detection [3], agricultural crop disease detection [44], facial recognition systems [45], and environmental monitoring [46]. Finally, ResNet-50 has a relatively lightweight structure compared to other adversarial training models, allowing for use in various hardware environments, including edge devices and resource-constrained settings. The lightweight nature of ResNet-50 aligns with our goal of developing a method that is both effective and computationally efficient.

**Details for the model training:** We attempted to control for extraneous variables to ensure a fair comparison across the different methods. Considering attacks and defense mechanisms are initially developed in various frameworks, we standardized our experiments using TensorFlow [47] as the primary environment and implemented the methods in-house as necessary. For each technique, the models were trained with a consistent batch size of 128 and a learning rate of 0.001 with 20 epochs, which are empirically chosen based on the performance of the model. The cross-entropy function [48] was employed as the loss function, since it can effectively measure the difference between the predicted probability distribution of the model and the ground-truth label and can improve model performance by sophisticated calculation of the loss, especially in classification problems. Adam [49] algorithm was used as the optimization method since it helps stabilize training and responds effectively to different optimization situations with adaptive learning rate adjustments, providing computational efficiency and ease of hyperparameter tuning.

**Protection techniques:** To establish baseline protection techniques, we implemented notable adversarial training methods, including TRADES [22], MART [23], and the DMAT technique [50], with each employed a distinct ResNet50 network for evaluation. Unlike traditional approaches, the proposed method leverages an independent network trained specifically on adversarial examples. We trained the ResNet50 model to serve as our complementary classifier, Revenant, to ensure a balanced comparison with the existing methods.

Our approach leverages a (ML-LOO) detector to identify the presence of adversarial attacks. However, the overall performance of the proposed method depends partially on the

[1]TRADES exhibited low performance with an epsilon value of 0.01; thus, only the results for the two models trained with epsilon values of 0.02 and 0.03 are presented.

detection accuracy of the ML-LOO. Preliminary evaluations demonstrated that the ML-LOO achieved a detection rate exceeding 98% across various attack types and noise levels. Consequently, employing a more sophisticated detection scheme can marginally improve the accuracy of the proposed method for perturbed images. In addition, the false-positive rate in ML-LOO was excluded from the cleanliness accuracy. The average false-positive rate of the ML-LOO was 2.84%. Considering the proposed method is based on the ML-LOO method, it can be combined with other detectors. Thus, we can develop a detection method that increases the detection rate in real-world situations and further improves the performance.

**Adversarial attacks:** The baseline network and protective strategies are rigorously tested against the PGD attack and the comprehensive Auto Attack suite, which encompasses four distinct components. Specifically, the PGD attack is executed over 20 iterations with a granularity of 0.01 for the step size, meticulously crafted to challenge the resilience of our defense mechanisms. Furthermore, to ensure a robust evaluation of adversarial training effectiveness, we utilized four key components of the auto-attack suite: APGD-CE (Adaptive Projected Gradient Descent targeted at Cross-Entropy), APGD-T (Adaptive Projected Gradient Descent with a Targeted approach), FAB-T (Fast Adaptive Boundary attack for Targeted aggression), and Square Attack. These attack scenarios demonstrate the vulnerability of unprotected networks and the fault coverage of protection schemes in simulating a wide array of adversarial attack scenarios, thus providing a comprehensive assessment of the protective measures in place.

**Choosing an epsilon value for attacks:** Determining an optimal epsilon value for adversarial attacks is crucial in this analysis because it directly influences the intensity of the perturbations applied to images within the datasets. Consequently, 0.03, the most used value in the comprehensive literature [51]–[53] on defense mechanisms against adversarial threats, was selected as the optimal value. Also, the rationale for this value of 0.03 is that it is the most powerful an attack can be without being noticed by humans [54]. Furthermore, we systematically trained each model to counter each type of attack using a series of epsilon values, 0.01, 0.02, and 0.03, to comprehensively assess the model resilience across varying levels of adversarial perturbations.

**Hardware environment:** The experiments were conducted on a robust hardware setup featuring an Intel(R) Core(TM) i9-10980XE CPU @3.00GHz, complemented by four NVIDIA GeForce RTX 3090 GPUs and a single NVIDIA A100 80GB SXM GPU.

