IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 6 June 2024, accepted 9 August 2024, date of publication 21 August 2024, date of current version 2 September 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3446834

RESEARCH ARTICLE

Uncovering Distortion Differences: A Study of Adversarial Attacks and Machine Discriminability

XIAWEI WANG[1], YAO LI[2], CHO-JUI HSIEH[3], AND THOMAS C. M. LEE[4], (Senior Member, IEEE)

[1]Graduate Group in BioStatistics, University of California, Davis, Davis, CA 95616, USA
[2]Department of Statistics and Operation Research, The University of North Carolina at Chapel Hill, Chapel Hill, NC 27599, USA
[3]Department of Computer Science, University of California at Los Angeles, Los Angeles, CA 90095, USA
[4]Department of Statistics, University of California at Davis, Davis, CA 95616, USA

Corresponding author: Thomas C. M. Lee (tcmlee@ucdavis.edu)

This work was supported in part by a Cisco Faculty Award and the National Science Foundation under Grant 1934568, Grant 2008173, Grant 2048280, Grant 2113605, Grant 2134107, Grant 2152289, Grant 2210388, and Grant 2331966.

ABSTRACT Deep neural networks have performed remarkably in many areas, including image-related classification tasks. However, various studies have shown that they are vulnerable to adversarial examples – images carefully crafted to fool well-trained deep neural networks by introducing imperceptible perturbations to the original images. To better understand the inherent characteristics of adversarial attacks, this paper studies the features of three common attack families: gradient-based, score-based, and decision-based. The primary objective is to recognize distinct types of adversarial examples, as identifying the type of information possessed by the attacker can aid in developing effective defense strategies. This paper demonstrates that adversarial images from different attack families can be successfully identified with a simple model. To further investigate the reason behind the observations, this paper conducts carefully designed experiments to study the distortion patterns of different attacks. Experimental results on CIFAR10 and Tiny ImageNet validated the differences in distortion patterns between various attack types for both L_2 and L_∞ norm.

INDEX TERMS Decision-based attacks, deep neural networks, gradient-based attacks, image classification, score-based attacks.

I. INTRODUCTION

Well-trained deep neural networks are capable of achieving outstanding performance in many areas, including image-related classification tasks [1], [2], [3]. However, various studies have shown that they may not be fully reliable and can be fooled by adversarial examples – images that are carefully crafted to fool such deep neural networks by introducing imperceptible perturbation to the original images [4], [5], [6], [7]. This raises serious security concerns for the AI community. Many works have been done to study and defend against adversarial attacks [8], [9], [10], [11]. In particular, adversarial detection methods have been proposed to determine whether an input image is an adversarial example or not [12], [13], [14], [15], [16], [17]. Moreover, it is helpful for the defender if reverse engineering can be done to reveal more information about the attacks based on the detected adversarial examples. For example, there are three main attack families to perform attacks: gradient-based, score-based, and decision-based, which rely on the gradient, predicted score, and predicted label of the victim model, respectively. Based on the detected adversarial examples, if the defender can tell what type of attack is used, the defender will know what information has been leaked to the attacker. Consequently, the defender can modify the model accordingly to prevent further attacks. Some works have been done to study the reverse engineering of adversarial attacks: Pang et al. [18] proposed

The associate editor coordinating the review of this manuscript and approving it for publication was Jeon Gwanggil.

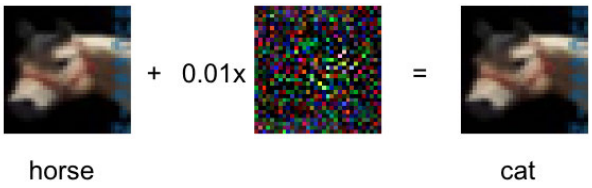

FIGURE 1. An adversarial example generated by Boundary attack: introducing adversarial perturbations to the horse image causes a classifier to label it as a cat.

the query of interest (QOI) estimation model to infer the adversary's target class by model queries in black-box settings. Goebel et al. [19] estimated adversarial setup from image sample for gradient-based attacks FGSM [5] and PGD [20]. Gong et al. [21] proposed a general formulation of the reverse engineering of deceptions problem that can estimate adversarial perturbations and provide the feasibility of inferring the intention of an adversary.

In this paper, we first demonstrated that, given an adversarial example, the corresponding attack family can be accurately identified with a simple model. Once we had established this, we turned our attention to analyzing the specific features of each type of attack to understand the underlying differences between them better. Section II covers preliminary information presented in the paper. Section III focuses on our image classifier that accurately identifies attack families (gradient-based, score-based, or decision-based). In Section IV, we provide an extensive analysis of the features associated with each type of attack.

II. PRELIMINARIES

A. NOTATIONS

We consider an image classifier $f(\cdot)$ as the victim model of adversarial attacks. The input to the classifier is $\boldsymbol{x_0} \in [0,1]^{w,h,c}$, a c-channel image sample with width w and height h. The true label associated with $\boldsymbol{x_0}$ is denoted as y, and the adversarial example generated from $\boldsymbol{x_0}$ is denoted as $\boldsymbol{x}^*$. We denote $f(\boldsymbol{x_0})$ as the predicted score vector and $c(\boldsymbol{x_0}) = \arg\max_i f(\boldsymbol{x_0})$ as the predicted label, indicating the i^{th} label has the highest prediction score.

B. ADVERSARIAL EXAMPLES

An adversarial example $\boldsymbol{x}^*$ and the original image $\boldsymbol{x_0}$ are visually indistinguishable, but their predicted labels are different. That is, $\mathcal{D}(\boldsymbol{x_0}, \boldsymbol{x}^*)$ is very small in some distance metric $\mathcal{D}$, while $c(\boldsymbol{x}^*) \neq c(\boldsymbol{x_0})$. Taking Fig.1 as an example, humans will recognize that the two images are of the same horse. However, the image on the right is generated by adding imperceptible perturbations to the original image on the left, which causes a particular classifier to classify it as a cat. Existing methods use L_p metrics to evaluate the distance between adversarial and original samples. This paper focuses on L_2 and L_∞, the most commonly used metrics in adversarial attacks.

C. DATA SETS AND VICTIM MODELS

We use CIFAR10 [22] image data set with ten different classes of resolution 32×32. Another data set we use is Tiny Imagenet [23], which has 200 classes, and the resolution of the images is 64×64. For CIFAR-10, the victim model is VGG-16 with batch normalization [1], of which accuracy is 93.34%. For Tiny ImageNet, the victim model architecture is ResNet18 [3] with 68.64% accuracy.

D. ADVERSARIAL ATTACKS

Different attack methods can be classified into two categories according to their goals: untargeted and targeted. Untargeted attacks are successful as long as the adversarial example is misclassified. Targeted attacks, instead, are successful only when the adversarial example is classified into a target class. Take Fig.1 as an example; the untargeted attack is successful if the right-side image is not classified as a horse, while the targeted attack is successful only when it is predicted as a cat if the target class is a cat. In this paper, all experiments are based on untargeted attacks.

Depending on the information required, existing attack methods can be divided into three categories: gradient-based, score-based, and decision-based. The gradient-based attack is also known as a white-box attack, in which all information of the victim model is revealed to the attacker so that the attackers can calculate gradients. Popular gradient-based attacks are FGSM [5], PGD [20] and C&W [6]. If an attacker only has access to the predicted score of the victim model, it is a score-based attack, also known as a soft-label black-box setting. Popular score-based attacks include ZOO [24], NES [25] and Square [26]. In practical scenarios, the attacker only has access to the predicted labels of the model. Attacks under this setting are called decision-based attacks. Examples of such attacks include those described in [27] and [28], as well as popular methods like Boundary [29], Sign-OPT [30] and HopSkipJump (HSJ) [31]. Table 1 lists six representative attacks under different settings in L_2 or L_∞ metrics. In this paper, we conduct attack family classification with these attacks and study their perturbation patterns. Adversarial images are generated based on ART package [32].

