

Explore the Transformation Space for Adversarial Images

Jiyu Chen
jiych@ucdavis.edu
University of California, Davis

David Wang
davidwqd@gmail.com
Vestavia Hills High School

Hao Chen
chen@ucdavis.edu
University of California, Davis

ABSTRACT

Deep learning models are vulnerable to adversarial examples. Most of current adversarial attacks add pixel-wise perturbations restricted to some L^p -norm, and defense models are evaluated also on adversarial examples restricted inside L^p -norm balls. However, we wish to explore adversarial examples exist beyond L^p -norm balls and their implications for attacks and defenses. In this paper, we focus on adversarial images generated by transformations.

We start with color transformation and propose two gradient-based attacks. Since L^p -norm is inappropriate for measuring image quality in the transformation space, we use the similarity between transformations and the Structural Similarity Index. Next, we explore a larger transformation space consisting of combinations of color and affine transformations. We evaluate our transformation attacks on three data sets — CIFAR10, SVHN, and ImageNet — and their corresponding models. Finally, we perform retraining defenses to evaluate the strength of our attacks.

The results show that transformation attacks are powerful. They find high-quality adversarial images that have higher transferability and misclassification rates than C&W's L^p attacks, especially at high confidence levels. They are also significantly harder to defend against by retraining than C&W's L^p attacks. More importantly, exploring different attack spaces makes it more challenging to train a universally robust model.

CCS CONCEPTS

• Security and privacy → Software and application security;

KEYWORDS

Adversarial attacks; Image transformation; Deep learning security

ACM Reference Format:

Jiyu Chen, David Wang, and Hao Chen. 2020. Explore the Transformation Space for Adversarial Images. In *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy (CODASPY '20)*, March 16–18, 2020, New Orleans, LA, USA. ACM, New York, NY, USA, 12 pages. <https://doi.org/10.1145/3374664.3375728>

1 INTRODUCTION

Deep neural networks achieved impressive performance in solving complex human perception problems, such as image classification. However, they are vulnerable to *adversarial examples*, which are

crafted by attackers to cause the victim classifier to output different classes than humans do. The key challenge for the attacker is to find an oracle for human perception: given an input, the oracle should output the class perceived by humans. Since we do not understand the mechanism of human perception completely, constructing a precise oracle is elusive, so researchers proposed approximate oracles. The most popular one is based on distance: two images that are close to each other under some metric should have the same class label perceived by humans. This oracle reflects our experience that perception is a continuous function, i.e., a small perturbation in the input causes no large change in the output.

To the best of our knowledge, all distance-based oracles measure the distance between two images using the L^p -norm, where L^1 , L^2 , and L^∞ are the most common. Most attacks and defenses can work on different L^p -norms, but some work on only certain L^p -norms. E.g., MadryNet[?] guarantees robustness against adversarial examples only in the L^∞ -norm ball. However, recent work [?] showed that a small L^p -norm is neither sufficient nor necessary for a perturbed image to be perceptually the same as the original one. Therefore, we set to explore *unrestricted* adversarial examples outside L^p -norm balls.

One evidence of the unnecessary of the L^p -norm is the invariance of human vision to several transformations. For example, human image classification is invariant to color transformations such as adjustment of brightness and is also invariant to spatial transformations such as rotation. These transformations, together with their different parameters and their combinations, constitute a large space where to search for adversarial examples. We wish to ask this research question: what advantages will the attacker gain by searching for adversarial examples in this space compared within a L^∞ -norm ball?

As a first step towards exploring the space of vision-invariant transformations and how they allow the attacker to generate more powerful and various adversarial examples, we study color transformations in the RGB color space and propose two gradient-based adversarial attacks: the optimization-based attack and the PGDT (Projected Gradient Descent on Transformation) attack. Next, motivated by prior works showing that deep learning models are not invariant to spatial transformations, we expand our search space by a two-phase combination of color and spatial transformations. After, we leverage the Structural Similarity Index to measure the perceptual quality of the generated adversarial example.

For evaluation, we choose three datasets and their corresponding classification models: CIFAR-10, SVHN, and ImageNet. These datasets are complementary as they contain different contents (natural objects vs. digits), have images of different sizes, and their classifiers have different scales (from fewer than five layers to tens of layers). We run the optimization-based attack to obtain high-quality and high-confidence adversarial images, and run the PGDT

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CODASPY '20, March 16–18, 2020, New Orleans, LA, USA

© 2020 Association for Computing Machinery.