## V. EXPERIMENTAL RESULTS

### A. EFFICACY AGAINST ADVERSARIAL ATTACKS

Table Table 1 shows all conducted experimental results for the unprotected and protected models. Each sub-table shows the results against different adversarial attacks. Each protected models were trained with training epsilon from 0.01 to 0.03, except for TRADES which shows low performance with training epsilon 0.01. In each table, the clean accuracy column represents the classification accuracy of each model for clean images. On the other hand, robust accuracy means the accuracy of each model for the images attacked by the adversarial attack with different epsilon values, i.e., the magnitude of the attack. The last column shows the average robust accuracy across three epsilon values

**PGD attack:** Table 1(a) summarizes the performances of various methods under a PGD attack. For the unprotected ResNet50 model, the PGD attack significantly affected the classification accuracy – from 78.07% (clean accuracy) to 10.22%. While TRADES offers some protection (average robust accuracy: 35.38%), it results in a 9.25% reduction in cleanliness accuracy. MART achieved a slightly higher average robust accuracy (39.03%) but suffered a more substantial drop in cleanliness accuracy (11.99%). DMAT exhibited the most significant decline in clean accuracy (18.10%) while providing an average robust accuracy (35.48%) comparable to TRADES. The proposed Reverse Attack method addresses this trade-off by leveraging separate classifiers, resulting in almost no reduction in clean accuracy (75.48%) while achieving a superior average robust accuracy of 66.09% and 75.48% for the adversarial examples.

**APGD-CE attack:** Next, we evaluated the methods against an APGD-CE attack, a component of an Auto Attack, shown in Table 1(b). The data presented in this table indicates that attacks are evolving and becoming more sophisticated, rendering the classification of adversarial examples increasingly challenging. The average robust accuracy for all epsilons and models for the TRADES, MART, and DMAT was 34.76%, 40.68%, and 38.32%, respectively. However, the accuracies for clean images were 68.82%, 66.08%, and 59.02%, respectively, highlighting a challenge in adjusting the decision boundaries that are both accurate and robust. Even the highest robust accuracy in the comparison group was 57.93% when the DMAT was trained with an epsilon value of 0.03 and attacked with an intensity of 0.01. However, this accuracy was 13.03% lower than the lowest robust accuracy value of the proposed method (70.96%). Without protection, the average accuracy of ResNet50 decreased from 78.07% to 8.19%.

**APGD-T attack:** The experimental results for the APGD-T attack shown in Table 1(c) show tendencies similar to those for the APGD-CE attack. This attack decreased the accuracy of ResNet50 from 78.07% to 7.94%. TRADES recorded over 50% accuracy for attacks with an epsilon value of 0.01 but exhibited underwhelming performance for more intense attacks, averaging a robust accuracy of only 32.80% and a clean accuracy of 68.82%. With MART, the average robust accuracy was 38.12%, and the average accuracy for clean images was 66.08%. DMAT showed the highest accuracy among the comparison groups: 55.74% against an epsilon 0.01 attack when trained with an epsilon value of 0.01; on average, it achieved 35.49% robust accuracy and 59.02% cleanliness accuracy. The proposed method also reliably

TABLE 1: Comparison of all attack scenarios under different epsilon values and defense strategies. The best performance of each model is underlined.

(a) PGD Attack

| Model | Training epsilon | Clean accuracy | Robust accuracy against attack with epsilon value: 0.01 | 0.02 | 0.03 | Robust accuracy average |
|---|---|---|---|---|---|---|
| No protection | - | 78.07 | 9.94 | 10.33 | 10.39 | 10.22 |
| TRADES [22] | 0.02 | 73.05 | 14.97 | 30.45 | **51.31** | 32.24 |
| | 0.03 | 64.59 | 26.21 | 37.95 | **51.39** | 38.52 |
| | Avg | 68.82 | 20.59 | 34.20 | **51.35** | 35.38 |
| MART [23] | 0.01 | 73.03 | 16.89 | 32.67 | **53.27** | 34.28 |
| | 0.02 | 66.03 | 27.70 | 40.19 | **53.09** | 40.33 |
| | 0.03 | 59.17 | 35.13 | 42.98 | **49.36** | 42.49 |
| | Avg | 66.08 | 26.57 | 38.61 | **51.91** | 39.03 |
| DMAT [55] | 0.01 | 73.86 | 16.10 | 34.30 | **55.74** | 35.38 |
| | 0.02 | 57.25 | 27.61 | 37.46 | **47.37** | 37.48 |
| | 0.03 | 48.79 | 27.79 | 33.35 | **39.61** | 33.58 |
| | Avg | 59.97 | 23.83 | 35.04 | **47.57** | 35.48 |
| Ours | 0.01 | 76.13 | 64.50 | 64.61 | **67.61** | 65.57 |
| | 0.02 | 76.45 | 65.54 | **65.89** | 65.20 | 65.54 |
| | 0.03 | 73.86 | 67.07 | **67.38** | 65.20 | 66.55 |
| | Avg | 75.48 | 65.88 | 66.09 | **66.29** | 66.09 |