TABLE 1. Representative attacks of different metrics from different families under L_2 and L_∞.

	L_2	L_∞
gradient-based	C&W	PGD
score-based	ZOO	Square
decision-based	Boundary	HopSkipJump

E. PERTURBATION VISUALIZATION

Perturbations are the differences between the adversarial example and the corresponding original image, showing how the original image is modified. Since perturbations are imperceptible, we amplify the perturbation by 100 times for visualization purposes in this paper.

III. REVERSE ENGINEERING OF ADVERSARIAL ATTACKS

Most current reverse engineering methods focus on analyzing specific attack methods. However, this section explores the potential for identifying attack families associated with adversarial examples. Successful detection of attack families (gradient-based, score-based, or decision-based) can be a useful tool for defenders, as it allows them to understand better the level of information that has been leaked during attacks so that defenders can properly assess the potential impact of that attack family.

When an adversarial attack is launched, it exploits weaknesses in the model: gradient-based attacks take advantage of the model gradients; score-based attacks rely on the predicted scores of the model; and decision-based attacks rely on the predicted labels. This knowledge can help develop an effective response to the attack. Overall, by identifying the specific attack families and taking targeted actions to address the vulnerability exploited by the attack, defenders can improve model resilience and minimize the damage caused by attacks.

A. EXPERIMENTS: CLASSIFYING ATTACK FAMILIES

We generate adversarial examples of each attack family and two metrics (L_2 and L_∞) using attacks in Table 1 with data sets and victim models mentioned in Section II.

For the L_2 attacks, the perturbation upper bounds are 1.00 and 5.00 on CIFAR10 and Tiny ImageNet, respectively. The perturbation upper bound is 0.03 for different L_∞ attacks on both CIFAR10 and Tiny ImageNet.

With the generated adversarial examples, we perform the following experiments: (1) classifying attack families in L_2 metric; (2) classifying attack families in L_∞ metric; and (3) classifying attack families with adversarial examples of both L_2 and L_∞ metrics. A classifier with VGG16 architecture is trained for multi-class classification to identify the attack family based on adversarial examples. The same architecture is used for both CIFAR10 and Tiny ImageNet in all the following experiments except in Experiment D, where the task is six-class classification, and the last layer has six neurons instead of three.

1) EXPERIMENT A

For L_2-norm based attacks, we choose C&W (gradient-based), ZOO (score-based), and Boundary (decision-based) as representatives of each attack family. If all three attacks can successfully fool the victim model by modifying the same original image under the perturbation bound, we keep the corresponding adversarial examples and split them into training and test sets for the attack family classification task. These adversarial examples are called successful adversarial examples across three attacks.

2) EXPERIMENT B

For L_∞-norm based attacks, we choose PGD (gradient-based), Square (score-based), and HopSkipJump (decision-based) as representative attacks. A similar procedure is applied as in Experiment A to obtain the training and test sets for the attack family classification task.

3) EXPERIMENT C

Adversarial examples in Experiments A and B are merged into three classes so that each class contains adversarial examples generated by attacks from the same attack family but different norm metrics. Similarly, we only keep successful adversarial examples across six attacks. Gradient-based class includes adversarial examples generated by C&W(L_2) and PGD(L_∞). Score-based class includes ZOO(L_2) and Square(L_∞). Decision-based class includes Boundary(L_2) and HopSkipJump(L_∞). The classification task is to do a three-class classification, identifying the attack family given an adversarial example.

4) EXPERIMENT D

To investigate if there are not just differences between attack families but also differences between attack methods, this experiment uses the same data as in Experiment C but performs six-class classification to identify specific attacks, not attack families.

TABLE 2. Accuracy of attack family classification task (Experiments A, B, C) and attack method classification task (Experiment D) on CIFAR10 and Tiny ImageNet without original images.

	CIFAR10	Tiny ImageNet
Experiment A	82.74%	81.08%
Experiment B	95.51%	96.96%
Experiment C	85.58%	85.77%
Experiment D	76.30%	73.84%

The first three rows (Experiments A, B, C) in Table 2 show the attack family classification accuracies on CIFAR10 and Tiny ImageNet datasets. The last row (Experiment D) shows the attack method classification accuracy. The first three experiments achieve high accuracies on different datasets, which suggests that attack families modify the image in different ways and machines can learn the pattern based on adversarial examples, although adversarial examples are indistinguishable from the original images to humans. The testing accuracies are not bad for Experiment D, which implies that attacks of the same family also have different patterns.

In many real-world scenarios, whether the input has been perturbed or not is often unknown to the models. We incorporate non-perturbed original images into the classification task to address this concern. The outcomes of the experiment can be found in Table 3. The experimental setup remains consistent, with the only variation being the inclusion of original images as a distinct category in the input. Except for Experiment A, all experiments stay at a high accuracy level. Experiment A experiences a decrease in accuracy due to its utilization of the L_2 norm attack, which considers the cumulative perturbations across all pixels, leading to smaller discrepancy to original images when a

certain threshold of the cumulative sum is applied. On the other hand, the L_∞ norm attack focuses on the maximum perturbed pixel while allowing other pixels to be perturbed as long as their individual perturbations are below the threshold, leading to more noticeable perturbation patterns.

TABLE 3. Accuracy of attack family classification task (Experiments A, B, C) on CIFAR10 and Tiny ImageNet with original images.

	CIFAR10	Tiny ImageNet
Experiment A	74.84%	65.45%
Experiment B	92.15%	91.87%
Experiment C	80.65%	89.36%

B. ROBUSTNESS OF ATTACK FAMILY CLASSIFICATION

This section presents evidence for the robustness of attack family identification, even when they have varying perturbation levels or involve ensemble attacks.

1) WITH VARIOUS NORM LIMITS

In this section, we demonstrate that attack family types of adversarial examples can be accurately identified despite having different perturbation levels. The CIFAR10 dataset was used for Experiment A and Experiment B to investigate the effect of different limits on the attack family classification under L_2 and L_∞ norms. Experiment A of classifying L_2 attacks from three different attack families achieved high levels of accuracy across a range of limit values, including 1.0, 0.8, and 0.6. Similarly, in Experiment B of classifying L_∞ attacks from three different attack families, high levels of accuracy were achieved across a range of limit values including 0.03, 0.02, and 0.01, see Table 4. However, we observed a decrease in accuracy as L_2 or L_∞ norm limit becomes smaller, which can be attributed to the limited number of successful adversarial samples across three attacks under smaller limits.

2) WITH ENSEMBLE ATTACK

Auto attack is an ensemble attack algorithm that includes four attacks: APGD-CE, APGD-DLR, FAB [33], and Square Attack, where APGD-CE and APGD-CE are two extensions of the PGD attack overcoming failures due to suboptimal step size and problems of the objective function [34]. This algorithm iterates over the list of attacks until an adversarial example is successfully generated. Though both gradient and score information are involved, we consider auto attack as a gradient-based attack for the purpose of the attack family classification task. In our evaluation, we classify adversarial examples of CIFAR10 generated by Auto-attack(gradient-based), ZOO(Score-based), and Boundary(decision-based) under L_2 norms and achieved an accuracy of 83.40%; under L_∞ norms, we evaluated Auto-attack(gradient-based), Square(Score-based), HopSkipJump(decision-based), and accuracy achieved 97.40%. These results demonstrated that different attack families could be effectively classified even when the gradient-based attack involves more than just gradient information. Besides, the accuracy of the attack family classification remains consistent regardless of the specific attacks involved.

TABLE 4. Accuracy of attack family classification for various L_2 and L_∞ limits.

