CAN MACHINE TELL THE DISTORTION DIFFERENCE? A REVERSE ENGINEERING STUDY OF ADVERSARIAL ATTACKS

Anonymous authors
Paper under double-blind review

ABSTRACT

Deep neural networks have achieved remarkable performance in many areas, including image-related classification tasks. However, various studies have shown that they are vulnerable to adversarial examples – images that are 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, we study the features of three common attack families: gradient-based, score-based, and decision-based. In this paper, we demonstrate that given adversarial examples, attacks from different families can be successfully identified with a simple model. To investigate the reason behind it, we further study the perturbation patterns of different attacks with carefully designed experiments. Experimental results on CIFAR10 and Tiny ImageNet confirm the differences of attacks in distortion patterns. The code will be available at `https://github.com/reveng-adv/RevEng`.

1 INTRODUCTION

Well-trained deep neural networks are capable of achieving outstanding performance in image-related classification tasks. 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 (Szegedy et al., 2013; Goodfellow et al., 2014; Carlini & Wagner, 2017). This raises serious security concerns for the AI community. Many works have been done to study and defend against adversarial attacks. In particular, adversarial detection methods have been proposed to determine whether an input image is an adversarial example or not. 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, gradient-based, score-based, and decision-based, which rely on the gradient, predicted score and predicted label of the victim model respectively to perform attacks. 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. (2020) proposed the query of interest (QOI) estimation model to infer the adversary's target class by model queries in black-box settings. Goebel et al. (2021) estimated adversarial setup from image sample for gradient-based attacks FGSM (Goodfellow et al., 2014) and PGD (Madry et al., 2017). Gong et al. (2022) proposed a general formulation of the reverse engineering of deceptions problem that is able to estimate adversarial perturbations and provided the feasibility of inferring the intention of an adversary. In this paper, we find that adversarial attacks can be naturally identified by a simple VGG16-based model given adversarial examples. We focus on studying the features of each attack type to investigate where the differences of adversarial attacks come from.

We present our work in two aspects: (1) Given an adversarial image, use an image classifier to determine which attack family it belongs to (gradient-based, score-based, or decision-based). (2) We further study the features of different attacks and explore the reasons behind the good performance of the attack family classifier.

2 PRELIMINARIES

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. We denote $f(\boldsymbol{x}_0)$ as the output vector and $c(\boldsymbol{x}_0) = \arg\max_i f(\boldsymbol{x}_0)$ as the predicted label. The true label associated with $\boldsymbol{x}_0$ is denoted as y, and the adversarial example generated from $\boldsymbol{x}_0$ is denoted as $\boldsymbol{x}^*$.

Adversarial Examples An adversarial example $\boldsymbol{x}^*$ and the original image $\boldsymbol{x}_0$ are visually indistinguishable, but the predicted label of $\boldsymbol{x}^*$ is different from that of $\boldsymbol{x}_0$. 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 Figure 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. In this paper, we focus on ℓ_2 and ℓ_∞, the most commonly used metrics in adversarial attacks.

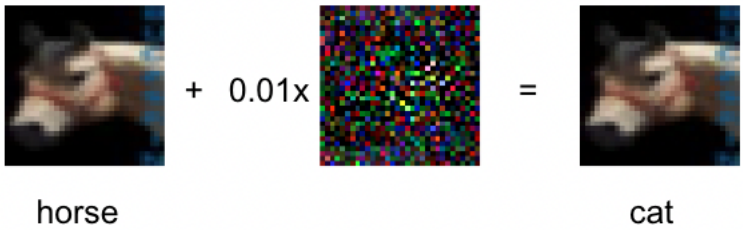


Figure 1: An adversarial example generated by Boundary attack: introducing adversarial perturbations to the horse image causes a classifier to label it as a cat.

Datasets and Victim Models We use CIFAR10 (Krizhevsky et al., 2009), a natural image dataset, which contains $50,000$ training images and $10,000$ testing images in ten different classes of resolution 32×32. Another dataset we use is Tiny Imagenet (Deng et al., 2009), which has 200 classes, and each class has 500 training images, 50 testing images. The resolution of the images is 64×64. For CIFAR-10, the victim model is VGG-16 with batch normalization (Simonyan & Zisserman, 2014), of which accuracy is 93.34%. For Tiny ImageNet, the victim model architecture is ResNet18 (He et al., 2016) with 68.64% accuracy.

Table 1: Attacks of different metrics (ℓ_2 and ℓ_∞) from different families.