(b) APGD-CE Attack

| Model | Training epsilon | Clean accuracy | Robust accuracy against attack with epsilon value: 0.01 | 0.02 | 0.03 | Robust accuracy average |
|---|---|---|---|---|---|---|
| No protection | - | 78.07 | 20.91 | 3.35 | 0.30 | 8.19 |
| TRADES [22] | 0.02 | 73.05 | **52.71** | 16.03 | 16.03 | 28.26 |
| | 0.03 | 64.59 | **53.18** | 40.70 | 29.89 | 41.26 |
| | Avg | 68.82 | **52.95** | 28.37 | 22.96 | 34.76 |
| MART [23] | 0.01 | 59.17 | **51.37** | 42.74 | 18.30 | 37.47 |
| | 0.02 | 66.03 | **55.17** | 43.28 | 31.11 | 43.19 |
| | 0.03 | 73.03 | **54.68** | 34.38 | 35.08 | 41.38 |
| | Avg | 66.08 | **53.74** | 40.13 | 28.16 | 40.68 |
| DMAT [55] | 0.01 | 45.94 | **41.10** | 36.14 | 32.48 | 36.57 |
| | 0.02 | 57.25 | **49.49** | 40.97 | 31.13 | 40.53 |
| | 0.03 | 73.86 | **57.93** | 37.15 | 18.48 | 37.85 |
| | Avg | 59.02 | **49.51** | 38.09 | 27.36 | 38.32 |
| Ours | 0.01 | 75.34 | **71.76** | 71.02 | 70.96 | 71.25 |
| | 0.02 | 75.32 | 71.17 | 71.17 | **71.36** | 71.23 |
| | 0.03 | 76.73 | **71.68** | 71.04 | 71.17 | 71.30 |
| | Avg | 75.80 | **71.53** | 71.08 | 71.16 | 71.26 |

(c) APGD-T Attack

| Model | Training epsilon | Clean accuracy | Robust accuracy against attack with epsilon value: 0.01 | 0.02 | 0.03 | Robust accuracy average |
|---|---|---|---|---|---|---|
| No protection | - | 78.07 | 20.51 | 3.10 | 0.22 | 7.94 |
| TRADES [22] | 0.02 | 73.05 | **51.31** | 14.97 | 14.97 | 27.08 |
| | 0.03 | 64.59 | **51.39** | 37.95 | 26.22 | 38.52 |
| | Avg | 68.82 | **51.35** | 26.46 | 20.60 | 32.80 |
| MART [23] | 0.01 | 73.03 | 32.67 | **53.27** | 16.89 | 34.28 |
| | 0.02 | 66.03 | **53.09** | 40.19 | 27.71 | 40.33 |
| | 0.03 | 59.17 | 39.67 | **49.36** | 30.24 | 39.76 |
| | Avg | 66.08 | 41.81 | **47.61** | 24.95 | 38.12 |
| DMAT [55] | 0.01 | 73.86 | **55.74** | 34.33 | 16.12 | 35.40 |
| | 0.02 | 57.25 | **47.37** | 37.46 | 27.62 | 37.48 |
| | 0.03 | 45.94 | **39.61** | 33.35 | 27.79 | 33.58 |
| | Avg | 59.02 | **47.57** | 35.05 | 23.84 | 35.49 |
| Ours | 0.01 | 75.38 | 70.85 | **70.86** | 71.75 | 71.15 |
| | 0.02 | 76.83 | 71.09 | 70.98 | **71.52** | 71.20 |
| | 0.03 | 75.68 | 70.82 | 70.85 | **71.59** | 71.09 |
| | Avg | 75.96 | 70.92 | 70.90 | **71.62** | 71.15 |

(d) FAB-T Attack

| Model | Training epsilon | Clean accuracy | Robust accuracy against attack with epsilon value: 0.01 | 0.02 | 0.03 | Robust accuracy average |
|---|---|---|---|---|---|---|
| No protection | - | 78.07 | 24.45 | 4.80 | 0.59 | 9.95 |
| TRADES [22] | 0.02 | 73.05 | **51.31** | 14.97 | 14.97 | 27.08 |
| | 0.03 | 64.59 | **51.39** | 37.95 | 26.21 | 38.52 |
| | Avg | 68.82 | **51.35** | 26.46 | 20.59 | 32.80 |
| MART [23] | 0.01 | 73.03 | **53.27** | 32.67 | 16.89 | 34.28 |
| | 0.02 | 66.03 | **53.09** | 40.19 | 27.70 | 40.33 |
| | 0.03 | 59.17 | **49.36** | 39.89 | 30.82 | 40.02 |
| | Avg | 66.08 | **51.91** | 37.58 | 25.14 | 38.21 |
| DMAT [55] | 0.01 | 73.86 | **55.74** | 34.33 | 16.10 | 35.39 |
| | 0.02 | 57.25 | **47.37** | 37.46 | 27.61 | 37.48 |
| | 0.03 | 45.94 | **39.61** | 33.35 | 27.79 | 33.58 |
| | Avg | 59.02 | **47.57** | 35.05 | 23.83 | 35.48 |
| Ours | 0.01 | 75.62 | **71.39** | 71.09 | 71.18 | 71.22 |
| | 0.02 | 75.16 | **71.39** | 71.06 | 71.16 | 71.20 |
| | 0.03 | 76.09 | **71.28** | 71.02 | 71.19 | 71.16 |
| | Avg | 75.62 | **71.35** | 71.06 | 71.18 | 71.20 |