ACM ISBN 978-1-4503-7107-0/20/03...\$15.00

<https://doi.org/10.1145/3374664.3375728>

attack to generate a large adversarial set for retraining defenses efficiently. The attacks are run with different settings in terms of the different number of image segments for color transformations, different initial points, and different confidence levels. Finally, we perform defenses by retraining the model in different attack spaces. Our evaluation shows that our transformation-based attack is more difficult to defend against by adversarial retraining than C&W’s L^∞ attack and L^2 attack.

In this paper, we make the following contributions:

- We define the color transformation space and establish the relationship between color transformation and L^p -norm restricted attacks by image segmentation.
- We propose two gradient-based attacks for the image transformation space and a two-phase attack to explore the large combination space of color transformation and affine transformation, and we show that the combination attack is more powerful with higher transferability than C&W’s L^p attacks.
- We perform extensive retraining defenses to show that training a universally robust model for adversarial examples from multiple different attack spaces is even more difficult.

2 BACKGROUND

In this section, we will provide a brief background for deep learning, adversarial attacks, and retraining defense.

Deep learning. Deep learning is a subset of machine learning, which is a branch of artificial intelligence that aims to learn models from data to deal with specific tasks. Deep learning models are usually deep neural networks. Each layer of the neural network is usually linear operations combined with non-linear activations. Formally, let x be the input example, f be a linear function with parameter θ and α be a non-linear activation function, we represent one layer by:

$$l(x, \theta) = \alpha(f(x, \theta))$$

the neural network model M is the composition of several layers:

$$M = l_0 \circ l_1 \circ l_2 \circ \dots \circ l_{n-1} \circ l_n$$

Usually, parameters of neural networks are trained by the Back-Propagation algorithm.

In this paper, we mainly study image classifiers, which are deep convolutional neural networks that take as input an image and outputs the predicted class label for the object contained in the image.

Adversarial examples. In 2013, Szegedy et al.[?] first pointed out the existence and crafted an adversarial image, which is a slightly perturbed image that can make the model give the wrong prediction. The definition of adversarial examples can be generalized. An example is an adversarial example of a target image classifier if it has the following two properties:

- Fidelity: Human can easily identify its true class.
- Adversary: The classifier’s prediction varies from the human’s.

The action of generating adversarial examples is called an adversarial attack. Depends on how much information does the attacker require, the attack can be divided into two categories:

- White-box attack: The attackers know everything about the model, including the dataset, the model structure, and the parameters.
- Black-box attack: The attackers only can access the model by sending input and retrieving the output.

Depends on the output of the classifier, the attacks can be divided into another two categories:

- Targeted attack: The attacker predetermines the output label of the adversarial example.
- Untargeted attack: The output label can be anything other than the true label.

In this paper, our attacks are targeted white-box attacks.

Attack methods. In recent years, researchers proposed many attack methods. We briefly introduce some representative attacks:

(1) Gradient ascent attacks: FGSM and PGD

Goodfellow et al.[?] proposed the Fast Gradient Sign Method or the FGSM. It is an untargeted attack by moving the image towards the opposite direction of the gradient of the loss function. Formally,

$$x' = x - \epsilon \cdot \text{sign}(\nabla_x J(\theta, x, y))$$

where x' is the adversarial example, ϵ is the step length, J is the loss function. FGSM attack is speedy and simple to implement. However, it is not a strong attack and can be easy to defend against[?][?].

An advanced variant of FGSM is the Projected Gradient Descent (PGD) attack. It iteratively performs FGSM attack at each step and projects the result onto the surface of a restricted space. The advantage of PGD attack is that it finds adversarial examples in the restricted space, such as a L^p -norm ball.

(2) Optimization based attack: C&W’s attack

C&W’s attack[?] is an optimization-based attack which minimizes the L^p distance between the adversarial image and the original image. Meanwhile, it leverages a hinge loss between the correct label and the target label to mislead the model. Formally, the C&W’s L^2 attack is:

$$\min_w \left\| \frac{1}{2} (\tanh(w) + 1) - x \right\|_2 + \lambda \cdot g\left(\frac{1}{2} (\tanh(w) + 1)\right),$$

$$g(x) = \max(-k, \max(f(x)_{i \neq t}) - f(x)_t)$$

where $x' = x + \eta = \frac{1}{2} (\tanh(w) + 1)$ is the adversarial image represented in the tanh space, g is the hinge loss term, k is the confidence score. The higher the confidence score, the higher the target class’s logit and also the possibility of attack transferring. C&W’s attack is shown the most powerful attack for breaking many defenses[?], and it has many variants such as ZOO attack[?], EAD attack[?], etc.

