DI-AA: An Interpretable White-box Attack for Fooling Deep Neural Networks

Yixiang Wang[1], Jiqiang Liu[1*], Xiaolin Chang[1], Ricardo J. Rodríguez[2], Jianhua Wang[1]

[1]Beijing Key Laboratory of Security and Privacy in Intelligent Transportation, Beijing Jiaotong University, Beijing 100044, China

[2]Dept. of Computer Science and Systems Engineering, University of Zaragoza

[1]{18112047, jqliu, xlchang, 20112051}@bjtu.edu.cn, [2]rjrodriguez@unizar.es

Abstract

White-box adversarial example (AE) attacks on deep neural networks (DNNs) have a more powerful destructive capacity than black-box attacks using AE strategies. However, few studies have been conducted on the generation of low-perturbation adversarial examples from the interpretability perspective. Specifically, adversaries who conducted attacks lacked *interpretation from the point of view of DNNs*, and the perturbation was not further considered. To address these, we propose an interpretable white-box AE attack approach, DI-AA, which not only explores the application of the interpretable method of deep Taylor decomposition in selecting the most contributing features but also adopts the Lagrangian relaxation optimization of the logit output and L_p norm to make the perturbation more unnoticeable. We compare DI-AA with eight baseline attacks on four representative datasets. Experimental results reveal that our approach can (1) attack nonrobust models with low perturbation, where the perturbation is closer to or lower than that of the state-of-the-art white-box AE attacks; (2) evade the detection of the adversarial-training robust models with the highest success rate; (3) be flexible in the degree of AE generation saturation. Additionally, the AE generated by DI-AA can reduce the accuracy of the robust black-box models by 16%~31% in the black-box manner.

Keywords: adversarial example, deep learning, interpretability, robustness, white-box attack

1. Introduction

The past years have witnessed notable advances in the development of deep neural networks (DNNs), which have led to breakthroughs in various areas, including but not limited to bioinformatics [1], language learning [2] and causal inference [3]. However, the emergence of adversarial examples (AEs) is threatening the widespread deployment of security-sensitive DNN-based applications [4,5]. In AE attacks, a clean input with a small and unnoticeable perturbation can mislead a well-trained DNN. The vulnerability and counterintuitive behaviors of DNNs toward AE make it difficult for users to trust DNN decisions. Therefore, it is urgent and critical to have a deeper understanding of AE attacks and make DNNs more dependable in practice.

Depending on the ability of adversaries, there are two types of attacks: white-box attacks [4-9] and black-box attacks [10-12,14,15]. The adversaries have complete knowledge of the target model and data information in white-box settings. In contrast, in black-box settings, adversaries can transfer the generated AE to the unknown deployed model based on AE transferability [14]. Empirically, white-box attacks are more powerful for attacking a robust model than black-box attacks [9].

In terms of white-box attacks, Croce *et al.* [13] further divided them into two classes. The first is to minimize adversarial perturbation [7,13], where complicated approaches are employed to allow the perturbation to be as small as possible in L_p norm distances but generally have considerable computational cost. The other is to restrict the perturbation in the U-ball around the input [8,14,15], where low computational cost is caused with a large perturbation introduced. The common point is both approaches mentioned above adopt the first-order gradient information.

However, these white-box approaches still have to address two primary issues:

- The existing traditional approaches lack *interpretation from the point of view of DNNs*. The adversaries only look for a more optimizable landscape to generate AE by gradients without considering what features the DNN actually *learns*. It is intuitive that if adversaries attack the features learned by the DNN thereby causing misclassification to occur more easily.
- The existing studies on interpretation and adversarial examples have only focused on finding the vulnerabilities of the interpretation methods [16,46] by using specific adversarial examples. The quality of adversarial examples is not considered. For instance, Subramanya *et al.* [16] tried to create adversarial patches to fool the target model and the Grad-

CAM interpretation method simultaneously. The perturbation, however, is so large and even recognizable by human eyes.

In summary, few studies have been conducted on the generation of low-perturbation adversarial examples from the perspective of interpretability. This motivates the work described in this paper. Here, we propose an interpretable and effective adversarial example generation approach, namely, the deep Taylor *D*ecomposition *I*terative white-box *A*dversarial example *A*ttack (DI-AA). Unlike the previous approaches, our DI-AA approach uses the interpretable saliency map through the reliable deep Taylor decomposition (DTD) method [18]. With the guideline of this interpretable saliency map, our approach applies heuristic searches and Lagrangian relaxation of the L_p norm constraint to find the features that should be perturbed and to minimize the adversarial perturbation, respectively.

To the best of our knowledge, we are the first to apply the interpretation method, the deep Taylor decomposition method, to low-perturbation AE generation, which makes the attack more transparent and interpretable and makes the perturbation more unnoticeable, as shown in Fig. 1. We summarize the key features in DI-AA:

- DI-AA is interpretable. The deep Taylor decomposition interpretation method is robust to deception by the malicious attack [46], which makes its interpretation more reliable. By the reliable deep Taylor decomposition interpretation method, DI-AA finds the most contributing features and then perturbs these features. This process is intuitive and easy to understand.
- DI-AA is general. DI-AA can intrinsically be used to generate AE in any scenario, especially in any type of dataset, since the deep Taylor decomposition method can interpret different data [18,25].
- DI-AA is low in perturbation. It jointly constrains the L_2 norm distance explicitly and the L_0 norm distance implicitly in order to decrease the perturbation and make the perturbation more unnoticeable.
- DI-AA is flexible in terms of L_0/L_1 and L_2 norm distances under the L_0 and L_2 joint constraint. Concretely, when the adversarial example generation process is saturated, the consequence is that the L_0 and L_1 distance values are decreased while the L_2 distance values increase. The saturated AE generation process means that when the perturbation rate is fixed and the iterations still increase, the crafting rate is always 100%. This can help adversaries generate a flexible AE according to their needs. For instance, if the adversary wants a small L_2 perturbation, it can implement DI-AA that is just saturated. Additionally, if the adversary wants the small L_0 perturbation, it can implement DI-AA that is oversaturated.

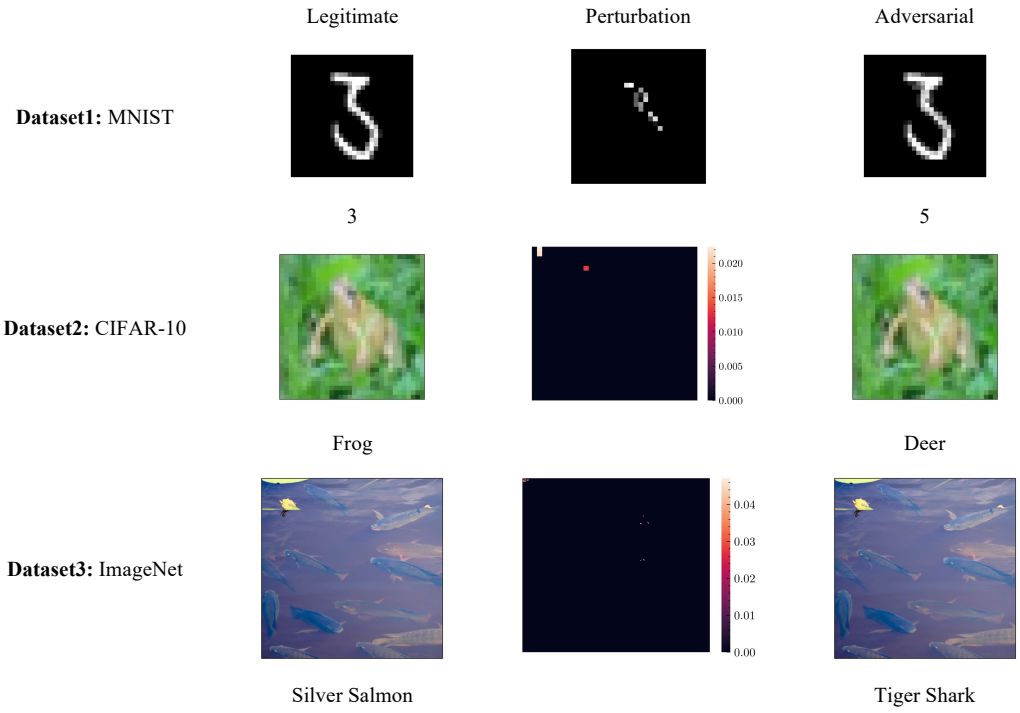

Fig. 1. Examples of AE generated by DI-AA in three datasets. The 'Legitimate' column denotes the original and legitimate examples in the dataset, and 'Adversarial' denotes the corresponding adversarial examples. The 'Perturbation' column is generated by DI-AA to add to the legitimate examples. The model prediction is at the bottom of each figure. In the CIFAR-10 and ImageNet datasets, the 'Perturbation' column is plotted as the heatmap for better color separation.

Extensive experiments were carried out to evaluate our approach. We verified DI-AA on four datasets, namely, NSL-KDD, MNIST, CIFAR-10 and ImageNet, to ensure its effectiveness in various scenarios. Compared with eight baseline methods, our proposed approach can attack the nonrobust models with fairly low perturbation, which is lower than that of AutoAttack [9] and IWA [19] attacks and is close to the L_2-constrained-only attacks, CW [7] and FAB [13]. Moreover, our approach dramatically reduces the number of perturbation points in an AE, compared with L_2-constrained-only attacks. In addition, our approach is more general than the L_0-constrained-only OnePixel method [20]. Furthermore, ours is effective in any dataset, whereas the OnePixel method only performs well on the CIFAR-10 dataset. Moreover, the proposed attack evades the detection by TRADES [21] adversarial training models with the highest success rate. Finally, the generated AE can reduce the robust accuracy of robust black-box models from 16% to 31% in black-box transfer attacks.

The rest of the paper is organized as follows. Section 2 reviews adversarial example attacks, defenses and interpretation methods. We introduce the proposed DI-AA and evaluate it in Sections 3 and 4, respectively. Section 5 concludes the paper and points to future work.

2. Related Work

This section reviews related work on adversarial examples in attacking, defending and interpreting DNNs.

2.1. Adversarial Attacks

As mentioned in [47], adversarial examples are generated for two types of attacks: white-box attacks and black-box attacks.