	ℓ_2	ℓ_∞
gradient-based	C&W	PGD
score-based	ZOO	Square
decision-based	Boundary	HopSkipJump

Adversarial Attacks Different attack methods have been proposed to generate adversarial examples based on the original images and the victim model. Adversarial attacks can be classified into two categories according to their goals: untargeted and targeted. Untargeted attacks are successful as long as the adversarial example is mis-classified. Targeted attacks, instead, are successful only when the adversarial example is classified into a target class. Take Figure 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 cat. In this paper, all experiments are done with untargeted attacks.

Existing attack methods can be divided into three categories depending on the information required: gradient-based, score-based, and decision-based. The gradient-based attack is also known as white-box attack, in which all information of the victim model is revealed to the attacker so that the attackers can calculate gradients with the revealed information. Popular gradient-based attacks are FGSM (Goodfellow et al., 2014), PGD (Madry et al., 2017) and C&W (Carlini & Wagner, 2017). If an attack 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 (Chen et al.,

2017), NES (Ilyas et al., 2018) and SQUARE (Andriushchenko et al., 2020). In practical scenarios, the attacker only has access to the predicted labels of the model. Attacks work under this setting are called decision-based attacks, which include BOUNDARY (Brendel et al., 2017), Sign-OPT (Cheng et al., 2019) and HopSkipJump (HSJ) (Chen et al., 2020). Table 1 lists six representative attacks under different settings in ℓ_2 or ℓ_∞ metrics. In this paper, we conduct experiments with these attacks and study their properties.

Perturbation Visualization Perturbations are the difference 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. Another visualization way is to scale the perturbations to $[0, 1]$ range, then visualize them. Amplified perturbations tend to reveal the pattern when perturbation values are closer to each other, while the scaled perturbations work better when the perturbation values are relatively different. In this paper, we use amplified perturbation visualization for consistency.

3 Reverse Engineering of Adversarial Attacks

Most current reverse engineering methods focus on specific attack methods. In this section, we study if attack family (gradient-based, score-based, or decision-based) can be identified based on adversarial examples. Success detection of attack family can help defender figure out what level of information has been leaked to the attackers as different attack-family relies on different levels of information.

3.1 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 datasets and victim models mentioned in section 2. The perturbation upper bound is 0.03 for different ℓ_∞ attacks on CIFAR10 and Tiny ImageNet. For the ℓ_2 attacks, the perturbation upper bounds are 1.00 and 5.00 on CIFAR10 and Tiny ImageNet respectively. 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; (3) and classifying attack families with adversarial examples of both ℓ_2 and ℓ_∞ metrics. A classifier with VGG16 architecture is trained to do 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.

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.

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.

Experiment C: Adversarial examples in Experiment A and B are merged together 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 three-class classification, identifying the attack family given an adversarial example.

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: Testing accuracy of attack family classification task (Experiments A, B, C) and attack method classification task (Experiment D) on CIFAR10 and Tiny ImageNet.

	CIFAR10	Tiny ImageNet
Experiment A	80.81%	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 have different patterns as well. This brings out one question: what patterns does the classification model learn to identify the attack family and attack method?

4 Exploring Characteristics of Attack Families

Although adversarial examples from different attack families appear to be indistinguishable from each other, machines can learn and classify them with some subtle signatures. Since the differences of adversarial attacks are embedded in the perturbations, we propose to study why attack families can be easily identified by studying the perturbation patterns of different attacks.

4.1 ℓ_2 Attacks

Different ℓ_2 attacks modify the original images in different ways, resulting in different perturbation patterns. Figure 2 shows adversarial examples and amplified perturbations from C&W, ZOO, and Boundary from left to right. It is obvious that the perturbations of the three attacks are different. The perturbations of 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. See more examples in Appendix A.1. In the following sections, we study the characteristics of C&W, ZOO, and Boundary and discuss why they generate perturbations of different patterns.

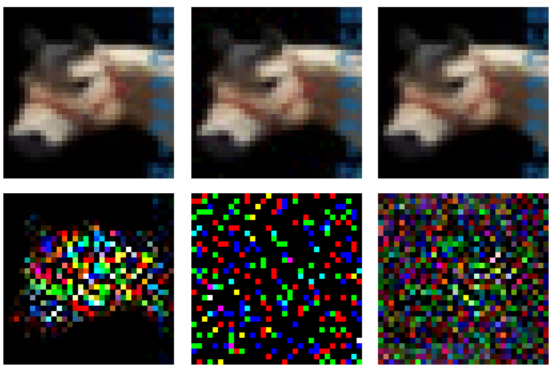


Figure 2: From left to right, the first row shows the adversarial image generated by C&W, ZOO and Boundary, and the second row shows the corresponding amplified perturbations. Though adversarial examples are indistinguishable, the perturbations show different patterns: C&W's perturbations focus on the horse in the image; ZOO introduces scattered bright per-pixel perturbations; Boundary's perturbations are more uniform across the image.