L_2 Norm Limit	1.0	0.8	0.6
Accuracy	82.74%	81.30%	76.30%

L_∞ Norm Limit	0.03	0.02	0.01
Accuracy	95.51%	92.63%	81.57%

IV. EXPLORING CHARACTERISTICS OF ATTACK FAMILIES

Although adversarial examples from different attack families appear to be indistinguishable, machines can learn and classify them with some subtle signatures. One question arises: What patterns does the classification model acquire to recognize the attack family and attack method? Since the differences in adversarial attacks are embedded in the perturbations, we propose to investigate the reasons behind the ease of identifying attack families by analyzing the perturbation patterns exhibited in various attacks. Visualization examples for representative L_2 attacks and L_∞ attacks are displayed in Fig. 2 and Fig. 3. More examples are in the Appendix.

A. L_2 ATTACKS

Different L_2 attacks modify the original images in different ways, resulting in different perturbation patterns; see Fig. 2, each subfigure lists adversarial examples from C&W, ZOO, and Boundary and corresponding amplified perturbations from left to right. It is obvious that the perturbations of the three attacks are different. The perturbations of the C&W attack seem to focus on the location of the object. ZOO introduces large perturbations for some pixels. The perturbations of the Boundary attack are relatively smaller and all over the place. In the following sections, we study the characteristics of C&W, ZOO, and Boundary and discuss why they generate perturbations of different patterns.

1) C&W ATTACK

C&W attack is one of the strongest gradient-based attacks to date. It can perform targeted and untargeted attacks with L_2 or L_∞ metric. Although L_∞ norm is feasible, L_2 norm is widely used in C&W attacks and can be formulated as the following regularized optimization problem:

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}\in[0,1]^n}{\operatorname{argmin}}\{\|\boldsymbol{x}-\boldsymbol{x_0}\|_2^2 + cg(\boldsymbol{x})\}. \tag{1}$$

The first term $\|\boldsymbol{x}-\boldsymbol{x_0}\|_2^2$ enforces a slight distortion to the original input $\boldsymbol{x_0}$ and the second term $g(\boldsymbol{x})$ is a loss function that measures how successful the attack is. The parameter $c > 0$ controls the trade-off between distortion and attack success.

Compared to the other two attacks, it seems that the perturbations of C&W concentrate on the object, see Fig. 2.

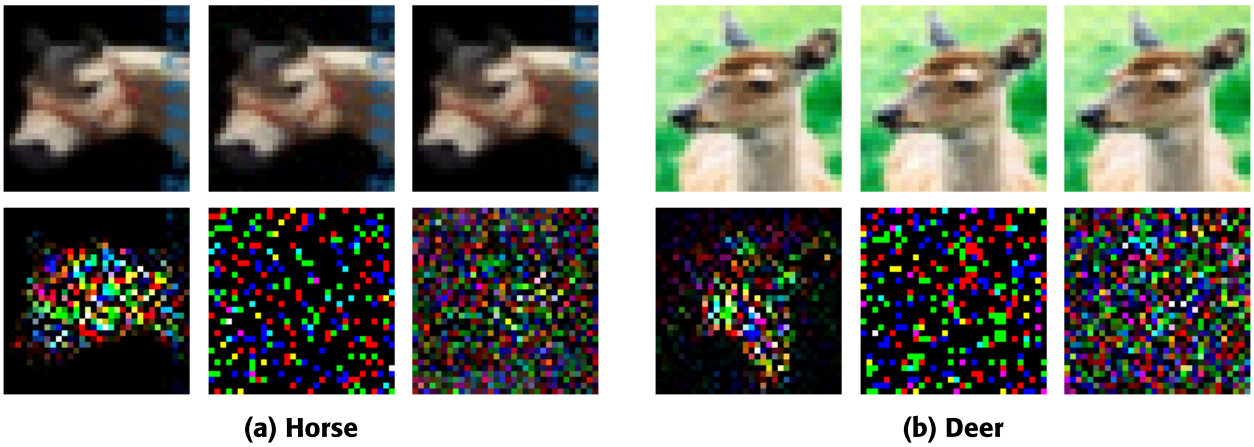

FIGURE 2. Visualization examples for C&W, ZOO, and Boundary are displayed in each subfigure, sampled from CIFAR10. From left to right, the first row shows the adversarial image generated by C&W, ZOO, and Boundary, and the second row shows corresponding amplified perturbations. Though adversarial examples are indistinguishable, perturbations show different patterns: C&W's perturbations focus on the main object; ZOO introduces scattered bright per-pixel perturbations; Boundary's perturbations are more uniform across the image.

FIGURE 3. Visualization examples for PGD, Square, and HopSkipJump are displayed in each subfigure, sampled from the CIFAR10 data set. From left to right, the first row shows the adversarial image generated by PGD, Square, and HopSkipJump, and the second row shows corresponding amplified perturbations. PGD and HSJ have cluttered perturbation patterns, but HSJ is darker due to smaller perturbations. Square's perturbations consist of vertical strips covered by square-shaped regions, though vertical strips may not be obvious since too many squares cover them.

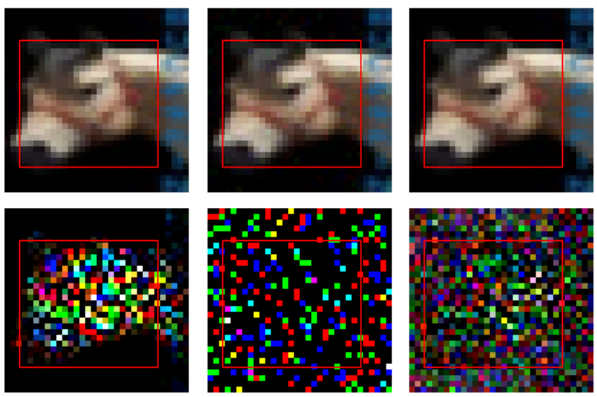

FIGURE 4. The proportion of perturbations inside the bounding box for C&W, ZOO, and Boundary are 96.40%, 69.25%, and 79.51% respectively, from left to right.

To verify if this observation is true, we draw a bounding box of the horse in Fig. 2 and compute the proportion of L_2 perturbations inside the box for all three attacks, see Fig. 4: the proportion of perturbation inside the bounding box for C&W is 96.40%, while for ZOO and Boundary, the proportions are 69.25% and 79.51% respectively.

To verify if this pattern is true for most cases, we randomly sample five images with success across three attacks from each class of CIFAR10 and draw bounding boxes for all 50 images per attack to calculate the proportions of perturbations inside bounding boxes. The proportion is calculated per sampled image for each attack. Fig. 5 shows the histograms of in-box perturbation proportion for each attack. It is evident that C&W has the most left-skewed distribution, indicating that C&W focuses on perturbing the main object in the image.

Two reasons might explain why C&W attacks the object: 1) C&W has access to the true gradients; 2) C&W method starts attacking from the original image. Gradients w.r.t. the input indicates the important areas in the input image and usually concentrate on the objects because the victim model is

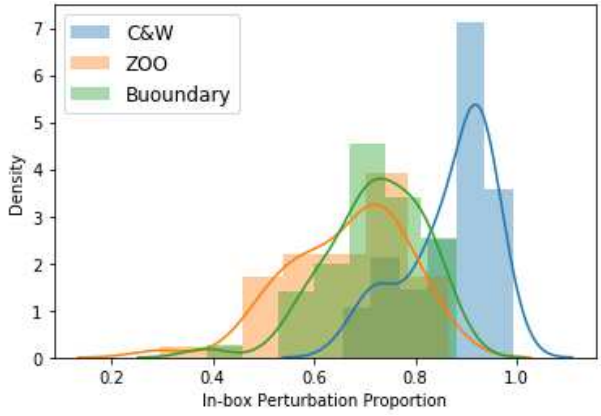

FIGURE 5. In-box perturbation proportion histograms for C&W, ZOO, and Boundary. C&W's distribution is most left-skewed, indicating C&W focuses on attacking the main object.