White-box attacks. Adversarial examples were first proposed by [4], who used the L-BFGS optimizer to find the perturbation. The authors in [5,6] further proposed fast and iterative AE generation approaches, namely, the Fast Gradient Sign Method (FGSM) and Basic Iterative Method (BIM/I-FGSM), to substitute the time-consuming L-BFGS. All these methods were untargeted attacks, in which the adversary does not assign the target label and their purpose is to misclassify the DNN. Subsequently, Carlini *et al.* [7] proposed a strong targeted attack called CW, which was the first to use the logit outputs to generate AE, and the output of AE could be specified to the target label. Since then, white-box attacks have been separated by two goals: minimizing perturbation [9,13] and fast AE generation in the U-ball around inputs [8,14,15]. The latter is generally used in adversarial training for acquiring AE more quickly, and then the adversarial training time can be efficiently reduced.

However, few researchers have paid attention to the influence of interpretable approaches on the generation of AE directly, which we think is a gap to be filled. Specifically, they do not investigate how to generate low-perturbation adversarial examples with the aid of the interpretation method. Wang *et al.* [19] used integrated gradients (IG) [24] to generate AE, but the IG approach has some inherent shortcomings for generating AE, which we will discuss in Section 3.3. This paper attempts to generate interpretable AEs with low perturbation in the white-box manner, considering the strong white-box attack capability.

Black-box attacks. A mainstream black-box attack is the transfer attack [14,15,22]. Specifically, the adversary generates AE in the local substitute DNN and then transfers the AE to attack the unknown deployed models based on the AE transferability. Generally, transfer black-box attacks rely on the optimization capability. Concretely, Dong *et al.* [14] integrated the Momentum optimization method into the BIM attack which dramatically boosted the transferability of AE. Then, researchers began to explore the possibility of integration between the advanced optimization and the BIM attack. Some successful transfer attacks have been proposed, such as NI-FGSM [15] and AI-FGSM [22]. Not only transfer attacks but also other black-box attacks exist [10-12], where the adversary estimates the gradient information by querying and collecting the inputs and outputs of the 'oracle' model and then generates AE by the estimated gradients. One impressive attack is the Zeroth-Order-Optimization (ZOO) attack [10], which monitors the changes in the model output when the input is tuned and then estimates the gradient. One disadvantage of estimating gradients is the low efficiency rate. Therefore, in this paper, we do not pay attention to estimating the gradients. Instead, we transfer our generated AE to unknown robust models to test its transferability.

2.2. Adversarial Defenses

Myriad defensive methods have been proposed to harden the models and defend from the potential threats of AE. Adversarial training (AT) was first proposed in [5] and later improved in [8]. Subsequently, several techniques were proposed to boost adversarial training, such as ensemble training [27] and one-step training [28]. Feature denoising methods were also proposed to distill the input information and remove the perturbation by a specifically designed model layer. For instance, Dhillon *et al.* [48] pruned a subset of the activations and scaled up the surviving activations to normalize each layer by the

proposed Stochastic Activation Pruning (SAP). In addition to modifying the model parameters and/or architectures to make the model robust, some pre-processing methods, such as JPEG compression and image quilting, are also effective in defending AE [49].

However, the emergence of the Expectation over Transformation (EoT) attack [50] broke the pre-processing defensive methods and some feature denoising methods, such as SAP [48]. AT, including TRADES AT [21], is still the most effective approach to making DNNs robust. TRADES AT was proposed to resolve the trade-off between *clean accuracy* and *robust accuracy*. In this paper, *clean accuracy* denotes the accuracy of the nonrobust models, and *robust accuracy* denotes the accuracy of the robust models. TRADES introduced a novel loss function to solve the problem that accuracy does not represent robustness [29]. This paper uses TRADES to adversarially train the targeted models, and then the robust models are attacked using the proposed approach.

2.3 Adversarial Examples with Interpretation Methods

Recently, an enormous amount of work has been devoted to discovering the inner mechanisms of deep networks, especially state-of-the-art convolutional neural networks [17,23-25]. The mainstream interpretation methods focus on the instance-wise interpretation. That is, the interpretation methods provide a contributive saliency map corresponding to the input features. For instance, Sundararajan *et al.* [24] focused on interpreting an individual prediction by the proposed integrated gradients. Inspired by previous work, researchers have attempted to understand AE through interpretable approaches to robust models [26].

Meanwhile, researchers found that some interpretation methods are not reliable and can be fooled by specific adversarial attacks [16,46]. Concretely, Subramanya *et al.* [16] used an interpretable approach, Grad-CAM [17], as the loss function to generate adversarial image patches. The patches, in turn, fool the interpretable heatmap of Grad-CAM. Zhang *et al.* [46] proposed the ADV^2 attack to fool the model without changing the interpretation saliency map, since general aggressive adversarial examples destroy the saliency map. Unfortunately, Section 3.3 shows that ADV^2 fails to be effective on the deep Taylor decomposition method thus empirically validating the deep Taylor decomposition method's reliability.

These previous works only paid attention to misclassifying the interpretation method by adversarial examples but did not consider the quality of the generated AE from the perspective of perturbation and transferability. This paper uses the reliable interpretation method, deep Taylor decomposition, to help generate adversarial examples with low perturbation and high transferability.

3. Methodology

In this section, we first describe the notations and the optimization problems in Sections 3.1 and 3.2, respectively. We then explain the motivation for integrating the deep Taylor decomposition method and present the DI-AA approach in Sections 3.3 and 3.4, respectively.

3.1. Notation

In this paper, we follow the notations from [7]. In classification tasks, given an input with *n*-dimensional features $x \in \circ^{n}$ with the ground-truth label $C^{*}(x)$, a DNN model can be viewed as a sophisticated function $F(x) = y$ that is stacked together by multiple layers and produces the corresponding *m*-class output $y \in \circ^{m}$, where *y* has two properties: $\forall y_{i,i\in\{0,1,..,m-1\}}, y_i \in [0,1]$; $\sum_i y_i = 1$. $\arg\max_i(y_i)$ is referred to as the prediction label and is also denoted by $C(x)$ in this paper. When the model *F* classifies the input correctly, $C(x)$ is $C^{*}(x)$.

Generally, the activation function of the last layer in $F(x)$ is the SoftMax function, turning the logit output to the probability distribution: $F(x) = \text{SoftMax}(Z(x))$. That is, $Z(x)$ is $F(x)$ without the SoftMax activation function. We do not exploit this probability output in our settings but rather the logit output $Z(x)$ to generate AE since the SoftMax function obscures the decision boundary [30]. Thus, the classification result of $Z(x)$ becomes $C(x) = \arg\max_i(Z(x)_i)$.

3.2. Problem Definition

Before constructing the AE, we first define the problem of finding an AE x' for the input x. AE generation can be treated as an optimization problem, where the optimization goal can vary depending on the attacker's needs. Specifically, the AE is

constructed by maximizing the cross-entropy (CE) loss value between the output of $F(x')$ and the ground-truth label $C^*(x)$:

$$\begin{aligned} &\max \ J\left(F(x'), C^*(x)\right) \\ &\text{s.t.} \ \ C(x') \neq C^*(x); \\ &\qquad \|x - x'\|_p \leq \text{Ú} \end{aligned} \tag{3.1}$$

where $J(\cdot)$ represents the CE function. In this way, the loss value can be maximized by gradient-based ascent iteration since a larger CE loss value implies an incorrect classification. In this paper, we propose to use a new optimization objective described in [19] and achieve optimization in two aspects simultaneously: 1) minimize the L_p norm distance, $\|x - x'\|_p$, to ensure small perturbation; and 2) minimize the logit value of $Z(x)_{C^*(x)}$ to cause a misclassification. Hence:

$$\begin{aligned} &\min \ \ Z(x)_{C^*(x)} + c \cdot \|x - x'\|_p \\ &\text{s.t.} \ \ x' \in \text{°}^{\,n} \end{aligned} \tag{3.2}$$

This definition is transparent and easy to interpret. Eq. (3.2) can be viewed as a Lagrangian relaxation version of Eq. (3.1), except that Eq. (3.1) is a maximization optimization, and the objective is different. Let us remark why we do not focus on the loss value but on the logit value $Z(x)$. We believe that in addition to increasing the loss value of x, there are other ways to cause misclassification. Misclassification $C(x') \neq C^*(x)$ also occurs when the logit value $Z(x)_{C^*(x)}$ is reduced to a value that is not the maximum of $Z(x)$. The way to reduce the logit value $Z(x)_{C^*(x)}$ is more intuitive and valid than increasing the loss value. The second term $c \cdot \|x - x'\|_p$ in Eq. (3.2) ensures that the perturbation is as small as possible in the L_p norm distance and c is a hyper-parameter to balance these two optimization tasks. The constraint $x' \in \text{°}^{\,n}$ ensures that the AE is in the valid input domain. In addition, Eq. (3.2) is a suitable optimization function that can be solved by using existing gradient descent optimization algorithms such as Adam [31] and AdaBelief [32].

In Eq. (3.2), the decrease in $Z(x)_{C^*(x)}$ only achieves the untargeted attack. To further exploit its potential, we extend the objective function for the targeted attack, as demonstrated in Eq. (3.3).

$$\begin{aligned} &\min \ \ c \cdot \|x - x'\|_p - Z(x)_t \\ &\text{s.t.} \ \ x' \in \text{°}^{\,n} \end{aligned} \tag{3.3}$$

The intuition in $Z(x)_t$ of Eq. (3.3) is the opposite of Eq. (3.2). Specifically, we want $Z(x)$ to output the desired target label t for targeted attacks, and consequently, $Z(x)_t$ should be the maximum of $Z(x)$. The form of Eq. (3.3) ensures that $Z(x)_t$ will increase iteratively, as $-Z(x)_t$ will decrease by the gradient descent optimizer.

In this section, we present a broad interpretable class of problem definitions about $Z(x)$ for attacks in various settings. We concentrate on the untargeted attack form and use Eq. (3.2) as our objective instance to propose the DI-AA approach in the next section.

3.3. Advantages of the Deep Taylor Decomposition Method

As mentioned in Section 2.1, Wang *et al.* [19] uses the integrated gradients (IG) [24] to generate AE. However, the IG approach has some inherent weaknesses: 1) **Instability**. The IG approach needs to find its corresponding baseline data point for each dataset, e.g., a pure black image for the image dataset. Hence, the baseline dramatically impacts the performance of the IG approach. 2) **Computational cost.** The IG approach is designed to sample around the input x, and the frequency is user-defined. Hence, a lower frequency will affect the performance. 3) **Nonconservativeness.** The sum of each input feature contribution generated by the IG method is not equal to the model output, which we call *nonconservativeness*. Here, the contribution is defined as the input feature's importance to the model output and is also called the relevant score. Intuitively, the

sum of contributions should coincide with the model output. Otherwise, it will induce unnecessary and irrelevant contributions [18]. 4) **Unreliability.** As introduced in Section 2.3, the ADV^2 attack [46] can fool the model while not changing the interpretation heatmap. Fig. 2 shows that ADV^2 can fool the IG interpretation method as well as Grad-CAM but not the deep Taylor decomposition method.