4.1.1 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 attack and can be formulated as the following regularized optimization problem:

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}\in[0,1]^n}{\arg\min}\{\|\boldsymbol{x} - \boldsymbol{x}_0\|_2^2 + cg(\boldsymbol{x})\}.$$

The first term $\|\boldsymbol{x} - \boldsymbol{x}_0\|_2^2$ enforces a small 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 Figure 2. To verify if this observation is true, we draw a bounding box of the object in the image and compute the proportion of ℓ_2 perturbations inside the box for all three attacks.

Take Figure 3 as an example, 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.

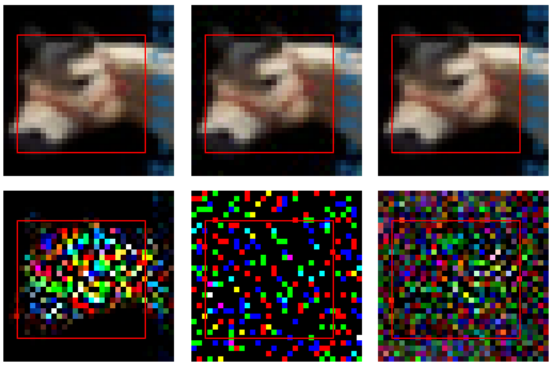


Figure 3: The proportion of perturbations inside the bounding box for C&W, ZOO, and Boundary are 96.40%, 69.25%, and 79.51% respectively.

To verify if the pattern is true for most cases, we randomly sample five images from each class of the CIFAR10 dataset and draw bounding boxes for all 50 images to calculate the proportions of perturbations inside bounding boxes. The proportion is calculated per sampled image for each attack. Figure 4 shows the histograms of in-box perturbation proportion for each attack. It is obvious that C&W has the most left-skewed distribution, indicating that C&W focuses on perturbing the main object in the image.

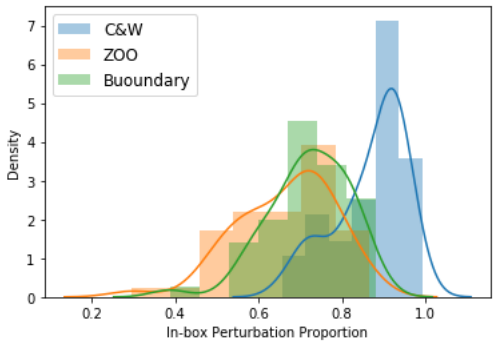


Figure 4: In-box perturbation proportion histograms for C&W, ZOO, and Boundary. C&W's distribution is most left-skewed, indicating C&W focuses on attacking the main object.

There are two reasons that might explain why C&W attacks the object: 1) C&W has access to the true gradients; 2) C&W method starts from the original image. Gradients w.r.t. the input indicate the important areas in the input image and usually concentrate on the objects because the victim model is trained to do object classification. Therefore, it is expected to see C&W focuses 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 (Wierstra et al., 2014), which was also used by Ilyas et al. (2018) 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 mis-classified 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 adversarial attacks are significantly different. Therefore, both gradients and random start affect the patterns of the C&W perturbations.

Figure 5 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 compared to 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 more noise in the background, even though many perturbations are in the object area. In conclusion, C&W's perturbations focusing on the object area comes from two factors: starting from original images and true gradients. See more examples in Appendix A.2.

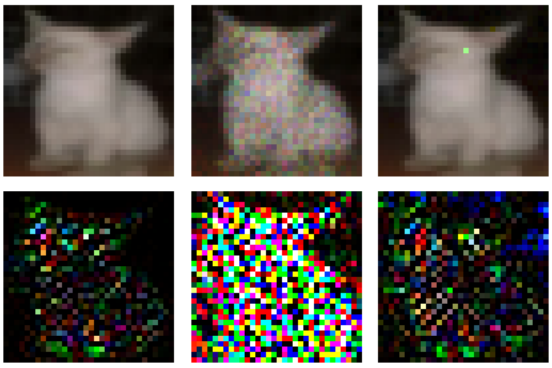


Figure 5: From left to right, a cat image is attacked by C&W, estimated-gradient C&W and random-start C&W. Even though the perturbations of estimated-gradient C&W and random-start C&W also roughly focus on the object area, it is not as obvious as in the perturbations of the original C&W.

4.1.2 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:

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

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 dimension of input. That is, ZOO is another variant of C&W but with estimated gradient and coordinate descent.

From Figure 2, we can see that ZOO's perturbations are made of a few bright pixels, which is expected as it uses coordinate descent to iteratively optimize each coordinate. Unlike gradient descent updating 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 between them 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. Figure 6 shows the adversarial examples and amplified perturbations of ZOO and estimated-gradient C&W from left to right, more examples are available in Appendix A.3. Compared to the estimated-gradient C&W, ZOO has more spread perturbations because of the optimization method. In section 4.1.1, 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.