Retraining defense. Among all the recent proposed defenses, retraining, which retrain the model with adversarial examples, have shown the effectiveness of defending against norm bounded adversarial examples. For example, Madry et al.[?] try to defend against adversarial examples in an $^\infty$ ball with radius ϵ by solving:

$$\underset{\theta}{\operatorname{argmin}} \quad \mathbf{E}_{(x,y) \in X} \left[\max_{\delta \in [-\epsilon, \epsilon]^N} l(x + \delta; y; F_\theta) \right]$$

However, it only guarantees effectiveness in the L^∞ -norm ball, but not with other norms [?] nor unrestricted adversarial examples.

3 DESIGN

In this section, we will formally define the color transformation and introduce two kinds of adversarial attacks based on the transformation. Opposite to L^P restricted attacks which add pixel-wise perturbations, image transformations modify the image as an individual object. Besides, we leverage another transformation space - spatial transformation space and study the effectiveness of combinatorial attacks.

3.1 Color transformation

3.1.1 Definition of color transformation. Many image adjustments keep the semantic information of an image. For example, to change the illumination of an image, we add a constant to the RGB value of all the image pixels; to adjust the contrast of an image, we multiply the RGB value of all the image pixels by a scalar. Essentially, these adjustments are nothing but applying the same linear transformation to all images pixels. Formally, we define the color transformation to an image as:

$$x' = \{A \cdot p + rb | \forall p \in x\}$$

where p is a 3×1 pixel vector of the base image x in RGB color space, A is a 3×3 weight matrix, $b \in [-1, 1]^3$ is a 3×1 bias vector, r is the constant that represents the value range of the image pixels, and x' is the transformed image.

It is obvious that when A is a 3×3 identity matrix and b is a zero vector, the transformation will be identity transformation, where the transformed image will be identical to the base image. For simplicity, in the context of this paper, we will refer the 3×4 color transformation matrix M as

$$M = \begin{bmatrix} A & b \end{bmatrix}$$

and refer the the 3×4 identity color transformation matrix I_C as

$$I_C = \begin{bmatrix} I_{3 \times 3} & \mathbf{0} \end{bmatrix}$$

3.1.2 Color transformation with image segmentation. To create a larger transformation space, we can segment the base image into different regions, and apply a different color transformation to each region. There are many image segmentation methods, either based on similar colors or based on object semantics. Our goal of the segmentation is that we want to keep the semantic relationship among pixels, which means similar regions should remain similar after transformations.

In this paper, we apply a k -center segmentation to images. In detail, we preset k default colors as the k class centers and assign each image pixel to the nearest class measured by its euclidean distances to class centers. Finally, the pixels inside each class will be one image segment. Some segments can be blank if the number of colors inside the image is less than the set number k . The k -center segmentation is efficient while other clustering-based segmentation methods are more time consuming and may get worse performance when the colors inside the image are too close to each other.

Moreover, we can establish a relationship between pixel-wise noises and color transformations. Specifically, if we segment the

$m \times n$ image into $m \times n$ parts, and apply a different transformation to each of the image segments, we essentially add the perturbations to each of the pixels of the base image, which can be represented as:

$$p' = \begin{bmatrix} r + \delta_0 & g + \delta_1 & b + \delta_2 \end{bmatrix}^T = I \cdot p + \delta$$

where $\delta = \begin{bmatrix} \delta_0 & \delta_1 & \delta_2 \end{bmatrix}^T$ is the perturbation vector. We can see adding pixel-wise noise to the image is an extreme case of the color transformation with image segmentation.

When considering images as points in a high-dimensional Euclidean space, the search space of pixel-wise adversarial images is dense and narrow, which means there are a lot of possible examples inside a small L^P -norm ball. Color transformations have a sparser but broader search space, which means we can transform images far away while keeping their semantics, but cannot reach all the images in their neighborhoods. Applying image segmentation is a trade-off between the two search spaces.

3.2 Quality metrics for transformed images

Since transformed images are not bounded by L^P -norms, we leverage two other metrics to measure the human perception quality of the images after transformations.

3.2.1 Matrix distance. We leverage the distance between the transformation matrix and the identity transformation matrix to quantify the amount of transformation applied to the base image. There are many choices of measuring the distance of two matrices, such as matrix norms, cosine similarity, and correlation coefficient. In this paper, we take matrix norms as the matrix distance metrics, so the quality Q of the resulting image can be measured by

$$Q(x') = \|M - I\|_p$$

Similar to L^P -norms, we can assume that the less the transformation, the more similar the transformed image will be with the base image, thus the higher the quality of the transformed image.

3.2.2 Image quality assessment algorithms. Though the distance of the matrices is one applicable metric that measures whether the resulting image is of good quality, small matrix distance is not a necessary condition for high-quality transformations. Instead, we can measure the image quality directly from the transformed images.