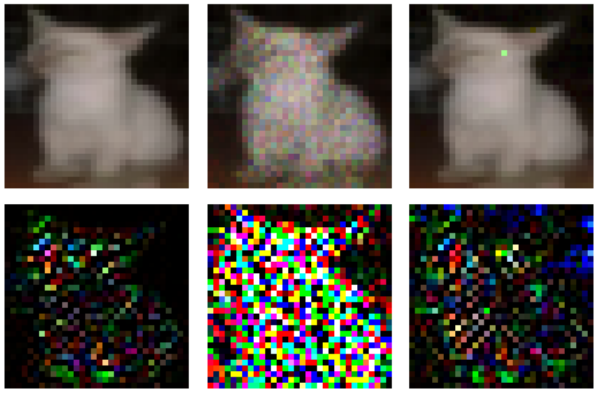

FIGURE 6. From left to right, a cat image is attacked by C&W, estimated-gradient C&W, and random-start C&W. Even though the perturbations of estimated-gradient C&W and random-start C&W also roughly focus on the object area, it is not as obvious as in the perturbations of the original C&W.

trained to do object classification. Therefore, it is expected to see C&W focus on modifying the object. Besides, the initial point of the optimization process is the original image, which excludes the possibility of unnecessary perturbations outside the object area.

To support the above hypothesis, we compare C&W with its two variants: estimated-gradient C&W and random-start C&W. Instead of using true gradients, estimated-gradient C&W uses gradients estimated by Natural Evolution Strategy [35], which was also used by Ilyas et al. [25] to do score-based attack. Random-start C&W starts the attack process with a random adversarial point instead of the original image. The random adversarial point is a random noise image that is not classified into the class of the original image. The point is already misclassified but not close to the original image.

We generate adversarial images with the original C&W and its two variants, then train a VGG16-based model to classify the three types of adversarial images. The classification accuracy reaches 96.03%, indicating that the three types of

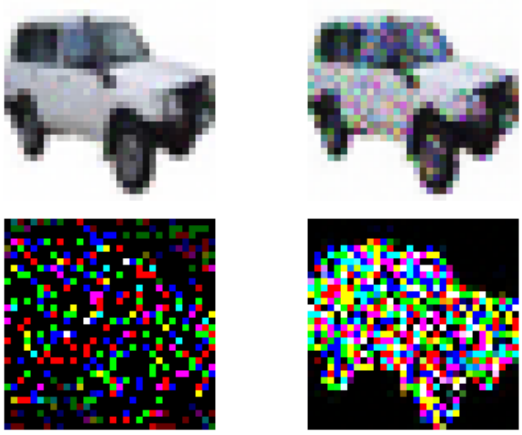

FIGURE 7. An automobile image is attacked by ZOO(left) and estimated-gradient C&W(right). The first row contains adversarial examples, and the second row contains amplified perturbations. ZOO's amplified perturbations are more spread due to coordinate descent.

adversarial attacks are significantly different. Therefore, both gradients and random start affect the patterns of the C&W perturbations.

Fig. 6 lists the adversarial examples and perturbations of C&W, estimated-gradient C&W, and random-start C&W from left to right. The perturbations of estimated-gradient C&W still roughly focus on the object area but are less accurate than those of the original C&W. Also, the overall perturbations are larger: with estimated gradients, it cannot converge to the same level as C&W, resulting in a larger distortion level. With a random adversarial start, C&W gets noisier in the background, even though many perturbations are in the object area. In conclusion, C&W's perturbations focusing on the object area come from two factors: starting from original images and accurate gradients. See more examples in Appendix A.

2) ZOO ATTACK

Zeroth Order Optimization Based Attack (ZOO) uses the finite difference method to approximate the gradients of the loss with respect to the input. The objective function is the same as that of C&W attack but using coordinate descent with estimated gradient:

$$\frac{\partial f}{\partial \boldsymbol{x_i}} \approx \frac{f(\boldsymbol{x} + h\boldsymbol{e_i}) - f(\boldsymbol{x} - h\boldsymbol{e_i})}{2h}, \tag{2}$$

where h is a small constant, $\boldsymbol{e_i}$ is a standard basis vector with a single nonzero entry with value 1 as the i-th element, and i ranges from 1 to the input dimension. That is, ZOO is another variant of C&W but with estimated gradient and coordinate descent.

From Fig. 2, we can see that ZOO's perturbations are made of a few bright pixels, which is expected as it uses coordinate descent to optimize each coordinate iteratively. Unlike gradient descent, that updates all coordinates at once, coordinate descent updates the coordinates by mini-batch. The nature of coordinate descent can lead to ZOO's

perturbation pattern. To show the effect of coordinate descent on perturbation patterns, we compare ZOO with the estimated gradient C&W. The difference is the optimization method: ZOO uses coordinate descent while estimated-gradient C&W uses gradient descent, but both methods need to estimate the gradient. A VGG16-based binary classifier achieves 97.62% accuracy in classifying the adversarial examples generated by the two methods, implying that different optimization methods will result in different perturbation patterns. Fig. 7 shows the adversarial examples and amplified perturbations of ZOO and estimated-gradient C&W. More examples are available in Appendix B. Compared to the estimated-gradient C&W, ZOO has more spread perturbations because of the optimization method. In Appendix IV-A1, we verified that the estimated gradient makes the perturbations larger and less accurate by comparing estimated-gradient C&W with the original C&W. This also helps explain why the perturbations of ZOO are so prominent and scattered. Therefore, coordinate descent and the estimated gradient together lead to ZOO's prominent scattered pixel-level perturbation pattern.

3) BOUNDARY ATTACK

Boundary attack starts with a random adversarial point from a different class, then seeks to minimize the perturbations by randomly walking on the boundary of two classes while remaining adversarial. Compared to C&W, the Boundary attack does not start from the original image and has no access to the gradient information. From Fig. 2, we noticed that the Boundary attack's perturbations distribute over the entire image compared to C&W and ZOO. In fact, we verified in Section IV-A1 that starting from an adversarial point instead of the original image will spread the perturbations, and the gradient information is the key to an accurate attack on the object. This explanation applies to the perturbation patterns of Boundary attacks as well. Fig. 8 shows adversarial examples and perturbations of C&W, random-start C&W, and Boundary. Compared to C&W, the other two attacks show noisy and spread perturbations, even though random-start C&W has most perturbations focused on the frog area. More examples are available in Appendix C.

Besides, unlike random-start C&W, Boundary's updating procedure relies on a random walk instead of gradients, which draws random perturbation from a proposal distribution at each iteration. Hence, Boundary's perturbations are more blurry than the random-start C&W. A VGG16-based three-class model achieves 88.12% accuracy in classifying the three attacks, indicating that the differences are obvious and easy to detect. Therefore, both random adversarial start and lack of gradient information contribute to Boundary's specific perturbation patterns.

B. L_∞ ATTACKS

L_∞ attacks in different attack families show different perturbation patterns as well. In this section, we study the L_∞-norm version of PGD (gradient-based), Square (score-based), and HopSkipJump (decision-based). In our experiments, perturbations are bounded by 0.03. Fig. 3 shows adversarial examples and perturbation patterns of PGD, Square, and HopSkipJump (HSJ). The perturbations of Square consist of vertical strips covered by square-shaped regions. Both PGD and HSJ have clutter perturbation patterns, but the perturbations of HSJ are darker. In the following sections, we discuss the characteristics of Square first and then compare PGD and HSJ.