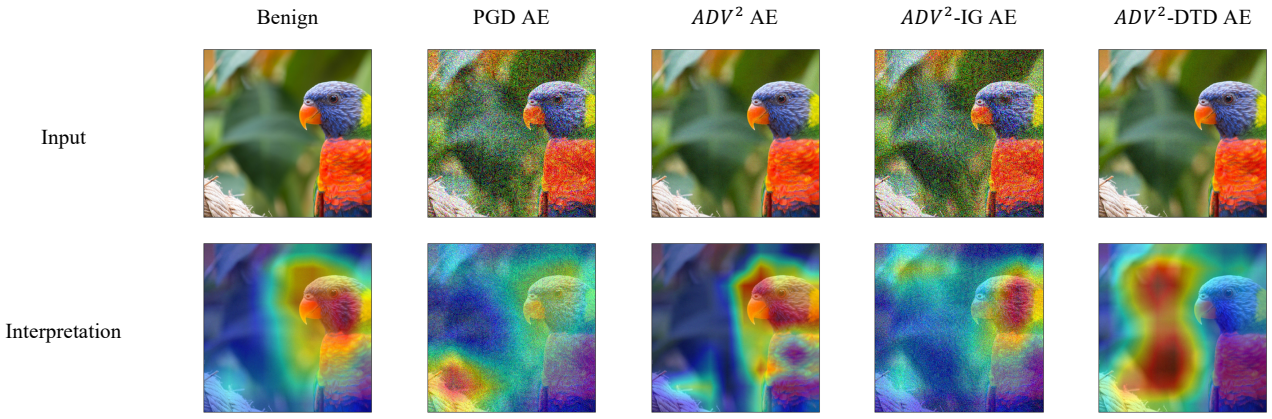

Fig. 2. The visualization of five images and their corresponding interpretable heatmap images. In the ADV^2 setup (last three columns), the greater the adversarial heatmap intersects with the benign heatmap 'lorikeet', the better ADV^2 fools the interpretation method. The interpretable image of the PGD adversarial example fails to highlight the most informative part because the traditional AE cannot retain the interpretation area. The nonoverlapping area between the interpretation heatmap of ADV^2-DTD and 'lorikeet' indicates that ADV^2 cannot fool the DTD interpretation method.

To overcome these weaknesses, a more theoretically complete method, namely, the deep Taylor decomposition interpretation method, was proposed in [18]. This method considers the individual neuron in the DNN as a function that can be decomposed by Taylor decomposition, and the output of the neuron can be decomposed into the input contribution. This method has the following three properties:

1) **Conservativeness**: $\forall x : Z(x) = \sum_i R(x_i)$. Compared to the nonconservativeness mentioned above, the conservativeness property ensures that no confusing and noisy contributions are generated when the output value is decomposed from the last layer to the input variable. The $R(x_i)$ function refers to the relevant score of the input feature x_i.

2) **Positive**: $\forall x_i : R(x_i) \geq 0$. This property ensures that each input feature x_i has a nonnegative relevant score.

3) **Reliability**: This property ensures that the output of the deep Taylor decomposition is reliable and can resist malicious attacks.

With these *three* properties, the deep Taylor decomposition method has tighter restrictions than the IG approach. More importantly, compared to the IG approach, the decomposition approach does not rely on the baseline input and consequently calculates the relevant values more robustly. In what follows, we use Fig. 3 to illustrate the advantages of the decomposition approach. As shown in the 1st row of Fig. 3, the relevant map of the IG approach (middle picture) is not appropriate, as the background should not contribute to the model outputs. In contrast, the elements in the relevant map of the decomposition approach are positive in the area around the number '2'. In the 2nd row, the cup texture obtained by the IG approach (middle picture) is not as nuanced as the one obtained by the deep Taylor decomposition method (right picture). Therefore, based on these distinctive advantages, we adopt this interpretable approach to generate AE.

Since the deep Taylor decomposition method is complex, we present it briefly here, and more details can be found in [18]. Assume that the output of the deep network has been decomposed into one neuron a_j of the *j*-th layer and R_j is the associated relevant score. $\{a_i\}$ are the neurons connected with a_j in the previous *i*-th layer. We define a root point r, and the root point is not specified practically since we can always find it implicitly in the intersection of a plane with the help of the w^2-rule. Then, the Taylor decomposition of R_i in terms of a_i in the previous layer with respect to R_j is:

$$R_i = \sum_j \frac{\partial R_j}{\partial a_i}\Big|_{r^j} \cdot (a_i - r) = \sum_j \frac{w_{ij}^2}{\sum_{i'} w_{i'j}^2} R_j$$

where w_{ij} is the weight between the *i*-th and *j*-th layers. The decomposition method spreads the above function from the output layer to the input layer, and therefore, it is the deep Taylor decomposition.

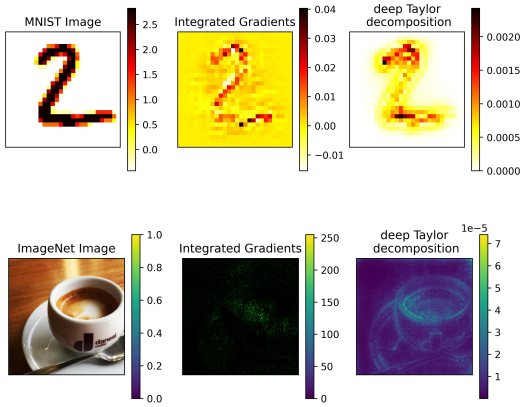

Fig. 3. Contrasting relevant maps between the IG approach and the deep Taylor decomposition in image '2' of the MNIST dataset and 'espresso' of the ImageNet dataset

3.4. DI-AA Approach

In this section, we present a new AE attack approach named DI-AA. DI-AA combines the deep Taylor decomposition interpretable method with the conventional AE generation approach. Furthermore, DI-AA reduces perturbation by explicitly constraining the L_2 norm distance and implicitly constraining the L_0 norm distance. The algorithm is described in **Algorithm 1** as follows.

Algorithm 1: **DI-AA**

Input: the legitimate input x with the corresponding label $C^*(x)$; the model learned function $Z(\cdot)$; the perturbation rate ε; the iterations T

Output: adversarial example x'

1. $x' = \text{clone}(x)$
2. $saliency\_map = \text{DeepTaylorDecomp}(Z(x), x, C^*(x))$
3. $sorted\_map = \text{sort}(saliency\_map)$
4. $masked = \text{zeros\_like}(x)$
5. **for** $idx = 0,1,\text{L}\ ,\text{len}(sorted\_map)$ **do**
6. $masked[sorted\_map[idx]] = 1$
7. $x' = \text{AEGen}(x', C^*(x), Z(\cdot), T, \varepsilon, masked)$
8. **if** $\text{argmax}(Z(x')) \neq C^*(x)$: **return** x'
9. **end for**
10. **return** x'

Algorithm 1 first initializes the original AE x' by cloning the legitimate input x, as shown in Line 1. Then, in Line 2, the deep Taylor decomposition method is used to obtain the contributive saliency feature map, which is the standard to guide the order of the features to be perturbed. In Line 3, the $\text{sort}(\cdot)$ function sorts the saliency map and returns the feature index in descending order. Here, we perturb one feature at a time in the AE generation process, which is what the iterative code in Line 5 does. To guarantee the individual perturbation point, we introduce a '**0**' masked matrix *masked* as defined on Line 4. When feature *i* must be perturbed, $masked[i] = 1$ (Line 6). Subsequently, a one-feature-perturbation loop is implemented to generate AE by **Algorithm 2** (Line 7). An obvious advantage of perturbing one feature at a time is that in this way, we can avoid pulling the redundant and unnecessary perturbation to the AE and restrict the features to be perturbed. Therefore, the L_0 distance is

implicitly bounded. When the AE is generated by **AEGen** (defined in **Algorithm 2**), the condition in Line 8 checks if it is valid: if it can cause a misclassification, it is a suitable AE; otherwise, it will continue to iterate through the loop.

Algorithm 2: **AEGen**

Input: AE x'; the legitimate input x and with the corresponding ground-truth label $C^*(x)$; the model learned function $Z(\cdot)$; the perturbation rate ε; the iterations T; the masked matrix $masked$.

Output: adversarial example x'

1. **for** $i = 1, 2, \ldots, T$ **do**
2. $obj = \text{OneHot}(C^*(x)) \text{ e } Z(x') + c \cdot L_2(x, x')$
3. $grad = \text{optim}(obj, x')$
4. $x' = x' - \varepsilon \cdot grad \text{ e } masked$
5. $x' = \text{clip}(x', clip\_min, clip\_max)$
6. **if** $\text{argmax}(Z(x')) \neq C^*(x)$**: return** x'
7. **end for**
8. **return** x'

In **Algorithm 2**, **AEGen** takes T iterations to generate an AE. Line 2 represents the objective function given in Eq. (3.2). Concretely, the Hadamard product of one-hot encoding of $C^*(x)$ and $Z(x)$ is $Z(x)_{C^*(x)}$. Then, the objective function consists of the sum of $Z(x)_{C^*(x)}$ and the L_2 norm distance. It is noteworthy that in Line 2, we set $p = 2$ to instantiate Eq. (3.2) since the previous work [19] found that $p = 2$ is more effective than other p values. When solving the objective function, the $\text{optim}(\cdot)$ function calculates the derivatives of the objective function with regard to x'. The Hadamard product of the *masked* matrix and the derivatives is manipulated to obtain the perturbation. The step ε plays an important role in controlling the perturbation in each iteration. The $\text{clip}(\cdot)$ function restricts each element of x' within the legitimate range $[clip\_min, clip\_max]$ to satisfy the input domain. In this paper, the legitimate domain is set to $[0,1]$ in all datasets.