(e) Sqaure Attack

| Model | Training epsilon | Clean accuracy | Robust accuracy against attack with epsilon value: 0.01 | 0.02 | 0.03 | Robust accuracy average |
|---|---|---|---|---|---|---|
| No protection | - | 78.07 | 3.35 | 14.37 | 5.27 | 7.66 |
| TRADES [22] | 0.02 | 73.05 | **51.31** | 30.45 | 14.97 | 32.24 |
| | 0.03 | 64.59 | **51.39** | 37.95 | 26.21 | 38.52 |
| | Avg | 68.82 | **51.35** | 34.20 | 20.59 | 35.38 |
| MART [23] | 0.01 | 73.03 | **53.27** | 32.67 | 16.89 | 34.28 |
| | 0.02 | 66.03 | **53.09** | 40.19 | 27.70 | 40.33 |
| | 0.03 | 59.17 | **49.36** | 39.89 | 30.82 | 40.02 |
| | Avg | 66.08 | **51.91** | 37.58 | 25.14 | 38.21 |
| DMAT [55] | 0.01 | 73.86 | **55.74** | 34.33 | 16.10 | 35.39 |
| | 0.02 | 57.25 | **47.37** | 37.46 | 27.61 | 37.48 |
| | 0.03 | 45.94 | **39.61** | 33.35 | 27.79 | 33.58 |
| | Avg | 59.02 | **47.57** | 35.05 | 23.83 | 35.48 |
| Ours | 0.01 | 76.28 | 71.63 | 71.05 | **71.64** | 71.44 |
| | 0.02 | 76.11 | **71.63** | 71.02 | 71.62 | 71.42 |
| | 0.03 | 75.81 | **71.52** | 70.98 | 71.02 | 71.17 |
| | Avg | 76.07 | 71.00 | 70.40 | **71.64** | 71.01 |

achieved an average robust accuracy of 71.15% and 75.96% for clean images and consistently achieved a classification accuracy above 70% for all epsilon values used in training and attacks.

**FAB-T attack:** The validation of the FAB-T attack is depicted in Table 1(d), following the same tendency as the previous attacks. These attacks decrease the average accuracy of the ResNet model from 78.07% to 9.95%. On average, TRADES achieved 68.82% for clean images, MART achieved 66.08%, and DMAT achieved 59.02%. The robust accuracies of TRADES, MART, and DMAT were 32.80%, 38.21%, and 35.48%, respectively. Against this attack, the highest robust accuracy in the comparison group was 55.74% for the DMATs trained and attacked with 0.01 epsilon. The average robust accuracy of our technique against FAB-T attack was 71.20%, 15.46% higher than the highest robust accuracy of the DMAT.

**Square attack:** Table 1(e) shows the results for a Square attack that does not require the internal information of the model. The ResNet50 model with no protection demonstrated a considerable reduction in accuracy, from 78.07% to 7.66%. For the clean images, TRADES, MART, and DMAT achieved an average accuracy of 68.82%, 66.08%, and 59.02%, respectively. By contrast, TRADES achieved an average robust accuracy of 35.38%, MART of 38.21%, and DMAT of 35.48%. These results indicated that a Square attack is relatively ineffective against adversarial training models owing to its lack of reliance on the information contained within the target model. As with other attacks, our method classified adversarial examples well, maintaining over 70% accuracy for all experiments, with an average robust accuracy of 71.01%. In addition, clean images were classified with an average accuracy of 76.07%.

**Overall:** Our experiments, which evaluated various models under different noise rates and attack methods, yielded several key results. While existing adversarial training methods are effective at classifying adversarial examples, this improvement often comes at the cost of reduced accuracy for clean images. We observed a trend in which a higher epsilon value used during training corresponded to a decrease in clean image classification accuracy. However, the proposed Reverse Attack method does not suffer from this accuracy trade-off. Unlike adversarial training, where the highest clean image accuracy was 73.86% (achieved by DMAT), the proposed method maintained almost the same clean image accuracy as that of the original model (75.79% compared to 78.07% for clean images on the basic model).