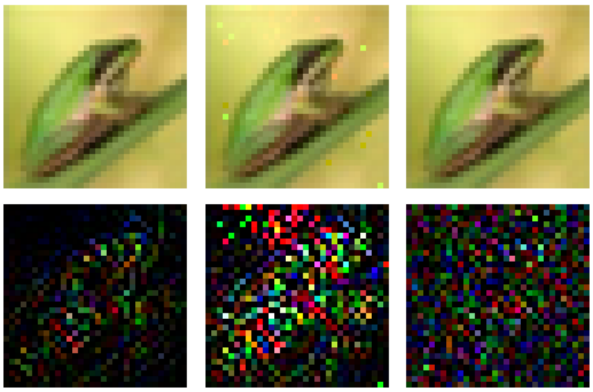

FIGURE 8. A frog image is attacked by C&W, random-start C&W and Boundary in turn. From left to right, the perturbations are getting noisier, and the frog outline is blurring. It indicates both random start and random walk iteration without gradient information contribute to Boundary's noisy perturbations.

1) SQUARE ATTACK

The Square attack is score-based, but unlike other score-based attacks, such as ZOO or NES, it does not estimate the gradients when generating adversarial examples. Instead, it adopts an iterative randomized search scheme: at each iteration, a local square update is chosen at random locations and projected to the input space, then this update is added to the current iteration if the objective function improves. This explains the square-shaped regions in the perturbation pattern. As for initialization, Square uses vertical stripes of width 1, where the color of each stripe is randomly and uniformly sampled. In some cases, it takes many iterations to generate a successful adversarial example, so the stripes are nearly covered by squares.

2) PGD AND HOPSKIPJUMP ATTACK

Projected-Gradient Descent Attack (PGD) crafts adversarial examples by solving the constraint optimization problem iteratively with projected gradient descent, widely used with L_∞ norm. It can be formulated as

$$\boldsymbol{x}^* = \underset{\|\boldsymbol{x}-\boldsymbol{x}^*\|_\infty<\epsilon}{\operatorname{argmax}} L(\theta, \boldsymbol{x}, y), \tag{3}$$

where L is the loss function used to train the victim model, θ is a fixed model parameter, and $(\boldsymbol{x}, y)$ is the input pair of the original image $\boldsymbol{x}$ and the corresponding label y. It uses a multi-step iteration scheme: at each iteration, take a small

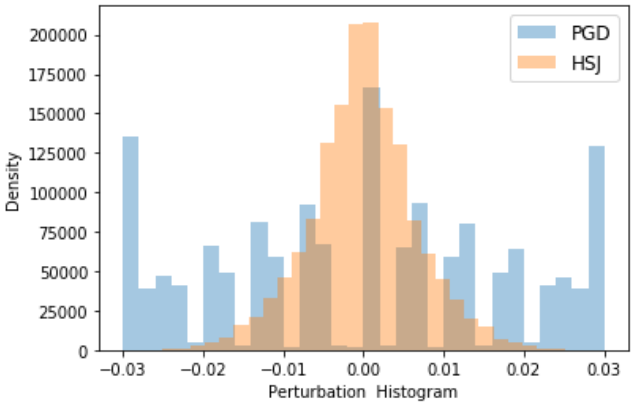

FIGURE 9. Histogram of perturbation values of PGD and HSJ. PGD has a bar-plot-like perturbation distribution because it uses a fixed step size to update, while HSJ has a normal-like perturbation distribution.

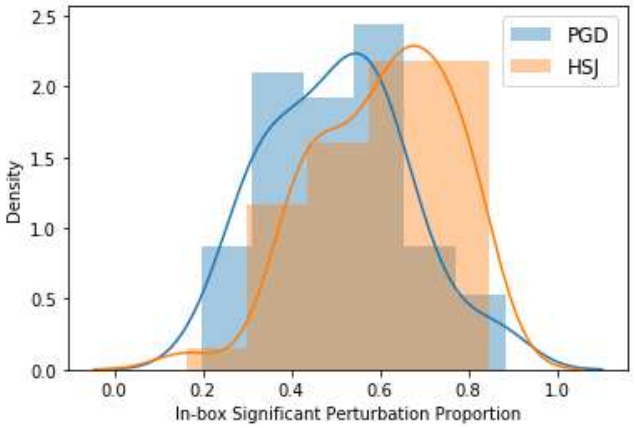

FIGURE 10. Histogram of In-box significant perturbation proportion of PGD and HSJ. HSJ's distribution is more left-skewed than PGD, indicating it has more significant perturbations in the object area.

step α according to the sign of the gradient and clip the result to the ϵ-ball of the original input:

$$\boldsymbol{x}^{t+1} = \Pi_{\epsilon}\{\boldsymbol{x}^{t} + \alpha \cdot \text{sign}\left(\nabla_{\boldsymbol{x}} L(\theta, \boldsymbol{x}^{t}, y)\right), \boldsymbol{x}_0\}. \quad (4)$$

HopSkipJump attack finds optimal adversarial examples by iterative procedure and gradient estimate. Like Boundary, it starts from an adversarial point of a different class. For each iteration, it first moves towards the boundary of the two classes (true class vs. a wrong class) through binary search, then updates the step size along the estimated gradient direction through geometric progression until perturbation is successful, and lastly projects the perturbed sample back to the boundary again.

Though PGD and HSJ belong to different attack families, both have cluttered perturbations, except that the perturbations of HSJ are dimmer due to smaller perturbations. Though both methods are L_∞-norm based and bounded by 0.03, HSJ has perturbations of different scales ranging from -0.03 to 0.03, while PGD has more extremely perturbed pixels with a perturbation value of 0.03. From Fig. 9, we can see that the histograms of the perturbations of PGD and HSJ are very different. The histogram of PGD perturbations is like a bar plot because it updates depending on the sign of the gradients with a fixed step-size α, which explains the discrete bars in the distribution of PGD's perturbations. While HSJ does not use a fixed step size to update, it does not have such a pattern. We also test if the perturbations of PGD and HSJ focus on the object area. The same bounding box method in Section IV-A1 is used to calculate the proportion of significant perturbations inside the box for both attacks. A significant perturbation is defined as a perturbation whose absolute value is larger than the 90% quantile. In Fig. 10, we can see the in-box significant perturbation proportion histogram. HSJ's distribution is more left-skewed than PGD's; the average in-box significant perturbation proportions of PGD and HSJ are 50.37% and 60.38%, respectively. Therefore, even though PGD has access to the true gradient information, HSJ has more significant perturbations in the object area.

V. CONCLUSION

Our findings demonstrate attack methods from different attack families (gradient-based, score-based, decision-based) possess different characteristics. Given adversarial examples, the machine can learn such characteristics to identify which attack family they belong to. Further studies show that even attacks from the same family can be different. We systematically study the properties of the perturbation patterns of different attacks and explore where their differences come from. We hope that our work can shed light on a deeper understanding of adversarial attacks and help with the reverse engineering of adversarial attacks.

APPENDIX

This supplementary material provides more illustrative examples and details of those classification experiments. As mentioned in Section IV, Fig. 11 provides extra adversarial examples and corresponding perturbation patterns for C&W, ZOO, and Boundary, and Fig. 12 provides extra adversarial examples and corresponding perturbation patterns for PGD, Square, and HopSkipJump.

APPENDIX A
SUPPLEMENTARY EXAMPLES AND EXPERIMENT IN SECTION IV-A1

In Section IV-A1, we proposed that the plausible reasons for C&W attacking the main object are true gradients and starting the attack process from the original image. To verify the idea, we generate adversarial images based on two variants of C&W: the estimated-gradient C&W uses estimated gradients from NES instead of the true gradients, and random-start C&W generates adversarial images starting from a random adversarial image instead of the original image. More examples are displayed in Fig. 13.