Image quality assessment algorithms for measuring the quality of human perception has been studied extensively in recent years. In this paper, we choose the Structural Similarity Index (SSIM)[?], which measures the processed image quality from a structural perspective. SSIM is one of the full-referenced image quality assessment algorithms which takes the original image as a reference.

Formally, suppose w_1 is a $n \times n$ window of the base image, where n is the window size that is less than the width of the image. w_2 is a corresponding $n \times n$ window of the transformed image. SSIM of w_1 and w_2 can be computed by:

$$SSIM(w_1, w_2) = \frac{(2\mu_{w_1}\mu_{w_2} + c_1)(2\sigma_{w_1 w_2} + c_2)}{(\mu_{w_1}^2 + \mu_{w_2}^2 + c_1)(\sigma_{w_1}^2 + \sigma_{w_2}^2 + c_2)}$$

where μ is the average value, and σ is the variance or covariance, $c_i = (k_i L)^2$ are variables to stabilize the division with weak denominator, and L is the dynamic range of the pixel-values ($max -$

\min). The windows will be slid over the entire image to obtain the final SSIM value of the two images.

Typically, the range of SSIM is in $[-1, 1]$, with 0 means no structural similarity and 1 means two images are exactly the same.

3.3 Attack methods

We propose two gradient-based attacks within the image transformation space: the optimization attack and the PGDT attack. The two attacks are related as they are both gradient-based. However, they have different properties and serve different purposes.

Firstly, the optimization attack can guarantee that the adversarial examples found are of high confidence but cannot guarantee the image quality requirement, such as having SSIM value higher than a threshold. On the contrary, the PGDT attack can force the images (or the matrices) to be projected onto some restricted spaces, while these images may not be adversarial.

Secondly, the optimization-based attack is time-consuming while the PGDT attack is efficient. Therefore in this paper, we perform optimization attacks to generate high confidence adversarial images and perform PGDT attacks to quickly generate a large number of untargeted adversarial examples for adversarial retraining.

3.3.1 The optimization-based attack. Similar to C&W’s L^p attacks[?], we form the problem of looking for adversarial transformations as an optimization problem:

$$\begin{aligned} \min_M \quad & L(x, M) \\ \text{s.t.} \quad & f(T(x, M)) = y_{\text{target}}, \\ & T(x, M) \in [0, 1]^{m \times n} \end{aligned}$$

where x is the base image, $T(x, M)$ is the adversarial image generated by the transformation matrix M , L is any metric that measures the human perception quality of the adversarial image, m, n are the dimensions of the input image, f is the target image classifier we want to attack, y_{target} is the target label that we want the classifier to output for the adversarial image. In other words, we are looking for such color transformation matrices that preserve the quality and meanwhile make the transformed image misclassified, instead of restricting the image to have tiny perturbations in L^p -norm ball.

We preprocess the image the same way as in the C&W’s attack to make sure the transformed image is still in the valid range: transform the image into \tanh space and use the intermediate variable $w \in (-\infty, \infty)$ as the target of the optimization. Formally, the adversarial perturbation δ_i is defined as[?]:

$$\delta_i = \frac{1}{2}(\tanh(w_i) + 1) - x$$

where $\frac{1}{2}(\tanh(w_i) + 1)$ is in the range of $[-0.5, 0.5]$ and represents the adversarial example, therefore the result image will be guaranteed valid.

To solve the optimization problem directly is difficult, we can instead construct a specific loss function $L(x, M, \theta)$ that takes into account all the constraints, where f_θ is the model with parameter matrix θ , M is the transformation matrix, and find the optimal M by gradient descent based optimizers. We construct the loss functions for the color transformation attack as following:

$$L(x, M, f_\theta) = L_{\text{cr}}(x, M) + \lambda \cdot L_{\text{cl}}(x, M, f_\theta)$$

where L_{cr} is the image quality loss for measuring how bad the image quality is, it can use either of the metrics we mentioned in the previous section as the loss function, for example if we use SSIM as the image quality loss, then

$$L_{\text{cr}} = -\text{SSIM}(x, x')$$

and L_{cl} is the classification loss, which we take the same form of hinge loss as in C&W’s attack, and λ is the hyper-parameter that balances the two loss terms. The classification loss is defined as:

$$L_{\text{cl}}(x, M, f_\theta) = \max(-k, f_\theta(T(x, M))_{\text{true}} - f_\theta(T(x, M))_{\text{target}})$$

where k is the confidence of the adversarial example, higher k indicates higher confidence. $f_\theta(T(x, M))_{\text{true}}$ and $f_\theta(T(x, M))_{\text{target}}$ are the logits scores (the pre-softmax layer) of the transformed image $T(x, M)$ output by the model f_θ for the true label and the target label.