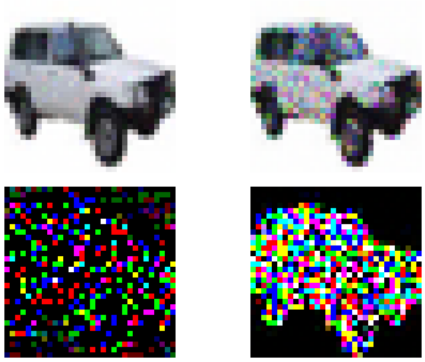


Figure 6: From left to right, an automobile image is attacked by ZOO and estimated-gradient C&W. The first row contains adversarial examples and the second row contains amplified perturbations. ZOO's amplified perturbations are more spread due to coordinate descent.

4.1.3 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, Boundary attack does not start from the original image and has no access to the gradient information.

From Figure 2, we noticed that Boundary attack's perturbations are spread across the entire image compared to C&W and ZOO. In fact, we verified in section 4.1.1 that starting from an adversarial point instead of the original image will make the perturbations more spread, and the gradient information is the key to an accurate attack on the object. This explanation applies to the perturbation patterns of Boundary attack as well. Figure 7 shows adversarial examples and perturbations of C&W, random-start C&W and Boundary from left to right, more examples are available in Appendix A.4. 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. Besides, unlike random-start C&W, Boundary's updating procedure relies on 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.

4.2 ℓ_∞ Attacks

ℓ_∞ attacks in different attack families show different perturbation patterns as well. In this section, we study the ℓ_∞-norm veriosn of PGD (gradient-based), Square (score-based) and HopSkipJump (decision-based). In our experiments, perturbations are bounded by 0.03. Figure 8 shows adversarial examples and perturbation patterns of PGD, Square and HopSkipJump (HSJ) from left to right. 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. See more examples

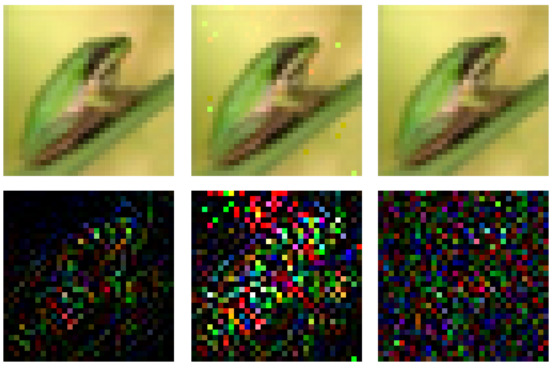


Figure 7: A frog image is attacked by C&W, random-start C&W and Boundary in turn. From left to right, the perturbations are getting noisier and the frog outline is blurring. It indicates both random start and random walk iteration without gradient information contribute to Boundary's noisy perturbations.

in Appendix A.5. In the following sections, we discuss the characteristics of Square first and then compare PGD and HSJ.

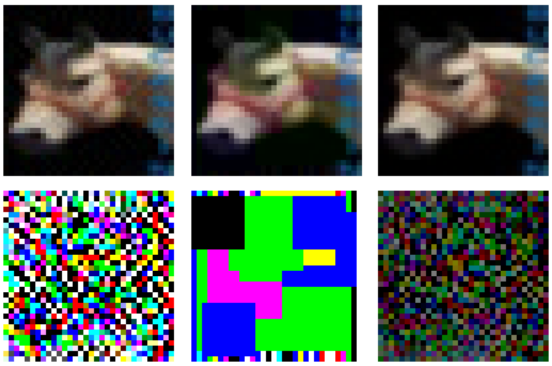


Figure 8: From left to right, the same horse image is attacked by PGD, Square and HopSkipJump as in the ℓ_2 example. It's noticed that PGD and HSJ have cluttered perturbation patterns, but HSJ is darker due to smaller perturbations. Square's perturbations consist of vertical strips covered by square-shaped regions.

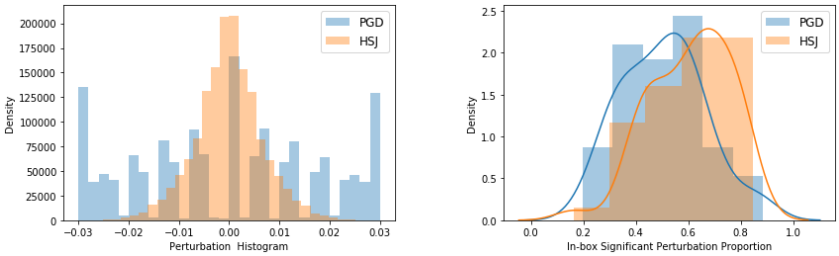
4.2.1 SQUARE ATTACK

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.