Select those images that have been successfully attacked by all three attacks and split them into training and test sets of size 1764 and 756, respectively. Train a VGG16-based

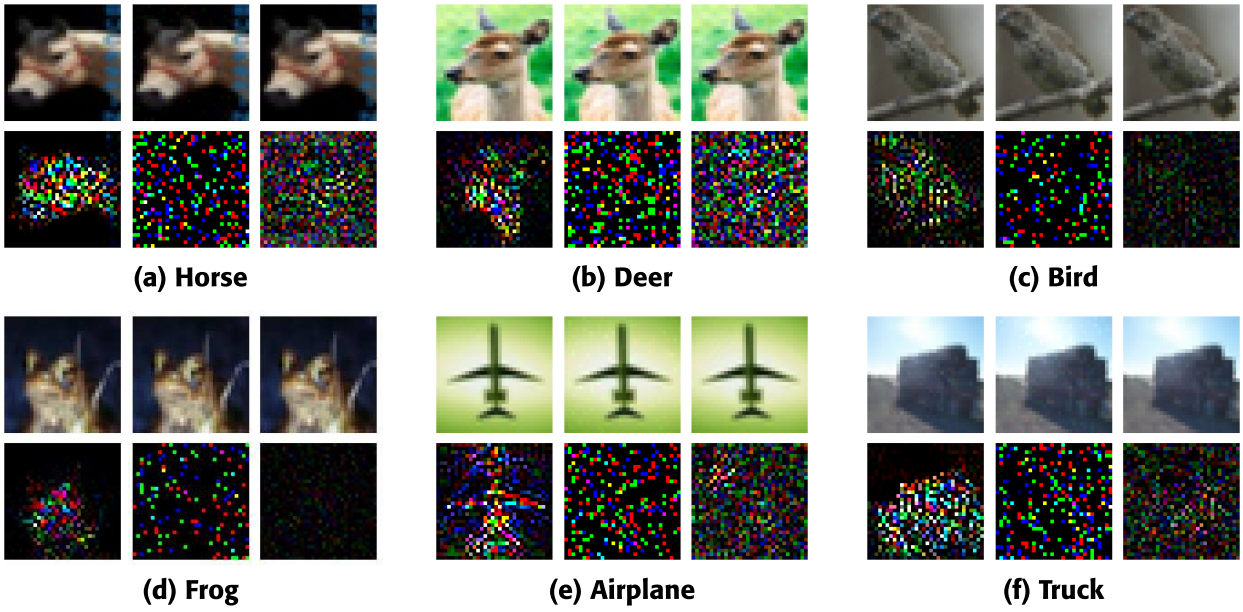

FIGURE 11. Visualization examples for C&W, ZOO, and Boundary are displayed in each subfigure, sampled from CIFAR10. From left to right, the first row shows the adversarial image generated by C&W, ZOO, and Boundary, and the second row shows corresponding amplified perturbations. Though adversarial examples are indistinguishable, perturbations show different patterns: C&W's perturbations focus on the main object; ZOO introduces scattered bright per-pixel perturbations; Boundary's perturbations are more uniform across the image.

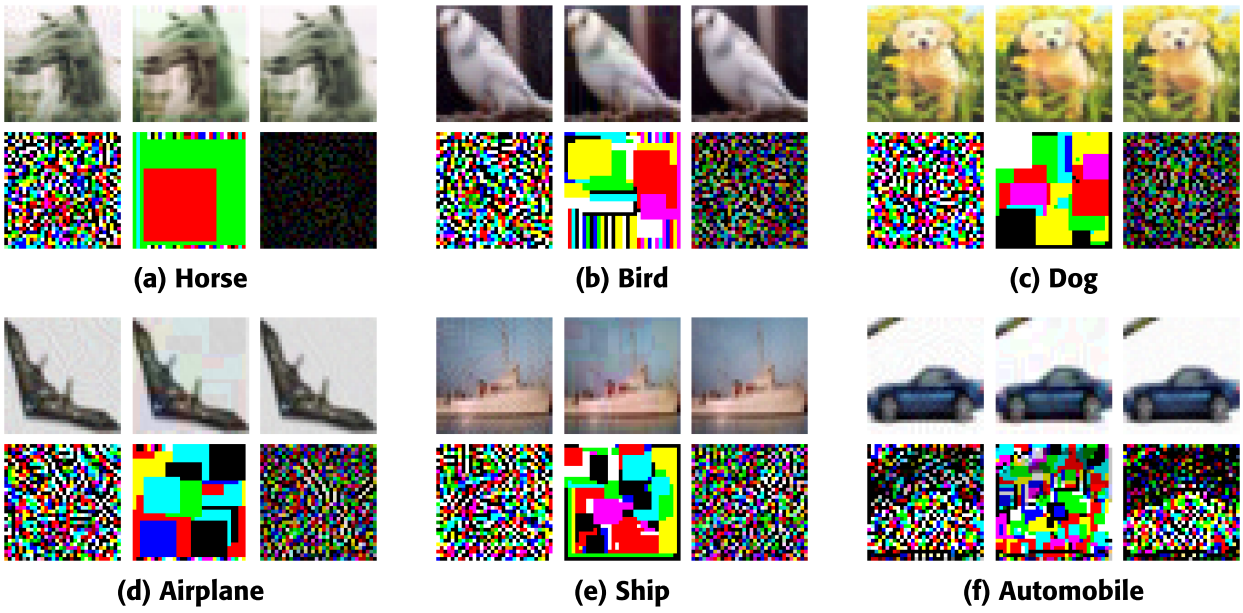

FIGURE 12. Visualization examples for PGD, Square, and HopSkipJump are displayed in each subfigure, sampled from the CIFAR10 data set. From left to right, the first row shows the adversarial image generated by PGD, Square, and HopSkipJump, and the second row shows corresponding amplified perturbations. PGD and HSJ have cluttered perturbation patterns, but HSJ is darker due to smaller perturbations. Square's perturbations consist of vertical strips covered by square-shaped regions, though vertical strips may not be obvious since it's covered by too many squares.

classifier to evaluate whether there's a difference among them. Accuracy reaches 96.03%. Table 5 records the confusion matrix of this classification task; we can see that both variants can be easily distinguished from C&W. This result further explains that the true gradients and original start affect C&W's performance.

APPENDIX B
SUPPLEMENTARY EXAMPLES AND EXPERIMENT IN SECTION IV-A2

ZOO is another variant of C&W with estimated gradients and coordinate descent. In Section IV-A2, to evaluate the optimization method's effect on perturbation patterns,