3.3.2 The Projected Gradient Descent attack for transformations. We propose the second attack based on the Projected Gradient Descent (PGD) attack. Since in color transformation attack, we are searching for the adversarial transformation matrices, we need to perform the gradient descent and the projection to the matrices instead of the images as in the pixel-wise PGD attack. Formally, we iteratively perform the gradient descent by

$$M_{i+1} = M_i + \delta \text{sign}(\nabla_{M_i} L(f(T(x, M)), y))$$

where M is the transformation matrix, i is the i th step, δ is the step length, f is the model prediction, L is the model loss function (eg. cross-entropy), T is the image transformation.

Next, we introduce two types of projections with respect to our different metrics of image quality for the transformation matrices.

- Project the transformation matrix onto the L^p -norm ball of the identity matrix. For L^0 -norm ball, We can project the matrix by clipping each entry of the transformation matrix by the radius ϵ . For L^2 -norm ball, we can project the matrix by dividing each entry of the transformation matrix by $\frac{\|M\|_2}{\epsilon}$.
- Project the transformation matrix so that the SSIM value of the transformed image is larger than a threshold value ϵ . Since we can also compute the gradient of the SSIM with respect to the transformation matrix M , inside each step of the gradient descent, we perform an inner gradient descent to project the SSIM value to our target value ϵ . The inner loop of projection requires more computation costs. Suppose the average number of iterations of SSIM projection is m , then the time complexity will be $O(mn)$, while the time complexity for L^p -norm ball projection is $O(n)$, where n is the number of steps of PGD.

We refer the PGD attack for transformation matrix as the *PGDT attack*. The detailed algorithm for the PGDT attack is shown in algorithm 1.

3.4 Enhance attacks by affine transformations

The spatial transformation space and the color transformation space are independent. Meanwhile, neural networks have been shown vulnerable to spatial transformations in prior works, so we can enhance the color transformation attack with the help of spatial

Algorithm 1: The PGDT attack algorithm

Data: Image x , True label y , Transformation matrix M , Radius or Threshold ϵ , Number of iterations n , Step length δ , Model loss function L , Model prediction f , Image transformation T

Result: The adversarial transformation matrix M , the adversarial example x'

```
 $i \leftarrow 0;$ 
 $M_i \leftarrow I;$ 
for  $i < n$  do
   $M_{i+1} \leftarrow M_i + \delta \text{sign}(\nabla_{M_i} L(f(T(x, M_i)), y));$ 
  if  $L^p$  projection then
     $M_{i+1} \leftarrow \text{clip}(M_{i+1}, B_p(I, \epsilon));$ 
     $x' \leftarrow T(x, M_{i+1});$ 
  end
  if SSIM projection then
     $x' \leftarrow T(x, M_{i+1});$ 
    while  $\text{SSIM}(x, x') < \epsilon$  do
       $M_{i+1} \leftarrow M_{i+1} + \delta \cdot \text{sign}(\nabla_{M_{i+1}} \text{SSIM}(x, T(x, M_{i+1})));$ 
       $x' \leftarrow T(x, M_{i+1});$ 
    end
  end
   $i \leftarrow i + 1$ 
end
return  $M_i, x'$ 
```

transformations, and explore the combination of different attack spaces. In this paper, we consider the linear spatial transformation - affine transformation.

3.4.1 Affine transformations. Affine transformations relocate the coordinates of image pixels while preserving the relative geometry properties. It is intriguing since it can also keep semantic information of the given image.

Formally, if we consider the following 2×3 affine transformation matrix:

$$M = \begin{bmatrix} m_{00} & m_{01} & m_{02} \\ m_{10} & m_{11} & m_{12} \end{bmatrix}$$

It will apply the following transformation to a given image:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} m_{00} & m_{01} \\ m_{10} & m_{11} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} m_{02} \\ m_{12} \end{bmatrix}$$

where (x, y) is the coordinate of one pixel. From the definition above, we can obtain the identity transformation matrix which maps one coordinate to itself:

$$I_A = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$$

Different elements of the transformation matrix correspond to different transformations to the image:

- m_{00}, m_{11} : change the scale of the width and height
- m_{01}, m_{10} : sheer the image towards some directions
- m_{02}, m_{12} : translate the image towards some directions

3.4.2 The two-phase combination attack. Motivated by the fact that simple semantic-preserving transformations such as scaling, translation, rotation are enough to decrease the accuracy of the model [?] largely. We explore the combination transformation space for adversarial examples in a two-phase manner:

- (1) Perform simple affine transformations to get different initial points
- (2) Perform small complex affine transformations combined with color transformations to find adversarial examples around the initial points.