4.2.2 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

$$\boldsymbol{x}^* = \underset{\|\boldsymbol{x}-\boldsymbol{x}^*\|_\infty<\epsilon}{\arg\max}\ L(\boldsymbol{\theta}, \boldsymbol{x}, y),$$



(a) (b)

Figure 9: (a) Histogram of perturbation values of PGD and HSJ. PGD has a bar-plot-like perturbation distribution because it uses fixed step size to do updates, while HSJ has a normal-like perturbation distribution. (b) 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.

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

$$\boldsymbol{x}_{t+1} = \Pi_\epsilon\{\boldsymbol{x}_t + \alpha * \text{sign}\left(\nabla_{\boldsymbol{x}} L(\boldsymbol{\theta}, x, y)\right)\}.$$

HopSkipJump attack finds optimal adversarial example 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 scale ranging from -0.03 to 0.03, while PGD has more extremely perturbed pixels with a perturbation value of 0.03. From Figure 9a, 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 do updates, so 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 4.1.1 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 in the top 10%. In Figure 9b, 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.

5 Concluding Remarks

Our findings demonstrate that attack methods from different attack families (gradient-based, score-based, decision-based) possess different characteristics. Given adversarial examples, such characteristics can be learned by the machine 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 shade lights on deeper understanding of adversarial attacks and help with reverse engineering of adversarial attacks.

REFERENCES

Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, pp. 484–501. Springer, 2020.

Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017.

Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pp. 39–57. IEEE, 2017.

Jianbo Chen, Michael I Jordan, and Martin J Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *2020 ieee symposium on security and privacy (sp)*, pp. 1277–1294. IEEE, 2020.

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui 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*, pp. 15–26, 2017.

Minhao Cheng, Simranjit Singh, Patrick Chen, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. Sign-opt: A query-efficient hard-label adversarial attack. *arXiv preprint arXiv:1909.10773*, 2019.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Michael Goebel, Jason Bunk, Srinjoy Chattopadhyay, Lakshmanan Nataraj, Shivkumar Chandrasekaran, and BS Manjunath. Attribution of gradient based adversarial attacks for reverse engineering of deceptions. *Electronic Imaging*, 2021(4):300–1, 2021.

Yifan Gong, Yuguang Yao, Yize Li, Yimeng Zhang, Xiaoming Liu, Xue Lin, and Sijia Liu. Reverse engineering of imperceptible adversarial image perturbations. *arXiv preprint arXiv:2203.14145*, 2022.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*, pp. 2137–2146. PMLR, 2018.

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

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

Ren Pang, Xinyang Zhang, Shouling Ji, Xiapu Luo, and Ting Wang. Advmind: Inferring adversary intent of black-box attacks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1899–1907, 2020.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

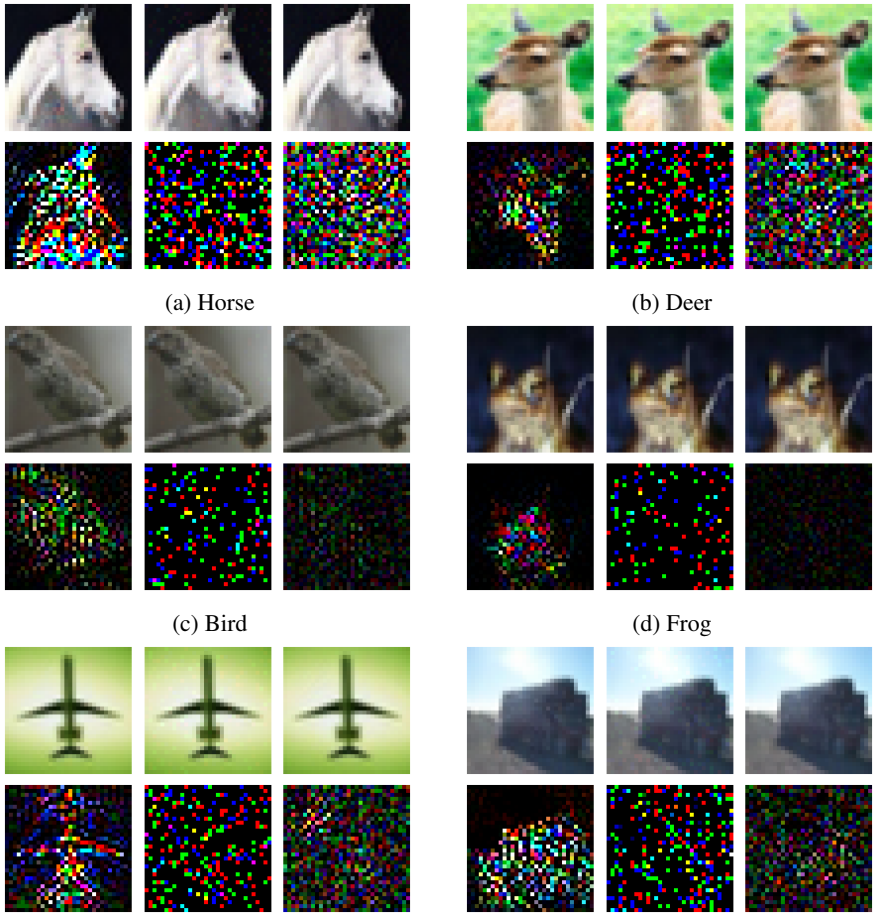
Daan Wierstra, Tom Schaul, Tobias Glasmachers, Yi Sun, Jan Peters, and Jürgen Schmidhuber. Natural evolution strategies. *The Journal of Machine Learning Research*, 15(1):949–980, 2014.