From **Algorithm 1** and **Algorithm 2**, we can conclude that the time complexity of DI-AA, $T(\text{DI-AA})$, is $O(n^2)$ since the two **for** loops (Line 5 in **Algorithm 1** and Line 1 in **Algorithm 2**) are nested in DI-AA. The quadratic time complexity relies on the number of input features and the iteration numbers, which are the implication of two **for** loops. When the number of input features and/or iterations becomes larger, DI-AA needs more time to generate adversarial examples.

4. Experimental Results and Analysis

This section demonstrates the effectiveness of DI-AA through experiments. We first present the datasets, models, baselines used in the experiments and hyperparameters of DI-AA. We then introduce the white-box attacks to the nonrobust models and robust models. Nonrobust models are the models that are not adversarially trained. Robust models are the models trained by the adversarial defensive method (TRADES in this case). Finally, to further investigate the transferability of the AE generated by DI-AA, we attack robust models in the black-box manner.

4.1. Setup

TABLE 1. Dataset information (#Training denotes the number of the training set)

Dataset	#Training	#Testing	Size	#Classes
NSL-KDD	395345	61388	1×122	5
MNIST	60000	10000	1×28×28	10
CIFAR-10	50000	10000	3×32×32	10
ImageNet	-	1000	3×299×299	1000

Dataset. We focus on four datasets to comprehensively validate the effectiveness of DI-AA, as we tested our approach not only on unstructured datasets such as MNIST [33], CIFAR-10 [34] and ImageNet [35], but also on the structured dataset such as NSL-KDD [36]. The dataset information is shown in TABLE 1. One thing to note is that the size of ImageNet validation sets is enormous. Therefore, we randomly selected 1000 samples as the test set to verify the effectiveness of DI-AA, and each sample corresponds to a unique class, as established in previous papers [14,15].

Models. TABLE 2 shows the details of the KDD model and the MNIST model. For the CIFAR-10 model, we used the standard ResNet-18 model [37] and omit its structure in TABLE 2. For detailed information about ResNet, please refer to [37]. The AdaBelief optimizer [32] was implemented to boost the performance of the models in the CIFAR-10 training phase. In the ImageNet dataset, we used the pretrained ResNet-34 model for subsequent experiments.

TABLE 2. The structure of the KDD and MNIST models

Layer	KDD model	MNIST model
$\mathrm{ReLU}(\mathrm{Conv})$	–	$3\times3\times32$
$\mathrm{ReLU}(\mathrm{BatchNorm}(\mathrm{Conv}))$	$1\times4\times16$	$3\times3\times32$
MaxPool	–	2×2
$\mathrm{ReLU}(\mathrm{Conv})$	–	$3\times3\times64$
$\mathrm{ReLU}(\mathrm{BatchNorm}(\mathrm{Conv}))$	$1\times3\times32$	$3\times3\times64$
MaxPool	–	2×2
$\mathrm{ReLU}(\mathrm{BatchNorm}(\mathrm{Conv}))$	$1\times3\times64$	–
Dropout	0.3	0.3
$\mathrm{ReLU}(\mathrm{FC})$	1024×256	1024×512
FC	256×10	512×10

Model Performance. Based on the normally trained models, the TRADES adversarial training approach was adopted to improve the robustness of the trained KDD, MNIST, and CIFAR-10 models. TRADES hyperparameters are provided by the authors' default settings [21]. The results of the nonrobust (*Clean Accuracy*) and robust models (*Robust Accuracy*) are shown in TABLE 3. The *Robust Accuracy* of the ImageNet model is '-' for the following reasons: the hyperparameters of TRADES on the robust ImageNet model are not provided in [21]; it is difficult to train a high-dimensional ImageNet model and find the best TRADES hyperparameters confined by the equipment resource.

TABLE 3. Model performance

Model	Clean Accuracy	Robust Accuracy
KDD model	88.18%	86.85%
MNIST model	99.54%	99.52%
CIFAR-10 model	94.78%	80.38%
ImageNet model	90.60%	-

Baselines. We compared DI-AA not only with L_2-constrained-only methods, such as *CW* [7] and *FAB* [13] but also with the L_0-constrained-only method, *OnePixel* (OnePix) [20]. Other conventional methods, FGSM families, and the state-of-the-art method, *AutoAttack* (AutoA) [9], are considered. FGSM families include *FGSM* [5], *BIM* [6] and *PGD* [8]. *IWA* [19] was also considered to verify the effectiveness of the deep Taylor decomposition method. FGSM, BIM, PGD, and CW methods were implemented in the Advertorch PyTorch Library [38]. OnePixel attack was implemented in the TorchAttacks PyTorch Library [39]. The code of AutoAttack can be found in [9].

Evaluation Metrics. To evaluate the effectiveness of DI-AA, we used four metrics: 1) *mean running time* (MRT) to evaluate the mean time required for the generation of one adversarial sample; 2) *accuracy score* (ACC) to assess the transferability of AE in the black-box setting; 3) the L_p, $p \in \{0,1,2\}$, norm metrics to measure the perturbation; and 4) *success rate* (SR) to evaluate effectiveness. The formula is:

$$SR = \frac{\#\text{adversarial samples}}{\#\text{correctly classified samples}}$$

4.2. Effect of Hyperparameters

This section investigates the impact of three hyperparameters, perturbation rate ε, iterations T and constant c in the objective function, on the attack performance.

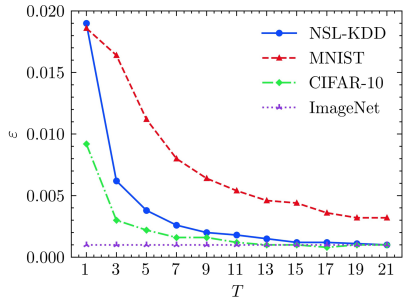

Fig. 4. The trend of ε when T varies from 1 to 21

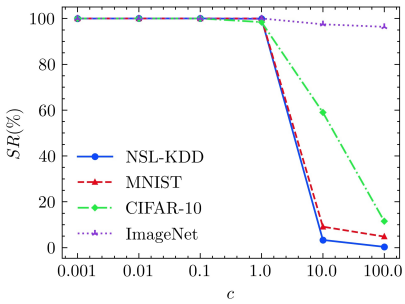

Fig. 5. The effect of c to the SR metrics

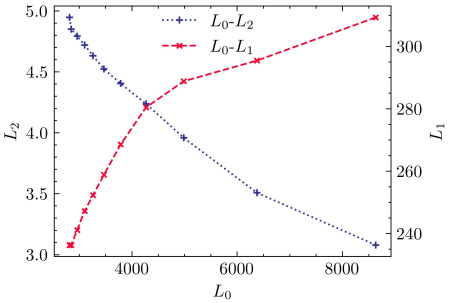

Fig. 6. The relationship between L_p mean metrics in a saturation

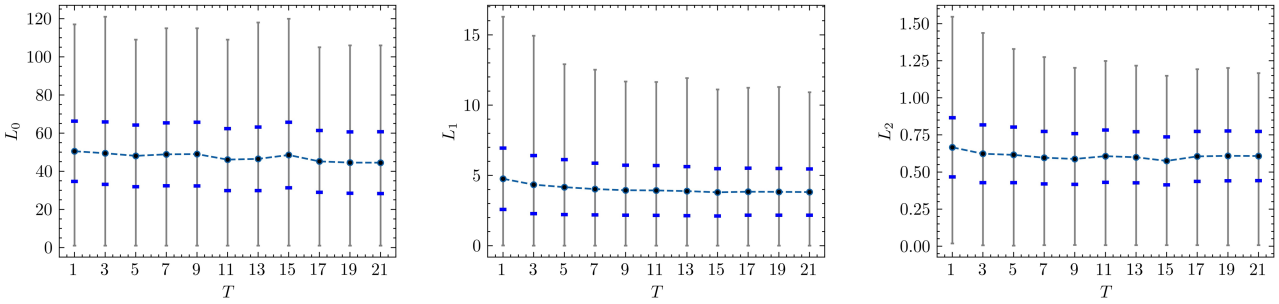

Fig. 7. L_p stacked error bars of NSL-KDD dataset when iteration T varies from 1 to 21 (L_0, L_1, L_2 from left to right)

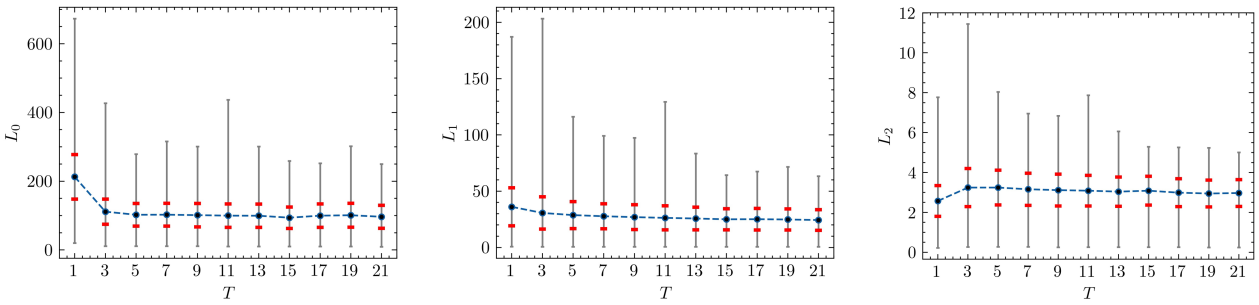

Fig. 8. L_p stacked error bars of MNIST dataset when iteration T varies from 1 to 21

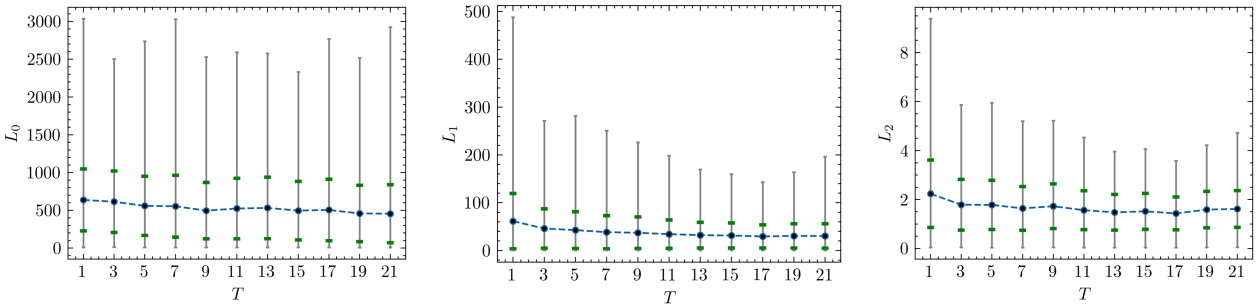

Fig. 9. L_p stacked error bars of CIFAR-10 dataset when iteration T varies from 1 to 21