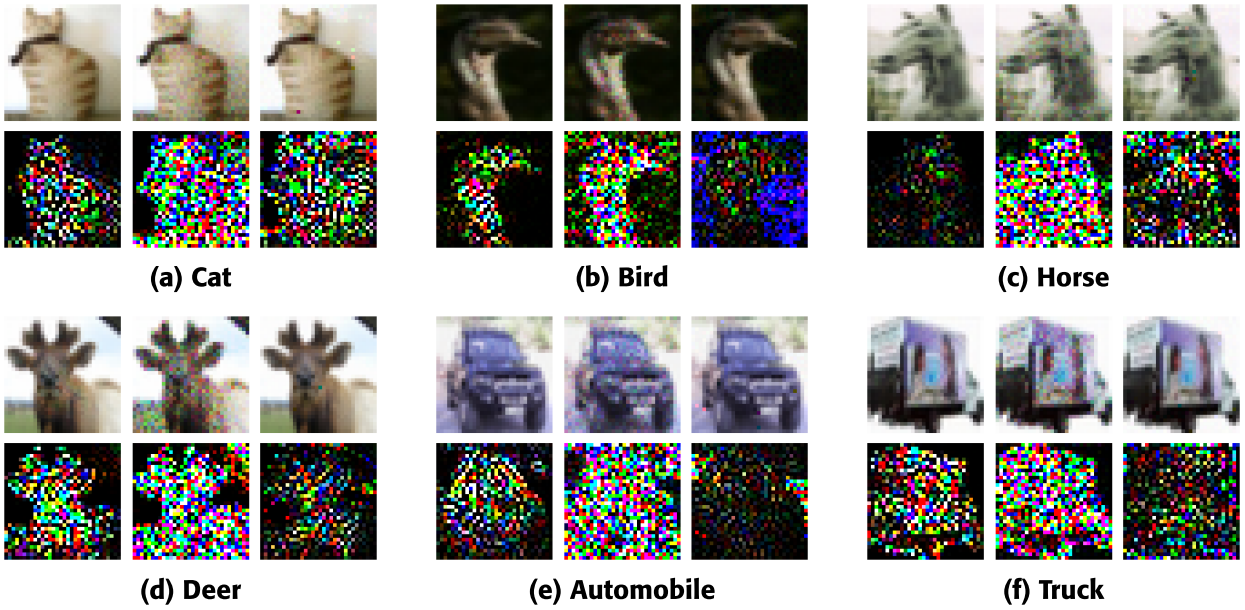

FIGURE 13. Each subfigure displays adversarial images and perturbations of C&W, estimated-gradient C&W, random-start C&W from left to right, sampled from CIFAR10 dataset.

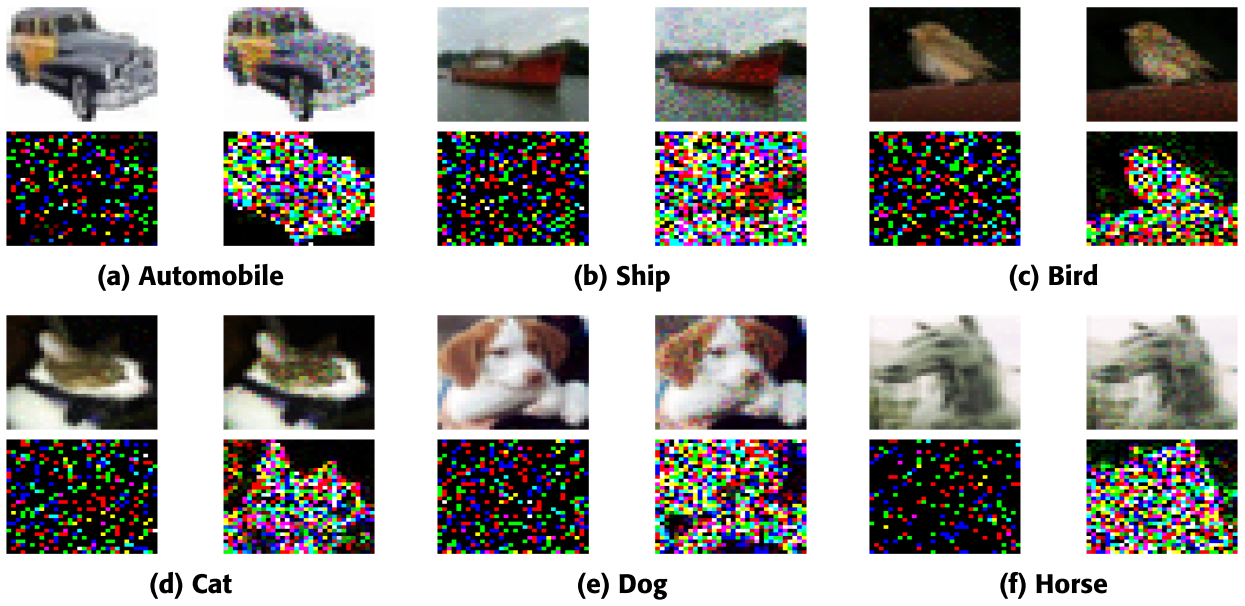

FIGURE 14. Additional visualization examples for ZOO and estimated-gradient C&W are displayed in each subfigure from left to right, sampled from the CIFAR10 dataset.

TABLE 5. Confusion Matrix for C&W, estimated-gradient C&W and random-start C&W.

		Predicted		
		C&W	estimated-gradient C&W	random-start C&W
Actual	C&W	247	0	5
	estimated-gradient C&W	2	249	1
	random-start C&W	22	0	230

TABLE 6. Confusion matrix for zoo and estimated-gradient C&W.

		Predicted	
		ZOO	estimated-gradient C&W
Actual	ZOO	825	38
	estimated-gradient C&W	3	860

we compare ZOO with estimated-gradient C&W; more examples are displayed in Fig. 14.

Select those images that have been successfully attacked by ZOO and estimated-gradient C&W and split them into training and test sets of size 2013 and 863, respectively. Table 6 records the confusion matrix of the classification result. The two attacks are separated by a highly accurate classifier, which shows an obvious effect when using different optimization methods.

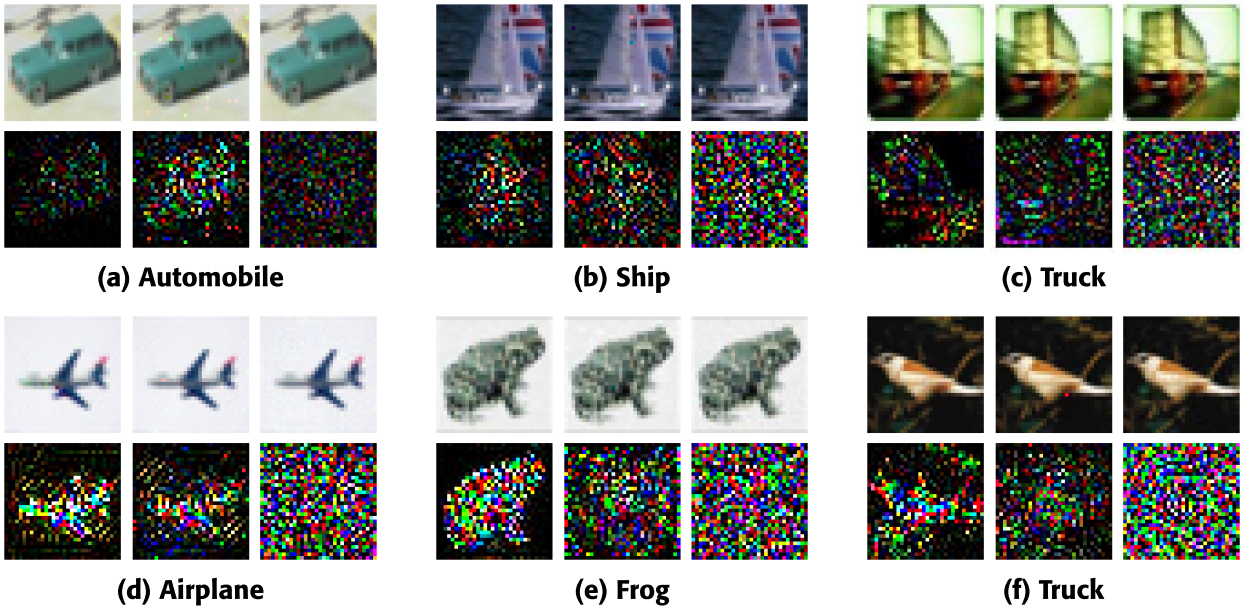

FIGURE 15. Additional visualization examples for C&W, random-start C&W, and Boundary are displayed in each subfigure from left to right, sampled from the CIFAR10 dataset.

TABLE 7. Confusion matrix for C&W, random-start C&W and boundary.

		Predicted		
		C&W	random-start C&W	Boundary
Actual	C&W	477	5	40
	random-start C&W	13	500	19
	Boundary	115	4	403

APPENDIX C
SUPPLEMENTARY EXAMPLES AND EXPERIMENT IN SECTION IV-A3

Boundary attack starts with a random adversarial image and uses a random walk for each update. In Section IV-A3, we study the effect of random start and lack of gradient information by comparing C&W, random-start C&W, and Boundary; more examples are displayed in Fig. 15.