A APPENDIX

This appendix provides more illustrative examples and details of those classification experiments in Section 4.

A.1 ADDITIONAL EXAMPLES IN SECTION 4.1

Figure 10 shows additional adversarial samples and amplified perturbations for C&W, ZOO and Boundary, and each subfigure displays C&W, ZOO and Boundary from left to right.



(a) Horse (b) Deer

(c) Bird (d) Frog

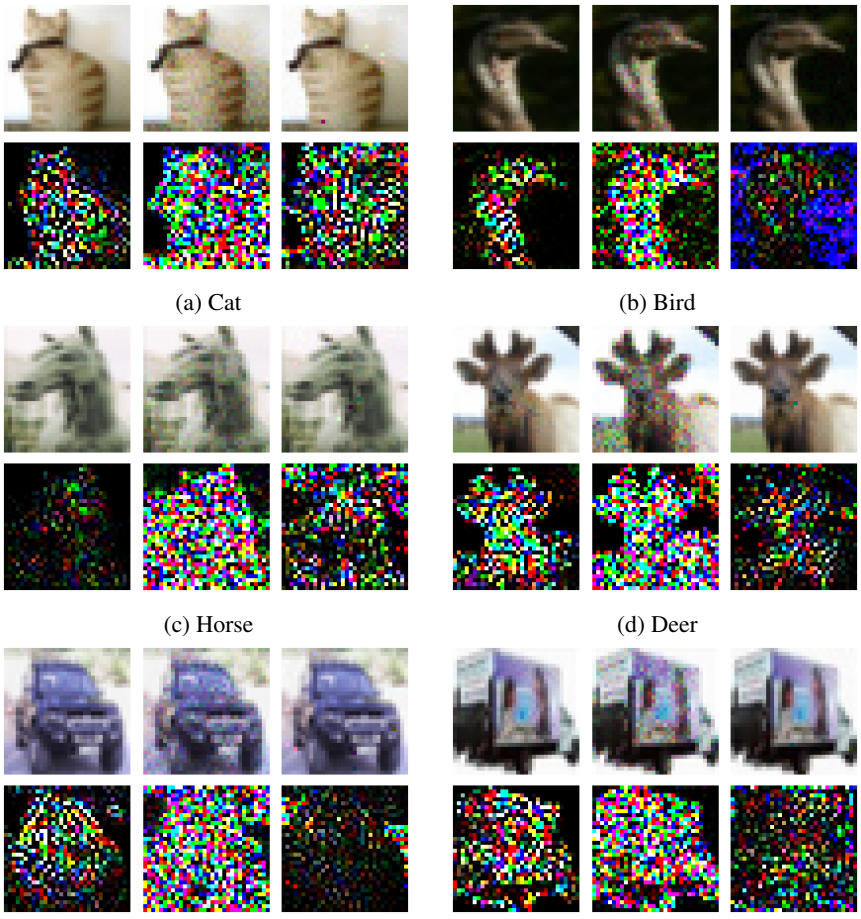
(e) Airplane (f) Truck

Figure 10: Additional visualization examples for C&W, ZOO and Boundary are displayed in each subfigure from left to right, sampled from CIFAR10 dataset.

A.2 SUPPLEMENTARY EXAMPLES AND EXPERIMENT IN SECTION 4.1.1

In Section 4.1.1, 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 Figure 11.

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 classifier to evaluate



(a) Cat (b) Bird

(c) Horse (d) Deer

(e) Automobile (f) Truck

Figure 11: 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.