In addition, our method consistently achieved the highest robust accuracy across all experimental configurations, confirming the capability of our Revenant classifier. Specifically, our approach achieved an average robust accuracy of 70.14% across all attacks and epsilon values ($\epsilon$ = 0.01, 0.02, and 0.03), significantly outperforming the next best method (MART) which achieved only 38.85% under the same conditions. Breaking this down by attack type, our method demonstrated robust accuracies of 66.09% against PGD, 71.26% against APGD-CE, 71.15% against APGD-T, 71.20% against FAB-T, and 71.01% against Square attacks, with these results being consistent across all tested epsilon values. This represented a substantial improvement over existing methods, with our approach outperforming the next best method by an average of 31.29 percentage points in robust accuracy while maintaining clean image accuracy within 2.28 percentage points of the accuracy on clean images for the unprotected model. Notably, our method's performance remained stable across different epsilon values, unlike other methods which showed declining performance with increasing epsilon. These results underscore the effectiveness of our Reverse Attack method in balancing robust defense against adversarial attacks with the preservation of clean image accuracy, regardless of the perturbation magnitude.

### B. ABLATION STUDY FOR REVERSE ATTACK

TABLE 2: Efficacy of the Reverse Attack, compared to images trained with the FGSM and PGD attacks without the Reverse Attack. ($\epsilon = 0.03$)

| Training method | Clean accuracy | Robust accuracy against | |
|---|---|---|---|
| | | PGD | APGD-CE |
| FGSM (without reverse) | 78.07 | 54.64 | 52.84 |
| PGD (without reverse) | 78.07 | 52.20 | 26.22 |
| Ours (with reverse) | 78.07 | 66.09 | 71.26 |

To assess the impact of Reverse Attack integration during training, we compared classifiers trained on all labels for standard FGSM and PGD attacks (without Reverse Attack) to a Revenant trained with Reverse Attack augmentation, using 0.03 for all epsilon values in Table 2 The models trained without Reverse Attacks displayed suboptimal overall accuracy, where the classifiers trained with FSGM attacks achieved 54.64% and 52.84% accuracies for PGD and APGDCE attacks, respectively, and those trained with PGD attacks achieved 52.20% and 26.22% accuracies, respectively. However, our Revenant classifier trained with Reverse Attacks demonstrates superior performance to the other classifiers, achieving a robust accuracy of 66.09% for PGD and 71.26% for APGD-CE. This significant improvement (approximately 15%) highlights the effectiveness of incorporating Reverse Attacks during training to enhance classification robustness against diverse adversarial attacks.

### C. COMPARISON WITH STATE-OF-THE-ART SOLUTIONS IN VARIOUS ENVIRONMENTS

TABLE 3: Results on CIFAR-10 dataset for different defense strategies and attack methods with $\epsilon = 0.03$ at epoch 50

| Model | Epsilon | Clean accuracy | Robust accuracy against | | | |
|---|---|---|---|---|---|---|
| | | | APGD-CE | APGD-T | FAB-T | Square |
| Unprotected | 0.03 | 80.15 | 4.26 | 2.51 | 2.81 | 10.91 |
| TRADES | 0.03 | 66.32 | **33.72** | 30.30 | 30.29 | 30.29 |
| MART | 0.03 | 65.25 | **36.09** | 31.62 | 31.62 | 31.62 |
| DMAT | 0.03 | 81.47 | **51.92** | 47.75 | 47.74 | 47.74 |

To evaluate the efficacy of our proposed methodology with a low epoch, we additionally evaluated the clean and robust accuracy of state-of-the-art technologies with a higher epoch. Table 3 shows the effect of increasing the number of epochs to 50 on the clean image accuracy for each method. Interestingly, the state-of-the-art DMAT method exhibited a notable improvement in clean image accuracy, reaching 81.47%, surpassing the classification rate of the original model (80.15%), likely owing to the limited baseline accuracy of the original model. Notably, training with additional epochs evidently increased the clean image classification rate while potentially causing a readjustment of the decision boundary. Despite this improvement, the average robust accuracy of the DMAT against attacks with an epsilon value of 0.03 remains at 48.79%, highlighting a crucial limitation: even with extensive training, DMAT falls short of achieving an accuracy comparable to that of our method (over 20% lower).

We also evaluated the robustness of our Reverse Attack against strong adversarial attacks, particularly PGD attacks, with an epsilon value of 0.1. The findings demonstrate that a Reverse Attack with an epsilon value of 0.03 is sufficient to recover from such attacks, achieving a robust accuracy of 63.02%. In other words, even when the attacker utilizes a larger epsilon value (0.1), our Reverse Attack with a smaller epsilon value (0.03) can effectively reconstruct the original image and maintain the classification accuracy.