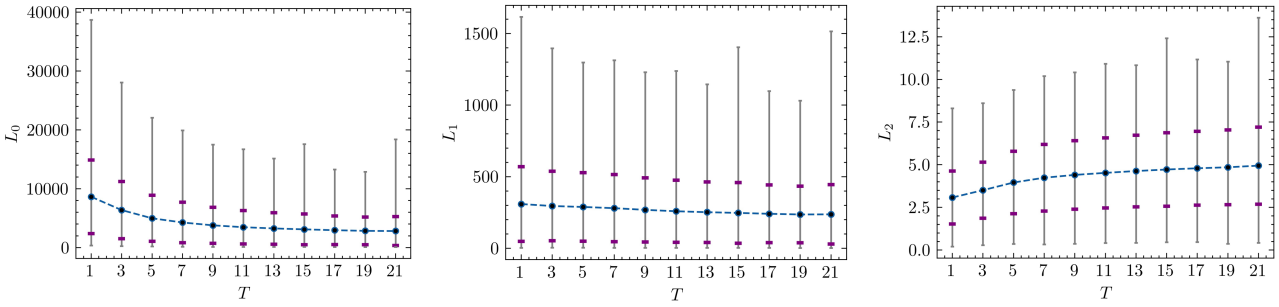

Fig. 10. L_p stacked error bars of ImageNet dataset when iteration T varies from 1 to 21

4.2.1. ε and T

We investigated the impact of ε and T in the objective function on the attack capability in terms of the L_p norm metrics. The first 10% of the test set samples of the four datasets were used to run the proposed approach on the nonrobust models. When the SR score reached 100%, we recorded the statistics of the L_p evaluation metrics with the corresponding ε and T. Evaluation metrics are plotted by stacked error bars to show trends, as shown in Figs. 7~10 for the four datasets. A stacked error bar contains the four statistical values: mean, standard deviation (std), minimum and maximum, which correspond to the dot in the middle, two squares around the dot, and the highest and lowest points in a vertical line, respectively.

We first examine the NSL-KDD dataset. Fig. 7 and the blue curve in Fig. 4 indicate that when the iteration increases, the declining trend appears in the three L_p figures and ε curve. Of these, the ε and L_1 mean curves show the smoothest declining trend. The L_0 and L_2 mean curves fluctuate slightly. For the standard deviation, all L_p std values gently decrease. The maximum trends of L_1 and L_2 decrease with random fluctuation. However, the L_0 maximum trend fluctuates irregularly, and we cannot find any useful information. Additionally, the three L_p minimum trends are close to 0, which indicates that AE does exist around the legitimate input. Empirically, when $17 \le T \le 21$ and $\varepsilon = 0.001$, DI-AA performs the best with L_p metrics comprehensively considered.

In the MNIST dataset, the ε curve (the red curve) trend in Fig. 4 is similar to that of the NSL-KDD datasets, but the trends of the L_p statistical values differ from those of the NSL-KDD dataset. Fig. 8 shows that the L_0 and L_1 mean curves show downward trends with slight fluctuations. In contrast, the L_0 mean curve on the NSL-KDD dataset fluctuates more. One unexpected finding is that the L_2 mean trend rises at first and then declines, but the difference in the L_2 mean value is only 0.3 numerically when $T = 21$ and $T = 1$, which we think is still reasonable. Changes in the trend of the L_p std values in the MNIST dataset are not different from those in the NSL-KDD dataset. Nevertheless, the L_p maximum trends fluctuate more than those of the NSL-KDD dataset. In summary, when $T = 21$ and $\varepsilon = 0.003$, the proposed approach performs best.

For the CIFAR-10 dataset, Fig. 9 reveals that there is a steady drop in the L_0 and L_1 mean curves. It is striking for the L_2 mean curve to drop at first but rise at $T = 17$. The L_1 and L_2 std trends drop steadily, while the L_0 std trend fluctuates more than the L_1 and L_2 std trends. In terms of the L_p maximum trends, the L_1 and L_2 trends are similar to each other, and both drop steadily with fluctuations and rise when $T > 17$. However, the L_0 maximum trend is haphazard, and no clue can be summarized. What can be seen in Fig. 4 is that the ε curve (the green curve) steadily declines with slight fluctuations. Empirically, when $T = 17$ and $\varepsilon = 0.001$, the proposed DI-AA performs best on the CIFAR-10 dataset.

In the ImageNet dataset, the ε curve (the purple curve) in Fig. 4 shows that DI-AA always reaches a 100% success rate when $\varepsilon = 0.001$. This means that the AE generation process is saturated. To discover how L_p changes when the generation process is saturated, we further increase the number of iterations. The results are shown in Fig. 10.

Interestingly, the L_1 mean curve decreases slowly with increasing T. The L_0 mean curve decreases sharply and obviously at the beginning and flattens afterward. Both the L_0 and L_1 mean curves show a decreasing trend. Instead, the trend of the L_2 mean curve diverges from the L_0/L_1 mean trends. Concretely, the L_2 mean curve rises steadily with iteration growth. The opposite trends of L_0/L_1 and L_2 reveal that when the success rate is saturated, the perturbation points (L_0) and the total perturbation (L_1) decrease, and the Euclidean distances (L_2) increase with increasing iteration. Similarly, the L_0 and L_1 std trends decrease when saturated, while the L_2 std trends continue to increase.

To better illustrate the opposing trends, we plot Fig. 6 to show the opposite variations. From Fig. 6 and Fig. 10, we can see that when L_0 increases, L_2 increases concurrently, but L_1 decreases inversely. From the viewpoint of AE, the two figures directly reveal that when the generation process is saturated, DI-AA adds more perturbation at a point. When a point is given more perturbation, DI-AA requires fewer perturbation points to reach misclassification. Then, the total perturbation is implicitly decreased since there are fewer perturbation points. However, because the changes in a perturbation point are larger, the Euclidean distance is larger than before. Hence, we can infer that DI-AA is a flexible AE generation approach according to the adversaries' demands: if the adversary wants the small L_2 perturbation, it can implement DI-AA whose process is just saturated. Alternatively, if the adversary wants the small L_0 perturbation, it can implement DI-AA whose process is oversaturated.

By comparison with baselines, we choose the same iterations as IWA (i.e., $T = 7$). Since there are no experiments on the ImageNet dataset in the IWA setup, we run this attack on the ImageNet dataset and find that when $T = 7$, IWA performs best.

4.2.2. Constant *c*

The four curves in Fig. 5 show that our approach is effective for a constant c in $(0,1]$ in all datasets. When $c > 1$, the objective function in Eq. (3.2) focuses on minimizing the L_p distance rather than $Z(x)_{C^*(x)}$, which implicitly affects the AE generation. $Z(x)_{C^*(x)}$ is critical to generating AE, and when $Z(x)_{C^*(x)}$ is sufficiently small, AE can be generated. Therefore, $c > 1$ has a negative effect on AE generation. We also observe that the negative effect decreases as the input size becomes large. Empirically, a larger input size implies a larger $Z(x)_{C^*(x)}$, and the value of $c \cdot \|x - x'\|_p$ is relatively steady. In this way, when the input size becomes larger, the negative effect of a larger c decreases.

4.2.3. Summary

In summary, based on the conclusions in Sections 4.2.1 and 4.2.2, the hyperparameters (iteration T, perturbation size ε and constant c) that work best on the four datasets are shown in TABLE 4. Specifically, the hyperparameter c is in the range $(0,1]$ on the four datasets, and T on the NSL-KDD dataset is in the positive integer range [17, 21].

TABLE 4. Hyperparameters in four datasets

Dataset	T	ε	c
NSL-KDD	$[17,21]$	0.001	$(0,1]$
MNIST	21	0.003	
CIFAR-10	17	0.001	
ImageNet	7	0.001	

4.3. White-box Attack on Nonrobust and TRADES-robust Models

TABLE 5. Comparison of eight AE white-box attacks to the nonrobust model on the NSL-KDD dataset

Attacks	L_0		L_1		L_2		SR
	Mean	Std	Mean	Std	Mean	Std	
Abl-L_0	20.41	21.67	18.07	19.51	3.14	2.72	94.91%
Abl-L_2	70.31	18.13	18.22	5.64	2.34	0.58	90.15%
Ours	18.94	6.83	*6.71*	**2.63**	**1.76**	*0.49*	**100.00%**
FGSM	109.92	11.96	91.53	20.12	12.24	7.39	17.10%
BIM	101.90	12.15	31.80	17.00	7.73	8.93	31.08%
PGD	122.00	0.00	119.12	5.13	10.86	**0.35**	70.91%
OnePix	**4.98**	**0.14**	**6.17**	4.79	3.30	3.81	64.71%
CW	121.58	3.49	11.40	10.56	3.60	6.52	100.00%
FAB	103.85	18.17	19.85	13.23	4.75	6.46	95.68%
IWA	10.19	5.95	8.79	5.09	2.90	1.27	93.83%
AutoA	100.87	10.42	31.33	15.74	6.64	7.87	44.55%

TABLE 6. Comparison of eight AE white-box attacks to the nonrobust model on the MNIST dataset

Attacks	L_0		L_1		L_2		SR
	Mean	Std	Mean	Std	Mean	Std	
Abl-L_0	93.70	30.89	27.01	10.87	3.25	0.83	100.00%
Abl-L_2	493.30	28.23	61.80	12.79	3.11	0.68	99.98%
Ours	*100.03*	28.12	27.67	9.82	3.14	0.74	**100.00%**
FGSM	469.91	30.73	131.38	9.11	6.24	0.22	87.21%
BIM	474.97	23.66	131.65	5.35	6.24	0.13	93.55%
PGD	475.42	23.05	131.84	5.13	6.25	**0.12**	93.27%
OnePix	**29.26**	**0.84**	15.78	**1.36**	3.22	0.22	47.89%
CW	781.67	4.32	**11.42**	4.11	**1.09**	0.34	**100.00%**
FAB	535.17	31.24	12.65	4.19	1.12	0.32	**100.00%**
IWA	177.29	111.86	62.78	45.01	5.08	2.08	99.00%
AutoA	727.70	105.21	124.33	5.88	5.42	0.33	**100.00%**

TABLE 7. Comparison of eight AE white-box attacks to the nonrobust model on the CIFAR-10 dataset