The first phase of our attack is to find simple affine transformation matrices which preserve the semantics of the base image meanwhile decrease the accuracy of the target model. We refer these affine transformation matrices as *initial points*. Note that images transformed by the initial points are not necessarily adversarial images themselves. The selection of initial points can be either by manually choosing, random search, or gradient-based search. Following are example initial points:

- (1) Reflection

For most of the images that do not contain direction-critical items, such as a traffic sign, reflecting the image would keep the semantic information. The transformation matrix for reflection would be:

$$I_{\text{ref}} = \begin{bmatrix} \pm 1 & 0 & 0 \\ 0 & \mp 1 & 0 \end{bmatrix}$$

- (2) Rotation

Similarly, rotation of 90° will also keep the original shape and the semantic information of the base image. The transformation matrix for rotation of 90° (clockwise or counter-clockwise) would be:

$$I_{\text{rot}} = \begin{bmatrix} 0 & \pm 1 & 0 \\ \mp 1 & 0 & 0 \end{bmatrix}$$

The reason why not we also obtain initial points by color transformations is that same color transformation matrix can make different visual modifications on different images while the same affine transformation matrix makes the same.

In the second phase, we perform the combination attack to find small color transformations and small affine transformations that together create adversarial transformed images. We can perform the two gradient-based attacks mentioned in the previous section. The only difference is the transformed image x' is now ¹

$$x' = T_{\text{color}}(T_{\text{affine}}(x, M_{\text{affine}}), M_{\text{color}})$$

which is the combination of transformations. Note that gradient-based attack require the affine transformation operation is differentiable with respect to the affine transformation matrix M_{affine} . Since the affine transformation is made in the coordinate space rather than pixel space, we leverage the Spatial Transformer Network (STN) as the approximation of affine transformation operation. We will discuss STN in detail in Section 4.

In the combination transformation attack, we apply the SSIM metric as the image quality loss to avoid multiple loss terms. Since the SSIM is very sensitive to spatial transformation, which means

¹Different orders of color and affine transformations lead to similar results.

maximizing SSIM requires the affine transformation matrix M_{affine} to be as close to the initial point I_A as possible.

4 EVALUATION

We evaluated our color transformation attack and two-phase combination attack and compared them with C&W’s L^p attacks in terms of adversarial perturbations, quality of adversarial images, and the effectiveness of retraining defenses.

4.1 Threat model and experiment set up

We performed white-box, defense-agnostic attack, where the attacker has complete information of the target model, and the model does not have defense mechanisms. All attacks in Section 4.2 and Section 4.3 are optimization-based attacks.

We evaluated our image transformation attacks on three different data sets and their corresponding classification models:

- **CIFAR10** [?] contains 10 classes of 32×32 color images. The victim model was a convolutional neural network classifier pre-trained without data augmentation and had 78% test accuracy.
- **SVHN** [?] contains color images of house numbers from different views. We separated each image into individual digits and padded them into 28×28 images. The victim model was a single-digit classifier with 93% test accuracy.
- **ImageNet** [?] contains large color images of various classes. The victim model was Inception-V3[?], one of the most powerful classifiers for Imagenet.

We choose those three datasets since they represent different image types and model scales:

- Objects: natural (CIFAR10, ImageNet) vs. digits (SVHN)
- Image size: small (CIFAR10, SVHN) vs. large(ImageNet)
- Model scale: small (CIFAR10, SVHN) vs. large(ImageNet)

Our attack framework was similar to C&W’s L^p attack. We used the Adam optimizer with a learning rate $\alpha = 0.01$, set the maximum iteration for each run to be 10000 (50000 for Inception model), and ran 10 times to find the optimal hyper-parameters. The image quality metric was SSIM. We chose the base images to attack randomly from the original test set. The experiment ran on Keras & Tensorflow 1.13 on an NVIDIA TITAN RTX GPU.

4.2 Color transformation attack

In this section, we evaluated color transformation attacks and show how image segmentation influence the attacks.

















Figure 1a and Figure 1b show the result of color transformation attack without image segmentation on the CIFAR-10 model and the SVHN model. Adversarial images exist even by a simple color transformation without segmentation, although some adversarial examples do not have good enough quality. Note that the quality of SVHN adversarial images is worse than CIFAR-10 adversarial images. One possible explanation is that SVHN images are simpler than MNIST: the digits in SVHN have fewer colors and simpler shapes than the natural objects in CIFAR-10, so the attack requires larger transformation to find adversarial images.

Next, we evaluated how the image quality changes with different image segmentation settings. As shown in Table 1, we selected

eight primary colors and another eight intermediate colors for the k-center image segmentation described in Section 3. We evaluated two different color segmentation settings: (1) segmentation with eight primary colors, and (2) segmentation with eight primary colors and eight intermediate colors.