### D. EFFICACY OF SINGLE-LABEL REVERSE ATTACK

TABLE 4: Classification outcomes following single-label image reconstruction in response to a PGD attack, utilizing an epsilon of 0.03 for both training and the attack phase

| Label | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| Accuracy | 72.0 | 80.2 | 50.1 | 38.8 | 60.3 | 65.8 | **85.0** | 68.6 | 74.2 | 75.9 |

The current image reconstruction process incurs significant computational overhead. To address this issue, we explored the feasibility of using a single label for a Reverse Attack instead of using all possible labels. Table 4 presents the classification results from image reconstruction using a single label under a PGD attack with an epsilon value of 0.03 (consistent with both training and attack). Analyzing image reconstruction and classification across all labels for a PGD attack revealed that specific labels consistently achieved higher accuracy than the majority voting approach. Notably, image reconstruction using Label 6 achieved a robust accuracy of 85.0%. These findings suggested that strategically selecting a single label for a Reverse Attack could potentially reduce computational costs considerably while maintaining high accuracy against various adversarial attacks.

## VI. CONCLUSION

Deep neural networks (DNNs) have become ubiquitous in safety-critical applications, such as malware detection, medical imaging, and autonomous vehicles. However, their vulnerability to adversarial attacks in which meticulously crafted inputs can lead to misclassifications poses a significant threat. This study presented a novel Reverse Attack defense method that leverages the modified FGSM to reconstruct adversarial examples. By proposing the Revenant classifier to reconstruct adversarial examples to be used with the original classifier, both clean and adversarial images can be accurately classified. Unlike previous adversarial training approaches, which sacrifice cleanliness accuracy, the proposed method performs reconstruction and reclassification only when an attack is detected. This selective approach outperforms existing methods, achieving an average classification accuracy improvement exceeding 20% under the same conditions. In addition, we explored potential optimizations to reduce the image reconstruction overhead. Future work will focus on further reducing this overhead and enhancing classification efficiency to mitigate the impact of adversarial attacks in real-world applications.

## REFERENCES

[1] Dragoş Gavriluţ, Mihai Cimpoeşu, Dan Anton, and Liviu Ciortuz. Malware detection using machine learning. In 2009 International Multiconference on Computer Science and Information Technology, pages 735–741, 2009.

[2] Ahmad Waleed Salehi, Shakir Khan, Gaurav Gupta, Bayan Ibrahimm Alabduallah, Abrar Almjally, Hadeel Alsolai, Tamanna Siddiqui, and Adel Mellit. A study of cnn and transfer learning in medical imaging: Advantages, challenges, future scope. Sustainability, 15(7), 2023.

[3] Kalyan Sudhakar and Research Publications. Machine learning - autonomous vehicles. SSRN Electronic Journal, 8:314–333, 07 2018.

[4] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples, 2015.

[5] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks, 2019.

[6] Francesco Croce and Matthias Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In International conference on machine learning, pages 2206–2216. PMLR, 2020.

[7] Chuan Guo, Jacob Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Weinberger. Simple black-box adversarial attacks. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, Proceedings of the 36th International Conference on Machine Learning, volume 97 of Proceedings of Machine Learning Research, pages 2484–2493. PMLR, 09–15 Jun 2019.

[8] Minhao Cheng, Simranjit Singh, Patrick Chen, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. Sign-opt: A query-efficient hard-label adversarial attack, 2020.

[9] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In European conference on computer vision, pages 484–501. Springer, 2020.

[10] Kenneth T Co, David Martinez Rego, and Emil C Lupu. Jacobian regularization for mitigating universal adversarial perturbations. In International Conference on Artificial Neural Networks, pages 202–213. Springer, 2021.

[11] Yian Deng and Tingting Mu. Understanding and improving ensemble adversarial defense. Advances in Neural Information Processing Systems, 36, 2024.

[12] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. arXiv preprint arXiv:1705.07204, 2017.

[13] Vadim Ziyadinov and Maxim Tereshonok. Low-pass image filtering to achieve adversarial robustness. Sensors, 23(22):9032, 2023.

[14] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In Proceedings of the 2017 ACM SIGSAC conference on computer and communications security, pages 135–147, 2017.

[15] Pouya Samangouei, Maya Kabkab, and Rama Chellappa. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In International Conference on Learning Representations, 2018.

[16] Uiwon Hwang, Jaewoo Park, Hyemi Jang, Sungroh Yoon, and Nam Ik Cho. Puvae: A variational autoencoder to purify adversarial examples. IEEE Access, 7:126582–126593, 2019.

[17] Jongmin Yoon, Sung Ju Hwang, and Juho Lee. Adversarial purification with score-based generative models, 2021.

[18] Puyudi Yang, Jianbo Chen, Cho-Jui Hsieh, Jane-Ling Wang, and Michael Jordan. Ml-loo: Detecting adversarial examples with feature attribution. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, pages 6639–6647, 2020.