Attacks	L_0		L_1		L_2		SR
	Mean	Std	Mean	Std	Mean	Std	
Abl-L_0	458.97	330.85	39.14	33.90	1.88	1.00	94.60%
Abl-L_2	3053.40	71.79	98.73	29.63	1.97	0.71	99.10%
Ours	485.51	382.57	31.38	25.97	1.57	0.74	99.94%
FGSM	3053.55	72.24	823.77	63.38	15.35	0.74	87.82%
BIM	3053.27	71.97	823.73	63.17	15.34	0.73	98.45%
PGD	3053.23	72.31	823.67	63.31	15.35	0.74	98.24%
OnePix	**15.00**	**0.11**	6.13	**1.25**	1.80	0.31	73.73%
CW	3071.86	0.96	**4.88**	4.23	**0.14**	0.11	99.98%
FAB	3054.75	65.95	5.21	4.30	**0.14**	0.11	99.89%
IWA	182.07	252.17	85.03	101.86	10.11	7.12	98.98%
AutoA	3066.01	24.31	66.55	3.34	1.32	**0.05**	**100.00%**

TABLE 8. Comparison of eight AE white-box attacks to the nonrobust model on the ImageNet dataset

Attacks	L_0		L_1		L_2		SR
	Mean	Std	Mean	Std	Mean	Std	
Abl-L_0	4375.11	3488.27	295.90	246.50	4.39	2.00	100.00%
Abl-L_2	264844.27	9796.20	1932.51	276.62	3.78	0.60	100.00%
Ours ($T=7$)	4246.85	3407.10	279.19	233.42	4.23	1.95	**100.00%**
Ours ($T=1$)	8631.26	6248.76	309.30	261.07	3.08	1.56	**100.00%**
FGSM	264844.81	9816.32	68862.34	6365.00	139.56	7.88	99.45%
BIM	229348.09	9047.41	15554.36	387.23	36.38	0.76	**100.00%**
PGD	267611.77	1694.58	35879.61	2731.85	82.76	4.26	**100.00%**
OnePix	**104.91**	**0.52**	**35.54**	**4.50**	4.18	0.46	17.68%
CW	268176.31	53.27	149.01	131.10	0.38	0.31	**100.00%**
FAB	264975.19	8733.16	52.34	38.97	**0.16**	**0.11**	**100.00%**
IWA	1667.83	1629.75	221.24	234.70	4.80	3.13	76.28%
AutoA	267394.50	3198.85	46672.36	7117.65	103.02	11.91	**100.00%**

We used the hyperparameters obtained in Section 4.2 to perform a white-box attack on the nonrobust and TRADES-robust models. We take the first look at the ablation experiments. Then, we analyze the efficiency comparisons and the results of white-box attacks.

4.3.1. Ablation Analysis

In the first three rows of TABLE 5~TABLE 8, we show the results of ablation experiments conducted to verify whether the deep Taylor decomposition method and L_p norm are effective. Specifically, Abl-L_0 denotes that DI-AA has no L_p norm constraint. That is, the objective function of DI-AA in Eq. (3.2) does not include $c \cdot \|x - x'\|_p$. Abl-L_2 denotes that DI-AA does not have the deep Taylor decomposition method as the guideline to help select the perturbation points. This implies that Algorithm 1 should not include Lines 2~5. 'Ours' denotes the complete DI-AA attack.

From the first three rows of the four tables, we observe three phenomena: 1) without the L_2 norm constraint, the L_1 and L_2 norm distances of Abl-L_0 are larger than that of DI-AA; 2) without the deep Taylor decomposition method, the L_0 norm distance of Abl-L_2 is much larger than that of DI-AA; 3) Abl-L_0 and Abl-L_2 sometimes can lead to the loss of SR scores, especially in the structured NSL-KDD dataset. In contrast, our DI-AA can finely balance the SR score and L_p norm distances. In particular, DI-AA implicitly minimizes the L_0 value and constrains the L_1 and L_2 values at the same time in the four datasets.

To further verify the effectiveness of DI-AA, we plot the process to demonstrate how DI-AA attacks an MNIST image '1' by the specific contributing features provided by the deep Taylor decomposition method. The attack flow is shown in Fig. 11. From the top of Fig. 11, we can see the legitimate image '1' and its corresponding SoftMax output of the MNIST model. We only plot the top-best and second-best class probabilities of the output since the sum of these two class probabilities is almost 99%. We thus omit 1% probabilities of the other 8 classes for clarity and brevity. There is 75% confidence that the model thinks the image is '1' from the output probabilities shown in the top right of the figure. When the image is fed into DI-AA, DI-AA implicitly produces the relevant map of the input by the deep Taylor decomposition method. With the guideline of the relevant map, DI-AA heuristically perturbs the single feature with the highest contribution score. To better visualize the attack process, we plot the variation of the model SoftMax output when DI-AA perturbs different numbers of high contributing features, as the bottom of Fig. 11 shows. From the figure, we observe that when the more contributing features are perturbed, the probability of class '1' decreases, and the probability of the wrong class '3' grows larger; therefore, misclassification occurs.

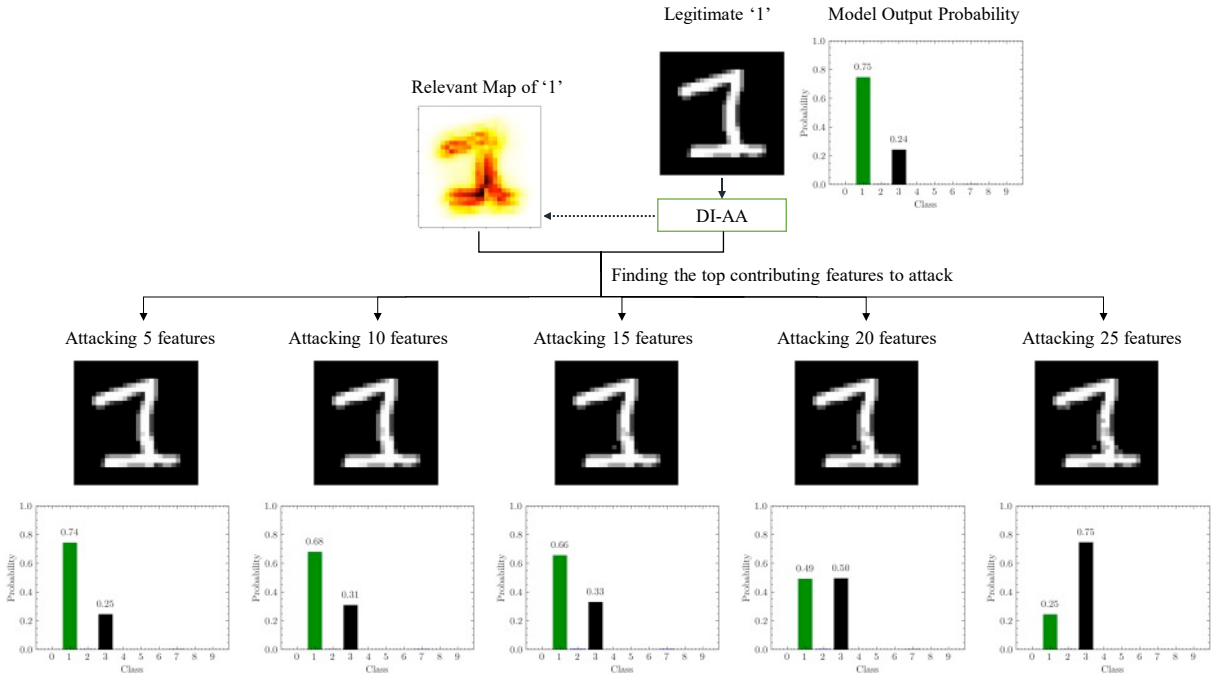

Fig. 11. Visualization of DI-AA attacking one image '1'

4.3.2. Analysis of White-box Attacks on Nonrobust Models

TABLE 5~TABLE 8 are the comparisons between our approach and eight baselines on four datasets. We show the best performance results in bold, and if our result is second-best, it is italicized. An ideal adversarial attack requires fewer perturbation points (L_0 metric), less total perturbation (L_1 metric), and less Euclidean distance (L_2 metric) with a high success rate (SR score). We follow this criterion to evaluate the attack performance. The 100% SR score denotes that all the samples that are classified correctly by the nonrobust model are changed to adversarial examples via attack methods.

On the NSL-KDD dataset, as shown in TABLE 5, our approach achieves the best SR score with the least perturbation and outperforms the other eight baselines. Specifically, the total perturbation and the Euclidian distance are the lowest. The perturbation points are the second least. The IWA attack reaches the third-best SR score, but its perturbation is the second-least among baselines. In particular, the perturbation points are the lowest. CW also reaches the 100% SR score with the third-least perturbation. The SR score of the FAB attack is 4% lower than ours, and its perturbation is slightly larger than that of CW. The poor performance of the AutoAttack and OnePixel methods indicates that researchers should test the proposed method on different scenarios, such as the types of datasets, to verify the effectiveness of the approach. Specifically, OnePixel overly concentrates on the L_0 constraint, which causes its L_2 distance to be larger and the SR score to be lower. Other baselines, such as the FGSM family, do not perform well in the structured dataset.

On the MNIST dataset, TABLE 6 shows that our approach achieves the best SR score with the third-least perturbation and the fewest perturbation points. CW, FAB and AutoAttack also reach the best SR score, but their perturbation increases progressively. Specifically, the perturbation points of CW and FAB are larger than ours. In the CW setup, they project the inputs to the new space, which always causes their perturbation points to be the largest. The IWA reaches the second-best SR score and perturbation points. Its total perturbation and Euclidean distance are lower than those of AutoAttack but larger than ours. The comparison between IWA and ours implies that the deep Taylor decomposition method effectively guides adversarial generation. In terms of the OnePixel method, its larger L_2 distance and lower SR score are as poor as those in the NSL-KDD dataset. The performance of FGSM families is consistent with that in the NSL-KDD dataset.

The results of the CIFAR-10 dataset are shown in TABLE 7. Our method achieves the third-best SR score, which is merely 0.04% lower than the second-best SR score of CW and 0.06% lower than the best SR score of AutoAttack. CW and FAB attacks achieve the lowest perturbation equally, but CW is more stable than FAB since the SR score of FAB is 0.09% lower than that of CW. AutoAttack, although it achieves the best SR score, has a total perturbation that is quite large compared with ours and CW. IWA does not perform as well as it does in the MNIST dataset since its perturbation is large, but the SR score is trivial. The OnePixel method performs well on the CIFAR-10 dataset compared with NSL-KDD and MNIST. However, its performance is still not good. Although its L_0 and L_1 values are smaller than others, its SR score is the lowest, and the L_2 values are larger than ours.