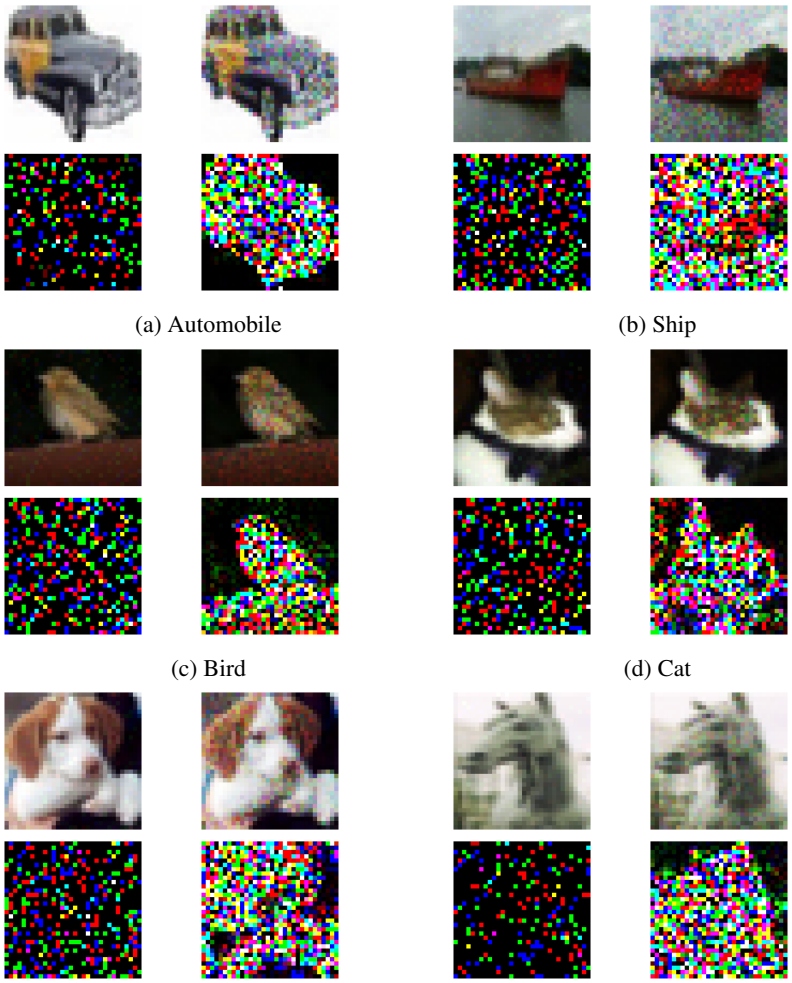
whether there's a difference among them. Accuracy reaches 96.03%. Table 3 records the confusion matrix of this classification task, we can see that both variants can be easily distinguished from C&W. This result further explains that the true gradients and original start affect C&W's performance.

Table 3: 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

A.3 Supplementary Examples and Experiment in Section 4.1.2

ZOO is another variant of C&W with estimated gradients and coordinate descent. In Section 4.1.2, to evaluate the optimization method's effect on perturbation patterns, we compare ZOO with estimated-gradient C&W, more examples are displayed in Figure 12. 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 4 records the confusion matrix of the classification



(a) Automobile (b) Ship

(c) Bird (d) Cat

(e) Dog (f) Horse

Figure 12: Additional visualization examples for ZOO and estimated-gradient C&W are displayed in each subfigure from left to right, sampled from CIFAR10 dataset.

result. The two attacks are separated by a classifier with high accuracy, which shows that there's an obvious effect when using different optimization methods.

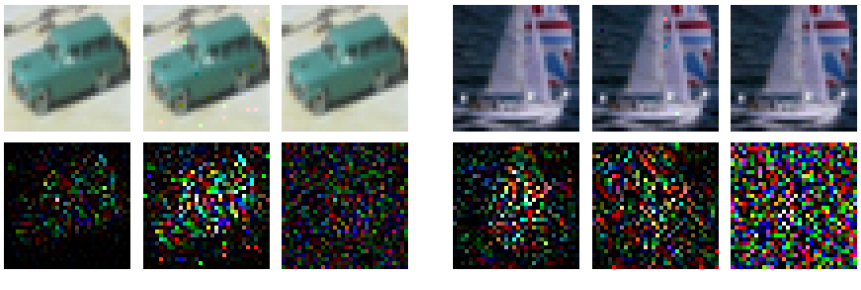
Table 4: Confusion Matrix for ZOO and estimated-gradient C&W