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

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[21] Francesco Croce and Matthias Hein. Minimally distorted adversarial examples with a fast adaptive boundary attack. In International Conference on Machine Learning, pages 2196–2205. PMLR, 2020.

[22] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In International conference on machine learning, pages 7472–7482. PMLR, 2019.

[23] Yisen Wang, Difan Zou, Jinfeng Yi, James Bailey, Xingjun Ma, and Quanquan Gu. Improving adversarial robustness requires revisiting misclassified examples. In International conference on learning representations, 2019.

[24] Zekai Wang, Tianyu Pang, Chao Du, Min Lin, Weiwei Liu, and Shuicheng Yan. Better diffusion models further improve adversarial training. In International Conference on Machine Learning, pages 36246–36263. PMLR, 2023.

[25] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. arXiv preprint arXiv:2010.02502, 2020.

[26] Weili Nie, Brandon Guo, Yujia Huang, Chaowei Xiao, Arash Vahdat, and Anima Anandkumar. Diffusion models for adversarial purification. arXiv preprint arXiv:2205.07460, 2022.

[27] Jianqi Chen, Hao Chen, Keyan Chen, Yilan Zhang, Zhengxia Zou, and Zhenwei Shi. Diffusion models for imperceptible and transferable adversarial attack. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024.

[28] Zhaoxia Yin, Hua Wang, Jie Wang, Jin Tang, and Wenzhong Wang. Defense against adversarial attacks by low-level image transformations. International Journal of Intelligent Systems, 35(10):1453–1466, 2020.

[29] Aamir Khan, Weidong Jin, Amir Haider, MuhibUr Rahman, and Desheng Wang. Adversarial gaussian denoiser for multiple-level image denoising. Sensors, 21(9):2998, 2021.

[30] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In International conference on machine learning, pages 274–283. PMLR, 2018.

[31] Alvaro Lopez Pellicer, Kittipos Giatgong, Yi Li, Neeraj Suri, and Plamen Angelov. Unicad: A unified approach for attack detection, noise reduction and novel class identification. In 2024 International Joint Conference on Neural Networks (IJCNN), pages 1–8. IEEE, 2024.

[32] Manzoor Hussain and Jang-Eui Hong. Reconstruction-based adversarial attack detection in vision-based autonomous driving systems. Machine Learning and Knowledge Extraction, 5(4):1589–1611, 2023.

[33] Yuanyuan Qing, Tao Bai, Zhuotao Liu, Pierre Moulin, and Bihan Wen. Detection of adversarial attacks via disentangling natural images and perturbations. IEEE Transactions on Information Forensics and Security, 2024.

[34] Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E. Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality, 2018.

[35] Jiwei Li, Will Monroe, and Dan Jurafsky. Understanding neural networks through representation erasure. arXiv preprint arXiv:1612.08220, 2016.

[36] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining, pages 785–794, 2016.

[37] Corinna Cortes and Vladimir Vapnik. Support-vector networks. Machine Learning, 20(3):273–297, 1995.

[38] Yang Bai, Yuyuan Zeng, Yong Jiang, Shu-Tao Xia, Xingjun Ma, and Yisen Wang. Improving adversarial robustness via channel-wise activation suppressing. arXiv preprint arXiv:2103.08307, 2021.

[39] Gaurang Sriramanan, Sravanti Addepalli, Arya Baburaj, et al. Towards efficient and effective adversarial training. Advances in Neural Information Processing Systems, 34:11821–11833, 2021.

[40] Tianlong Chen, Zhenyu Zhang, Sijia Liu, Shiyu Chang, and Zhangyang Wang. Robust overfitting may be mitigated by properly learned smoothening. In International Conference on Learning Representations, 2020.

[41] Shuo Yang, Tianyu Guo, Yunhe Wang, and Chang Xu. Adversarial robustness through disentangled representations. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 35, pages 3145–3153, 2021.

[42] Ling Yang, Haotian Qian, Zhilong Zhang, Jingwei Liu, and Bin Cui. Structure-guided adversarial training of diffusion models. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 7256–7266, 2024.

[43] Xiaojun Jia, Yong Zhang, Xingxing Wei, Baoyuan Wu, Ke Ma, Jue Wang, and Xiaochun Cao. Improving fast adversarial training with prior-guided knowledge. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2024.

[44] Sharada P Mohanty, David P Hughes, and Marcel Salathé. Using deep learning for image-based plant disease detection. Frontiers in plant science, 7:1419, 2016.

[45] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 815–823, 2015.

[46] Mohammad Sadegh Norouzzadeh, Anh Nguyen, Margaret Kosmala, Alexandra Swanson, Meredith S Palmer, Craig Packer, and Jeff Clune. Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. Proceedings of the National Academy of Sciences, 115(25):E5716–E5725, 2018.