We also tested our method on the large-scale ImageNet dataset, whose results are shown in TABLE 8. We show the results for $T = 1$ and $T = 7$. These two results can explicitly reflect the phenomenon mentioned in Section 4.2.1: the Euclidean distance increases while the perturbation points and the total perturbation decrease oppositely when the generation process is saturated and the iteration still increases. Our method achieves the best SR score and the third-least perturbation. FAB performs best among baselines, whose perturbation is the least and the SR score is the best. This is followed by CW. Although AutoAttack obtains a 100% SR score, its perturbation is even larger than that of PGD and BIM, which is not effective. IWA has the same perturbation as ours ($T = 7$), but its SR score is lower, indicating that the IG method has a limited guideline in the large-scale dataset. OnePixel performs poorly in the large-scale images, whose SR score is only 17.68%. Only restricting the L_0 distance is not proper since its success rate reflects that this attack is not effective. FGSM families perform well, and the disadvantage is that the perturbation is large.

In summary, when attacking nonrobust models, the interpretable method can help generate adversarial examples. Generally, DI-AA can generate adversarial examples with a high success rate in any scenario. The perturbation of DI-AA is lower than the baselines in the structured dataset. Additionally, in the unstructured datasets, the perturbation of DI-AA is close to CW and FAB attacks and lower than AutoAttack and IWA attacks.

4.3.3. Efficiency Analysis

As emphasized in Section 3.3, the computational cost of the IG method is very high, and if used to generate adversarial examples, it would be more time-consuming. To test this claim, the mean running time (MRT) indicator is used to compare the efficiency between ours and eight baselines, including IG-based IWA. The results are shown in TABLE 9. We only record the MRT on the nonrobust models since the running time of attacks on the nonrobust model is more realistic. We also omit the MRT of the ablation experiments since the ablation types of DI-AA are not deployed practically.

One clear conclusion from the extensive experiments is that FGSM families are the most time-efficient. AutoAttack is the second-best time-efficient. Our method is not as time-efficient as FGSM families and AutoAttack but is still more time-efficient than IWA in three datasets. This implicitly reflects the computational cost of the IG method. The FAB attack is efficient in small-size datasets but takes much more time in the large-size ImageNet dataset. OnePixel attack, although, is efficient in four datasets, its SR values in TABLE 5~TABLE 8 are unqualified. CW attack is generally stable and time-consuming in the four datasets. We infer that there is no early stopping mechanism in the CW setup and thus CW must run a fixed number of iterations.

In conclusion, our DI-AA can generate one adversarial example in a relatively efficient time on four datasets and the results are in accord with the time complexity $T(\text{DI-AA})$ in Section 3.4: when the number of input features and/or iterations grows larger, DI-AA requires more time to generate an adversarial example.

TABLE 9. Mean running time of attacks in four datasets on the nonrobust models

Attacks	NSL-KDD	MNIST	CIFAR-10	ImageNet
Ours	0.8	5.17	11.03	123.98 ($T=7$)/ 47.16 ($T=1$)
FGSM	0.003	0.003	0.01	0.02
BIM	0.03	0.03	0.12	0.2
PGD	0.03	0.03	0.11	0.2
OnePix	4.22	4.41	12.95	53.41
CW	46.15	53.22	39.08	67.84
FAB	1.98	3.59	10.67	414.66
IWA	1.46	7.94	7.65	127.69
AutoA	0.58	0.64	1.17	2.58

4.3.4. Analysis of White-box Attacks on Robust Models

This section reports on the results of our method and baselines attacking the TRADES-robust models, and the results are shown in TABLE 10, TABLE 11 and TABLE 12. As before, the best performance results are in bold. Note that if our result is the second-best, it will be in italics. One common result is that all the SR scores are low. This is reasonable since all the robust models are trained by the TRADES defensive method and the AE attacks have difficulty generating AE on the robust models. A relatively high SR score implies that the attack has the ability to evade the defense.

In TABLE 10, we can see that ours and CW achieve 100% SR scores on the NSL-KDD robust model, indicating that both have a strong evasion ability and that TRADES defensive methods do not perform well in the structured dataset. Defensive methods suitable for structured data should be considered for future work. Meanwhile, ours only needs the least perturbation to reach a 100% SR score, which is much better than other baselines.

TABLE 11 lists the results of the MNIST robust model. All the SR scores are lower than 10%, indicating that the TRADES method makes the MNIST model robust. Nevertheless, ours still reaches the best SR score with the second-least perturbation. IWA obtains the second-best SR score, but its perturbation is much higher than ours. Although CW receives the least perturbation, its SR score is low. Other baselines, such as FAB and AutoAttack, have a larger perturbation with an improper SR score.

TABLE 10. Comparison of eight AE white-box attacks to the TRADES-robust model on the NSL-KDD dataset

Attacks	L_0		L_1		L_2		SR
	Mean	Std	Mean	Std	Mean	Std	
Ours	26.11	8.96	*9.42*	**3.44**	**2.15**	**0.55**	**100.00%**
FGSM	109.67	11.02	88.44	21.47	11.59	6.91	15.84%
BIM	105.29	10.71	29.91	16.16	7.06	8.22	25.40%
PGD	104.62	7.30	29.41	14.07	5.76	6.87	60.28%
OnePix	**4.98**	**0.14**	**6.10**	5.26	3.25	4.66	38.81%
CW	121.20	3.85	11.53	10.91	3.45	6.44	**100.00%**
FAB	107.90	12.45	21.14	12.85	4.64	6.28	96.83%
IWA	15.76	9.06	11.32	6.98	3.08	1.47	88.54%
AutoA	117.33	2.43	31.79	11.88	9.47	9.06	9.24%

TABLE 11. Comparison of eight AE white-box attacks to the TRADES-robust model on the MNIST dataset

Attacks	L_0		L_1		L_2		SR
	Mean	Std	Mean	Std	Mean	Std	
Ours	167.02	68.39	27.52	13.99	3.58	1.24	**9.08%**
FGSM	445.46	32.07	121.51	6.53	5.99	**0.16**	1.31%
BIM	483.09	40.41	133.61	11.15	6.29	0.27	1.76%
PGD	483.22	35.02	133.37	10.31	6.28	0.25	1.79%
OnePix	**29.32**	0.86	15.70	**1.43**	3.17	0.23	7.25%
CW	783.79	**0.52**	**4.70**	2.99	**1.02**	0.53	4.26%
FAB	618.06	52.66	94.37	35.58	4.28	1.48	5.69%
IWA	626.02	210.33	130.16	46.69	7.28	2.39	8.64%
AutoA	646.51	31.13	140.49	8.69	6.15	0.28	6.53%

TABLE 12. Comparison of eight AE white-box attacks to the TRADES-robust model on the CIFAR-10 dataset

Attacks	L_0		L_1		L_2		SR
	Mean	Std	Mean	Std	Mean	Std	
Ours	439.17	324.10	86.42	70.03	5.48	2.76	**100.00%**
FGSM	3060.37	44.64	858.40	49.19	15.82	0.59	85.37%
BIM	3057.13	61.81	830.48	60.11	15.43	0.71	98.49%
PGD	3056.44	65.18	829.40	61.36	15.42	0.73	98.53%
OnePix	**15.00**	**0.00**	**7.63**	**1.50**	2.12	0.36	33.21%
CW	3071.88	0.52	19.94	13.42	**0.89**	0.58	99.99%
FAB	3057.96	59.95	120.44	79.99	2.25	1.53	99.99%
IWA	247.02	354.67	78.53	105.31	5.84	5.63	98.52%
AutoA	3060.38	50.06	332.65	20.61	6.20	**0.25**	98.46%

Unexpectedly, TABLE 12 shows that TRADES defensive methods have a limited effect on boosting the robustness of the CIFAR-10 model. From these three subtables, it can be inferred that the TRADES defensive method is not stable and cannot always effectively improve the model robustness on any DL model. Nevertheless, our method achieves the best SR score with the third-least perturbation.

In summary, the TRADES defensive method is not stable and generalized to boost the robustness of the DL model. When the model has boosted the robustness marginally, our method can evade the defense with a 100% success rate. When the model has greatly boosted the robustness, our method can still evade it with the least perturbation and the highest success rate.

4.4. Black-box Attack on Robust Models

To verify the transferability of AE, we transferred CIFAR-10 adversarial examples generated by DI-AA in Section 4.3.2 to attack unknown models with other new unknown defensive methods [40-45]. The results are shown in TABLE 13. We note that we only compare three methods since AutoAttack and CW perform well in the black-box manner. Clearly, our approach and AutoAttack can successfully transfer AE to attack the black-box models. Moreover, our approach can decrease accuracy by approximately 16%~31% on robust models, which is generally larger compared with AutoAttack.

TABLE 13. Accuracy decrease (%) of the three methods in the black box manner

Defensive Approach	Robust Accuracy	ours	AutoA	CW
[40]	88.02	-30.54	-30.41	-
[41]	89.69	-20.01	-16.55	-
[42]	88.51	-19.36	-12.09	-
[43]	92.41	-15.69	-23.83	-
[44]	89.05	-24.37	-13.68	-
[45]	89.36	-23.03	-18.61	-
[21]	84.92	-16.94	-31.84	-6.99

5. Conclusion

In this paper, we have taken a step in investigating how to integrate the interpretable method of deep Taylor decomposition with adversarial example generation algorithms to explore the effectiveness of the interpretable method for adversarial example generation. Extensive experimental results indicate that the saliency map generated by the interpretable method can be the criterion to guide the generation of AE. Moreover, our proposed white-box adversarial example generation approach, DI-AA, can attack the nonrobust and robust models with a high success rate and low perturbation. The perturbation is closer to or lower than that of the previous state-of-the-art methods. In addition, the AE generated by DI-AA can reduce the accuracy of the robust black-box models by 16%~31% in the black-box manner.

In future work, adversarial example generation that can fool the model and the interpretation method while maintaining low perturbation and high transferability will be considered. Furthermore, a more mathematical adversarial example generation method and reality simulation will be carried out.