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

A.4 Supplementary Examples and Experiment in Section 4.1.3

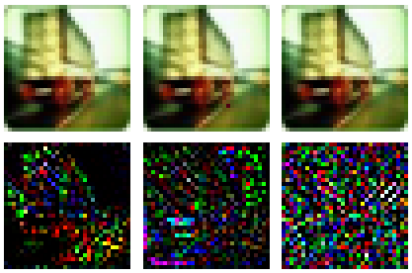
Boundary attack starts with a random adversarial image and uses random walk for each update. In Section 4.1.3, we study the effect of random start and lack of gradient information by comparing C&W, random-start C&W and Boundary, more examples are displayed in Figure 13:

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 5 records the confusion matrix. The



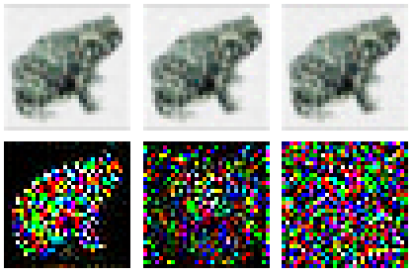
(a) Automobile

(b) Ship

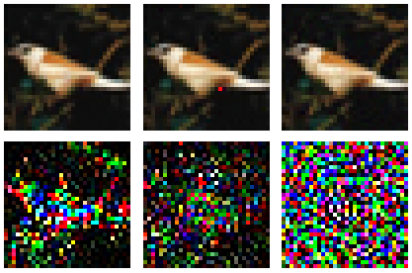


(c) Truck

(d) Airplane



(e) Frog



(f) Bird

Figure 13: Additional visualization examples for C&W, random-start C&W and Boundary are displayed in each subfigure from left to right, sampled from CIFAR10 dataset.

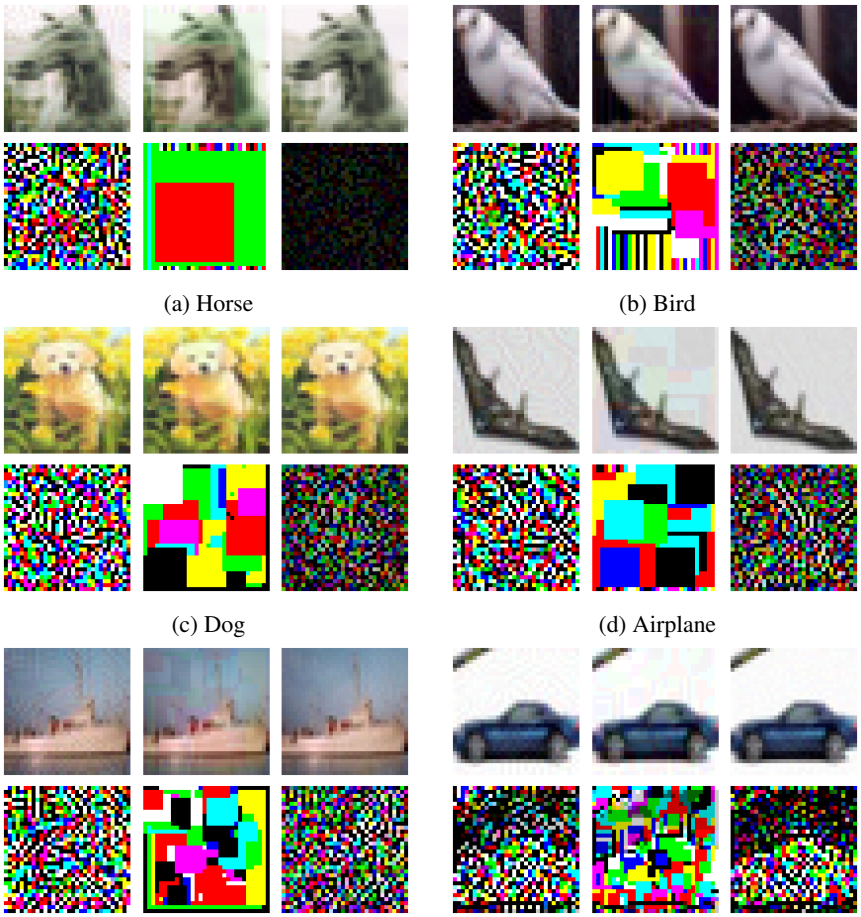
three attacks can be classified by a machine with high accuracy, indicating there's an obvious pattern among the attacks. This classification result proves that Boundary's blurry perturbations are caused by random start and random walk without gradient information.

Table 5: 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

A.5 Additional Examples in Section 4.2

Figure 14 shows additional adversarial samples and amplified perturbations. Each subfigure displays PGD, Square and HopSkipJump from left to right.



(a) Horse (b) Bird

(c) Dog (d) Airplane

(e) Ship (f) Automobile

Figure 14: Additional visualization examples for PGD, Square and HopSkipJump are displayed in each subfigure from left to right, sampled from CIFAR10 dataset.