[47] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. arXiv preprint arXiv:1603.04467, 2016.

[48] Anqi Mao, Mehryar Mohri, and Yutao Zhong. Cross-entropy loss functions: Theoretical analysis and applications. In International Conference on Machine Learning, pages 23803–23828. PMLR, 2023.

[49] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.

[50] Wei-An Lin, Chun Pong Lau, Alexander Levine, Rama Chellappa, and Soheil Feizi. Dual manifold adversarial robustness: Defense against lp and non-lp adversarial attacks. Advances in Neural Information Processing Systems, 33:3487–3498, 2020.

[51] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. arXiv preprint arXiv:1706.06083, 2017.

[52] Raphael Olivier and Bhiksha Raj. How many perturbations break this model? evaluating robustness beyond adversarial accuracy. In International Conference on Machine Learning, pages 26583–26598. PMLR, 2023.

[53] Pratyush Maini, Eric Wong, and Zico Kolter. Adversarial robustness against the union of multiple perturbation models. In International Conference on Machine Learning, pages 6640–6650. PMLR, 2020.

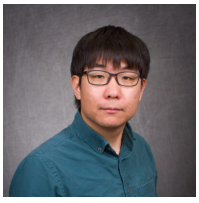
[54] Raz Lapid, Almog Dubin, and Moshe Sipper. Fortify the guardian, not the treasure: Resilient adversarial detectors. arXiv preprint arXiv:2404.12120, 2024.

[55] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. Advances in Neural Information Processing Systems, 35:26565–26577, 2022.

**Yebon Kim** (Member, IEEE) received the B.S. degree in Computer Engineering from Yonsei University, Wonju, Korea, in 2022 and he is currently in the Masters Course in Reliable Artificial Intelligence and Computer System Lab (RAISE Lab) at Yonsei University. His research interests are adversarial attack and defense in AI models with integrity maintenance conditions in deep neural networks.

**Jinhyo Jung** Received the B.S. degree in computer science from Yonsei University, Seoul, Korea, in 2019. He is currently in the integrated Ph.D. Course in Dependable Computing Lab (DClab) at Yonsei University. His research interests include vulnerability estimation of computer architectures and error protection schemes for deep neural networks.

**Hyunjun Kim** (Member, IEEE) received the B.S. degree in Computer Engineering from Yonsei University, Wonju, Korea, in 2024 and he is currently in the Masters Course in Reliable Artificial Intelligence and Computer System Lab (RAISE Lab) at Yonsei University. His research interests focus on adversarial attacks and defenses utilizing generative models, with an emphasis on maintaining integrity in deep learning models specialized for generative AI.



**Hwisoo So** Received a bachelor's degree and PhD degree in computer science from Yonsei University. He is working as post-doc in the same university, and visiting Arizona State University as a visiting research scholar. His main research interest is the reliability against hardware faults such as soft errors, including software-level redundancy solutions to mitigate hardware faults and comprehensive vulnerability estimation of computer architecture.

**Yohan Ko** (Member, IEEE) received a B.S. degree and Ph.D. in Computer Science from the Yonsei University, Seoul, Korea. He is currently an Associate Professor in the Division of Software, at Yonsei University. His research are connected with reliability, especially on Computer Systems and Deep Neural Networks. Even more, the adversarial attack and defense, and OOD(Out-Of-Distribution) in AI models.

**Aviral Shrivastava** (Senior Member, IEEE) received Ph.D. and M.S in Information and Computer Science from the University of California, Irvine, and B.S. in Computer Science and Engineering from the Indian Institute of Technology, Delhi. His research lies in the broad area of "Software for Embedded and Cyber-Physical Systems." More specifically, in making programming simpler for i) heterogeneous, many-core, and accelerated computing, ii) low-power and error-resilient computing, and that of iii) time-sensitive applications. Now he is a professor in the School of Computing and Augmented Intelligence at Arizona State University, where he has established and heads the Make Programming Simple (MPS) Lab.

**Kyoungwoo Lee** Received Ph.D. in Information and Computer Science from the University of California, Irvine, and B.S. and M.S in Computer Science from the Yonsei University, Seoul, Korea. Now he is a Professor of the Department of Computer Science and Engineering at Yonsei University and AI and Digital Secretary to the President's Office in South Korea. His research interests are Soft error, reliability, fault tolerance, IoT-based medical service, machine learning-based medical service, and artificial neural network accelerators.

**Uiwon Hwang** Received the B.S. degree in biomedical engineering from Korea University, Seoul, Korea, in 2016 and the Ph.D. degree in electrical and computer engineering from Seoul National University, Seoul, Korea, in 2023. He is currently an Assistant Professor in the Division of Digital Healthcare, at Yonsei University. His research interests include deep generative models, data-centric AI, and machine learning.