The examples of color transformation attacks with image segmentation are shown in Figure 1c through Figure 1f. Table 2 measures the quality of adversarial images generated by each of the image segmentation settings using the mean SSIM. For adversarial examples found in different segmentation settings based on the same image, the quality increases with the number of image segments, where adversarial examples based on 16 segments have the best quality. The results meet our expectations because the same color transformation is applied to every pixel in the same segment, so increasing the number of segments increases the attack space. However, it doesn’t mean that the more image segments, the better the attack. As we discussed in Section 3.1.2, more image segments means denser but narrower search space.

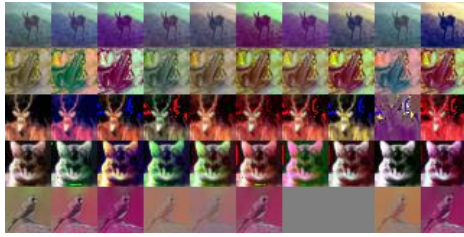
Table 1: Color classes for image segmentation

(a) Primary colors				
Color	Name	R	G	B
	Black	0	0	0
	White	255	255	255
	Red	255	0	0
	Lime	0	255	0
	Blue	0	0	255
	Yellow	255	255	0
	Cyan	0	255	255
	Magenta	255	0	255
(b) Intermediate colors				
Color	Name	R	G	B
	Silver	192	192	192
	Gray	128	128	128
	Maroon	128	0	0
	Olive	128	128	0
	Green	0	128	0
	Purple	128	0	128
	Teal	0	128	128
	Navy	0	0	128

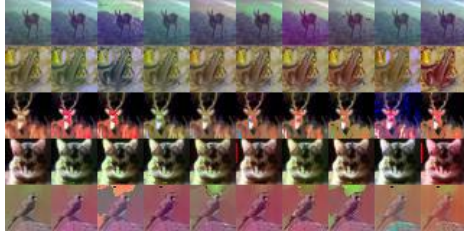
4.3 Combination attack

We evaluated the two-phase attack: (1) attack from different initial points, and (2) find adversarial images in the combined space of color transformation and affine transformation. We ran the combination attack with 16 image segments, and from three initial points we mentioned in Section 3.4.1: identity transformation, left-right reflection, and clockwise rotation of 90° .

The optimization attack requires that all the component of the loss function are differentiable with respect to the transformation



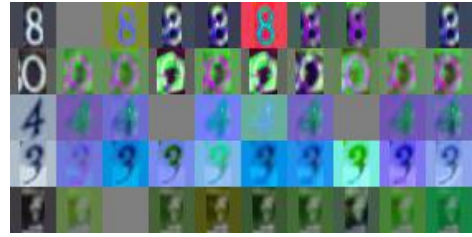
(a) CIFAR10, No image separation



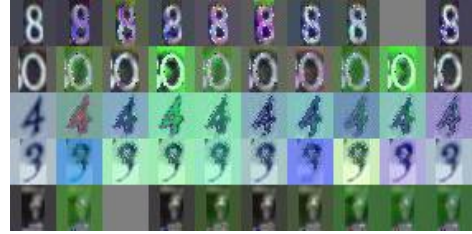
(c) CIFAR10, 8 image segments



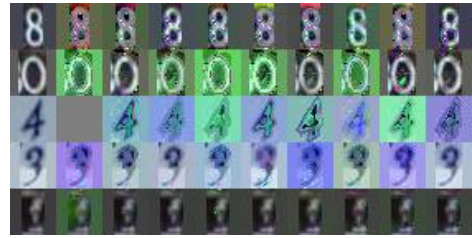
(e) CIFAR10, 16 image segments



(b) SVHN, No image separation



(d) SVHN with 8 image segments



(f) SVHN, 16 image segments

Figure 1: Adversarial examples of the color transformation attack. The leftmost column of images are the base images. The right columns are adversarial images whose targets are each of the other 9 classes of the dataset. For example, if the base image is a digit 0, then the images on the right are targeted adversarial examples to digits 1 – 9. For Inception model, the targeted classes are randomly chosen, and are shared among all the experiments.

Table 2: Quality of adversarial images for color transformation attacks with different segmentation settings. The quality is quantified by the mean SSIM value of the adversarial examples. As shown in the table, the adversarial images’ quality is increasing with the number of segments.

Model \ Segments	CIFAR10	SVHN
Single image	0.8479	0.6346
8 segments	0.9341	0.8380
16 segments	0.9408	0.8416

matrix. However, different from color transformations, affine transformations are applied to the pixel coordinates rather than the pixel values. As a result, we cannot compute the gradient flow directly through the affine transformation. We circumvented this problem by plugging in a pre-trained network called Spatial Transformer Network[?], which is a fully differentiable module trained

to take as input the affine transformation matrix and output the approximation of the transformed image, similar with the first-order method in [?].