References

[1] M. Wang, J. Xie, P.W. Grant, S. Xu, PSP-PJMI: An innovative feature representation algorithm for identifying DNA N4-methylcytosine sites, Information Sciences. 606 (2022) 968–983. https://doi.org/10.1016/j.ins.2022.05.060.

[2] S. Li, W. Chen, Y. Zhang, G. Zhao, R. Pan, Z. Huang, Y. Tang, A context-enhanced sentence representation learning method for close domains with topic modeling, Information Sciences. 607 (2022) 186–210. https://doi.org/10.1016/j.ins.2022.05.113.

[3] K. Zhao, D. Ji, F. He, Y. Liu, Y. Ren, Document-level event causality identification via graph inference mechanism, Information Sciences. 561 (2021) 115–129. https://doi.org/10.1016/j.ins.2021.01.078.

[4] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, in: International Conference on Learning Representations, 2014.

[5] I. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in: International Conference on Learning Representations, 2015.

[6] A. Kurakin, I. Goodfellow, S. Bengio, Adversarial examples in the physical world, in: ICLR Workshop, 2017.

[7] N. Carlini, D. Wagner, Towards evaluating the robustness of neural networks, in: 2017 IEEE Symposium on Security and Privacy (SP), IEEE, San Jose, CA, USA, 2017: pp. 39–57. https://doi.org/10.1109/SP.2017.49.

[8] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, A. Vladu, Towards deep learning models resistant to adversarial attacks, in: International Conference on Learning Representations, 2018.

[9] F. Croce, M. Hein, Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks, in: International Conference on Machine Learning, PMLR, 2020: pp. 2206–2216.

[10] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, C.-J. Hsieh, ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models, in: Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security - AISec '17, ACM Press, Dallas, Texas, USA, 2017: pp. 15–26. https://doi.org/10.1145/3128572.3140448.

[11] X. Wei, Y. Guo, B. Li, Black-box adversarial attacks by manipulating image attributes, Information Sciences. 550 (2021) 285–296. https://doi.org/10.1016/j.ins.2020.10.028.

[12] J. Shen, N. Robertson, BBAS: Towards large scale effective ensemble adversarial attacks against deep neural network learning, Information Sciences. 569 (2021) 469–478. https://doi.org/10.1016/j.ins.2020.11.026.

[13] F. Croce, M. Hein, Minimally distorted adversarial examples with a fast adaptive boundary attack, in: International Conference on Machine Learning, PMLR, 2020: pp. 2196–2205.

[14] Y. Dong, F. Liao, T. Pang, H. Su, J. Zhu, X. Hu, J. Li, Boosting adversarial attacks with momentum, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, IEEE, Salt Lake City, UT, 2018: pp. 9185–9193. https://doi.org/10.1109/CVPR.2018.00957.

[15] J. Lin, C. Song, K. He, L. Wang, J.E. Hopcroft, Nesterov accelerated gradient and scale invariance for adversarial attacks, in: International Conference on Learning Representations, 2020.

[16] A. Subramanya, V. Pillai, H. Pirsiavash, Fooling network interpretation in image classification, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019: pp. 2020–2029. https://doi.org/10/ghfhf6.

[17] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-CAM: Visual explanations from deep networks via gradient-based localization, in: 2017 IEEE International Conference on Computer Vision (ICCV), 2017: pp. 618–626. https://doi.org/10/gfkqbw.

[18] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, K.-R. Müller, Explaining nonlinear classification decisions with deep Taylor decomposition, Pattern Recognition. 65 (2017) 211–222. https://doi.org/10.1016/j.patcog.2016.11.008.

[19] Y. Wang, J. Liu, X. Chang, J. Mišić, V.B. Mišić, IWA: Integrated gradient-based white-box attacks for fooling deep neural networks, International Journal of Intelligent Systems. (2021). https://doi.org/10.1002/int.22720.

[20] J. Su, D.V. Vargas, S. Kouichi, One pixel attack for fooling deep neural networks, IEEE Trans. Evol. Computat. 23 (2019) 828–841. https://doi.org/10.1109/TEVC.2019.2890858.

[21] H. Zhang, Y. Yu, J. Jiao, E.P. Xing, L.E. Ghaoui, M.I. Jordan, Theoretically principled trade-off between robustness and accuracy, in: Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA, PMLR, 2019: pp. 7472–7482.

[22] H. Yin, H. Zhang, J. Wang, R. Dou, Boosting adversarial attacks on neural networks with better optimizer, Security and Communication Networks. 2021. https://doi.org/10.1155/2021/9983309.

[23] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, H. Lipson, Understanding neural networks through deep visualization, in: International Conference on Learning Representations, Lille, France, 2015.

[24] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: Proceedings of the 34th International Conference on Machine Learning - Volume 70, JMLR.org, Sydney, NSW, Australia, 2017: pp. 3319–3328.

[25] J. Kauffmann, K.-R. Müller, G. Montavon, Towards explaining anomalies: A deep Taylor decomposition of one-class models, Pattern Recognition. 101 (2020) 107198. https://doi.org/10.1016/j.patcog.2020.107198.

[26] A. Boopathy, S. Liu, G. Zhang, C. Liu, P.-Y. Chen, S. Chang, L. Daniel, Proper network interpretability helps adversarial robustness in classification, in: International Conference on Machine Learning, PMLR, 2020: pp. 1014–1023.

[27] H. Yang, J. Zhang, H. Dong, N. Inkawhich, A. Gardner, A. Touchet, W. Wilkes, H. Berry, H. Li, DVERGE: Diversifying vulnerabilities for enhanced tobust generation of ensembles, in: Advances in Neural Information Processing Systems, Curran Associates, Inc., 2020: pp. 5505–5515.
[28] M. Andriushchenko, N. Flammarion, Understanding and improving fast adversarial training, in: Advances in Neural Information Processing Systems, Curran Associates, Inc., 2020: pp. 16048–16059.
[29] D. Tsipras, S. Santurkar, L. Engstrom, A. Turner, A. Madry, Robustness may be at odds with accuracy, in: International Conference on Learning Representations, New Orleans, LA, USA, 2019.
[30] H. Wang, Y. Wang, Z. Zhou, X. Ji, D. Gong, J. Zhou, Z. Li, W. Liu, CosFace: Large margin cosine loss for deep face recognition, in: 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018: pp. 5265–5274. https://doi.org/10.1109/cvpr.2018.00552.
[31] D.P. Kingma, J. Ba, Adam: a method for stochastic optimization, in: ArXiv:1412.6980 [Cs], San Diego, California, USA, 2015. http://arxiv.org/abs/1412.6980.
[32] J. Zhuang, T. Tang, Y. Ding, S.C. Tatikonda, N. Dvornek, X. Papademetris, J. Duncan, AdaBelief Optimizer: Adapting stepsizes by the belief in observed gradients, Advances in Neural Information Processing Systems. 33 (2020) 18795–18806.
[33] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, Gradient-based learning applied to document recognition, Proc. IEEE. 86 (1998) 2278–2324. https://doi.org/10.1109/5.726791.
[34] K. Alex, Learning multiple layers of features from tiny images, University of Toronto, Toronto, 2009.
[35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, ImageNet: A large-scale hierarchical image database, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, 2009: pp. 248–255. https://doi.org/10/cvc7xp.
[36] M. Tavallaee, E. Bagheri, W. Lu, A.A. Ghorbani, A detailed analysis of the KDD CUP 99 data set, in: 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, IEEE, Ottawa, ON, Canada, 2009: pp. 1–6. https://doi.org/10.1109/CISDA.2009.5356528.
[37] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016: pp. 770–778. https://doi.org/10.1109/cvpr.2016.90.
[38] G.W. Ding, L. Wang, X. Jin, Advertorch v0.1: An adversarial robustness toolbox based on PyTorch, ArXiv:1902.07623 [Cs, Stat]. (2019). http://arxiv.org/abs/1902.07623.
[39] H. Kim, Torchattacks: A PyTorch repository for adversarial attacks, ArXiv:2010.01950 [Cs]. (2021). http://arxiv.org/abs/2010.01950.
[40] G.W. Ding, Y. Sharma, K.Y.C. Lui, R. Huang, MMA training: Direct input space margin maximization through adversarial training, in: International Conference on Learning Representations, 2019.
[41] Y. Carmon, A. Raghunathan, L. Schmidt, J.C. Duchi, P.S. Liang, Unlabeled data improves adversarial robustness, in: Advances in Neural Information Processing Systems, Curran Associates, Inc., 2019.
[42] S. Gowal, C. Qin, J. Uesato, T. Mann, P. Kohli, Uncovering the limits of adversarial training against norm-bounded adversarial examples, ArXiv:2010.03593 [Cs, Stat]. (2021). http://arxiv.org/abs/2010.03593.
[43] M. Augustin, A. Meinke, M. Hein, Adversarial robustness on in- and out-distribution improves explainability, in: Computer Vision – ECCV 2020, Springer International Publishing, Cham, 2020: pp. 228–245.
[44] S.-A. Rebuffi, S. Gowal, D.A. Calian, F. Stimberg, O. Wiles, T. Mann, Fixing data augmentation to improve adversarial robustness, ArXiv:2103.01946 [Cs]. (2021). http://arxiv.org/abs/2103.01946.
[45] J. Zhang, J. Zhu, G. Niu, B. Han, M. Sugiyama, M. Kankanhalli, Geometry-aware Instance-reweighted adversarial training, in: International Conference on Learning Representations, 2020.
[46] X. Zhang, N. Wang, H. Shen, S. Ji, X. Luo, T. Wang, Interpretable deep learning under fire, in 29th USENIX Security Symposium (USENIX Security 20), 2020, pp. 1659–1676.
[47] A. Aldahdooh, W. Hamidouche, S.A. Fezza, O. Déforges, Adversarial example detection for DNN models: a review and experimental comparison, Artif Intell Rev. (2022). https://doi.org/10.1007/s10462-021-10125-w.
[48] G.S. Dhillon, K. Azizzadenesheli, Z.C. Lipton, J.D. Bernstein, J. Kossaifi, A. Khanna, A. Anandkumar, Stochastic activation pruning for robust adversarial defense, in: International Conference on Learning Representations, 2018.
[49] C. Guo, M. Rana, M. Cisse, L. van der Maaten, Countering adversarial images using input transformations, in: International Conference on Learning Representations. 2018.
[50] A. Athalye, L. Engstrom, A. Ilyas, K. Kwok, Synthesizing robust adversarial examples, in: Proceedings of the 35th International Conference on Machine Learning, PMLR, 2018: pp. 284–293.