Select those images that have been successfully attacked by all three attacks and split them into training and test sets of size 3645 and 1566, respectively. Table 7 records the confusion matrix. The three attacks can be classified by a high accuracy machine, indicating an obvious pattern among the attacks. This classification result proves that Boundary's blurry perturbations are caused by random start and random walk without gradient information.

REFERENCES

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[4] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus, "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*.

[5] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," 2014, *arXiv:1412.6572*.

[6] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2017, pp. 39–57.

[7] P.-Y. Chen, Y. Sharma, H. Zhang, J. Yi, and C.-J. Hsieh, "EAD: Elastic-net attacks to deep neural networks via adversarial examples," in *Proc. AAAI Conf. Artif. Intell.*, 2018, vol. 32, no. 1, pp. 1–8.

[8] H. Zhang, Y. Yu, J. Jiao, E. P. Xing, L. E. Ghaoui, and M. I. Jordan, "Theoretically principled trade-off between robustness and accuracy," in *Proc. 36th Int. Conf. Mach. Learn. (ICML)*, Jun. 2019, pp. 7472–7482.

[9] C. Xie, J. Wang, Z. Zhang, Z. Ren, and A. Yuille, "Mitigating adversarial effects through randomization," 2017, *arXiv:1711.01991*.

[10] D. Meng and H. Chen, "MagNet: A two-pronged defense against adversarial examples," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, Oct. 2017, pp. 135–147.

[11] K. Sadeghi, A. Banerjee, and S. K. S. Gupta, "A system-driven taxonomy of attacks and defenses in adversarial machine learning," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 4, no. 4, pp. 450–467, Aug. 2020.

[12] J. Hendrik Metzen, T. Genewein, V. Fischer, and B. Bischoff, "On detecting adversarial perturbations," 2017, *arXiv:1702.04267*.

[13] Z. Gong, W. Wang, and W.-S. Ku, "Adversarial and clean data are not twins," 2017, *arXiv:1704.04960*.

[14] X. Li and F. Li, "Adversarial examples detection in deep networks with convolutional filter statistics," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5775–5783.

[15] Z. Zheng and P. Hong, "Robust detection of adversarial attacks by modeling the intrinsic properties of deep neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–10.

[16] R. Feinman, R. R. Curtin, S. Shintre, and A. B. Gardner, "Detecting adversarial samples from artifacts," 2017, *arXiv:1703.00410*.

[17] Y. Han and T. C. M. Lee, "Uncertainty quantification for sparse estimation of spectral lines," *IEEE Trans. Signal Process.*, vol. 70, pp. 6243–6256, 2022.

[18] R. Pang, X. Zhang, S. Ji, X. Luo, and T. Wang, "AdvMind: Inferring adversary intent of black-box attacks," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Aug. 2020, pp. 1899–1907.

[19] M. Goebel, J. Bunk, S. Chattopadhyay, L. Nataraj, S. Chandrasekaran, and B. S. Manjunath, "Attribution of gradient based adversarial attacks for reverse engineering of deceptions," *Electron. Imag.*, vol. 33, no. 4, pp. 1–6, Jan. 2021.
[20] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, *arXiv:1706.06083*.
[21] Y. Gong, Y. Yao, Y. Li, Y. Zhang, X. Liu, X. Lin, and S. Liu, "Reverse engineering of imperceptible adversarial image perturbations," 2022, *arXiv:2203.14145*.
[22] A. Krizhevsky, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep., 2009. [Online]. Available: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf
[23] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[24] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proc. 10th ACM Workshop Artif. Intell. Secur.*, Nov. 2017, pp. 15–26.
[25] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," in *Proc. 35th Int. Conf. Mach. Learn.*, 2018, pp. 2137–2146.
[26] M. Andriushchenko, F. Croce, N. Flammarion, and M. Hein, "Square attack: A query-efficient black-box adversarial attack via random search," in *Proc. Eur. Conf. Comput. Vis.* Glasgow, U.K.: Springer, 2020, pp. 484–501.
[27] Z. Li, H. Cheng, X. Cai, J. Zhao, and Q. Zhang, "SA-ES: Subspace activation evolution strategy for black-box adversarial attacks," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 7, no. 3, pp. 780–790, Jun. 2023.
[28] H. Zanddizari, B. Zeinali, and J. M. Chang, "Generating black-box adversarial examples in sparse domain," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 6, no. 4, pp. 795–804, Aug. 2022.
[29] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," 2017, *arXiv:1712.04248*.
[30] M. Cheng, S. Singh, P. Chen, P.-Y. Chen, S. Liu, and C.-J. Hsieh, "Sign-OPT: A query-efficient hard-label adversarial attack," 2019, *arXiv:1909.10773*.
[31] J. Chen, M. I. Jordan, and M. J. Wainwright, "HopSkipJumpAttack: A query-efficient decision-based attack," in *Proc. IEEE Symp. Secur. Privacy (SP)*, May 2020, pp. 1277–1294.
[32] M.-I. Nicolae, M. Sinn, M. N. Tran, B. Buesser, A. Rawat, M. Wistuba, V. Zantedeschi, N. Baracaldo, B. Chen, H. Ludwig, I. M. Molloy, and B. Edwards, "Adversarial robustness toolbox v1.0.0," 2018, *arXiv:1807.01069*.
[33] F. Croce and M. Hein, "Minimally distorted adversarial examples with a fast adaptive boundary attack," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 2196–2205.
[34] F. Croce and M. Hein, "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *Proc. Int. Conf. Mach. Learn.*, 2020, pp. 2206–2216.
[35] D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, J. Peters, and J. Schmidhuber, "Natural evolution strategies," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 949–980, 2014.

XIAWEI WANG received the bachelor's degree in statistics from Nankai University, in 2017, the master's degree from the University of Wisconsin–Madison, in 2018, and the Ph.D. degree from the University of California, Davis, in 2024. Her research interests include responsible AI and the application of deep learning to medical imaging.

YAO LI received the bachelor's degree in statistics from Fudan University, in 2014, and the Ph.D. degree from the University of California, Davis, in 2020. Currently, she is an Assistant Professor of statistics and operations research with The University of North Carolina at Chapel Hill. Her research interests include trustworthy machine learning, computational pathology, and machine learning applications in other scientific disciplines.

CHO-JUI HSIEH is currently an Associate Professor with the Computer Science Department, UCLA. His work primarily focuses on enhancing the efficiency and robustness of machine learning systems, and he has made significant contributions to multiple widely-used machine learning packages. He has been honored with the NSF Career Award, the Samsung AI Researcher of the Year, and the Google Research Scholar Award, and his work has been acknowledged with several article awards in ICLR, KDD, ICDM, ICPP, and SC.

THOMAS C. M. LEE (Senior Member, IEEE) received the B.App.Sc. and B.Sc. (Hons.) degrees in math from the University of Technology, Sydney, Australia, in 1992 and 1993, respectively, and the Ph.D. degree jointly from Macquarie University and CSIRO Mathematical and Information Sciences, Sydney, in 1997.

Currently, he is a Professor of statistics and the Associate Dean of the Faculty of Mathematical and Physical Sciences, University of California, Davis. His research interests include inference methods, machine learning, and statistical applications in other scientific disciplines. He received the University Medal for the B.Sc. degree. He is an elected fellow of American Association for the Advancement of Science (AAAS), American Statistical Association (ASA), and the Institute of Mathematical Statistics (IMS). From 2013 to 2015, he served as the Editor-in-Chief for the *Journal of Computational and Graphical Statistics*. From 2015 to 2018, he served as the Chair of the Department of Statistics, UC Davis. He is the Review Editor for the *Journal of the American Statistical Association*.