Figure 2 shows adversarial examples found by the combination attack. Table 3 measures the amount of transformation by computing the Frobenius-norm distance between the identity transformation matrix (initial point for affine transformation) and the adversarial transformation matrix. To make a complete ablation test, we also performed the optimization attack with only affine transformation, where we replaced the color transformation node by the STN module. Same as in the previous section, we measured the human perception quality of the adversarial image by the mean SSIM. We can see that the combination attack usually required much less amount of both transformations, and the quality of the adversarial images is much better than single transformation attacks.

In Table 4, we compare different initial points. We can see that adversarial examples from initial points other than the identity matrix usually have better quality and need fewer transformations due to the lack of robustness of the model to simple affine transformations.

We ran the combination attack on the CIFAR-10 model with different confidence level from 0 to 50 to examine how the confidence level will affect the amount of transformation required and the quality of the image. Table 5 shows that when the confidence level increases, the amount of transformation required increases, and the image quality decreases. This result is consistent with that of C&W’s attacks.

Finally, we compared our combination attack (with 16 image segments and identity transformation matrix as the initial point) with C&W’s L^2 attack, which is one of the most powerful norm bounded attack. We computed the average L^2 distances from the adversarial images to the base images. Table 6 shows that adversarial images generated by our combination attack have good qualities similar to those by C&W’s attack while the average L^2 distance of our attack is larger than that of C&W’s attack. This result suggests that the transformation attack space is broader than the attack space restricted to the L^p -norm ball.

Table 3: Comparison of the quality and the amount of transformation required for different attacks. For color and combine transformation attacks, we set 16 image segments. The initial point is the identity affine transformation matrix. D_c and D_a are the Frobenius-norm distances between the transformation matrices and the identity matrix.

(a) SSIM value of different attacks			
Model \ Attack	Color	Affine	Combine
CIFAR10	0.9408	0.6370	0.9648
SVHN	0.8416	0.5759	0.8616

(b) D_c of different attacks			
Model \ Attack	Color	Affine	Combine
CIFAR10	273.1951	–	136.8899
SVHN	327.2462	–	389.2607

(c) D_a of different attacks			
Model \ Attack	Color	Affine	Combine
CIFAR10	–	0.1411	0.0046
SVHN	–	0.6392	0.0442

4.4 Retraining defenses

To evaluate the advantage of exploring different attack spaces rather than L^p norm ball, we performed different retraining-based defenses, and we evaluated each of the retrained models on adversarial examples from different attack spaces. We used the PGDT attack (Section 3.3.2) to create a large number of adversarial examples.

We considered the following retraining scenarios:

Table 4: Comparison of different initial points (IPs). The results (combination attack with 16 image segments) show that generating adversarial examples from initial points other than identity transformation leads to higher quality, and it usually requires less amount of transformations.

(a) SSIM values of different IPs			
Model \ IP	Identity	Reflection	Rotation
CIFAR10	0.9525	0.9566	0.9733
SVHN	0.8616	0.9402	0.9710
Inception	0.9786	0.9520	0.9879

(b) D_c of different IPs			
Model \ IP	Identity	Reflection	Rotation
CIFAR10	212.7731	154.4860	112.1367
SVHN	389.2607	569.1314	148.7909
Inception	176.9519	18.0094	110.9637

(c) D_a of different IPs			
Model \ IP	Identity	Reflection	Rotation
CIFAR10	0.0049	0.0004	0.0003
SVHN	0.0442	0.0080	0.0035
Inception	4.906×10^{-5}	1.363×10^{-5}	6×10^{-7}

- **Random augmentation**

The baseline defense. Any model designer can randomly augment the dataset by performing random color transformations without being aware of any attacks. In our experiment, we make the random data augmentation in the following steps:

- (1). Generate a 3×4 matrix Δ with each entry randomly drew from $[0,1]$.
- (2). Multiply Δ with a scalar α which controls the amount of maximum possible transformation. Here we set $\alpha = 0.25$.
- (3). Add the matrix Δ to the identity transformation matrix, and the final random transformation matrix will be $M = I + \Delta$.
- (4). Generate a randomly transformed image $x' = T(x, M)$.
- (5). For each of the examples in the training set, repeat the step (1)–(4).

Since the amount of affine transformation that our attack required is very small, we will not make augmentation for affine transformations.

- **Retrain with the PGDT attack from single initial point**

The defense targets the combination attack from the single initial point (identity matrix). We applied the PGDT attack to obtain an untargeted adversarial example for each of the training examples in the training set.

- **Retrain with PGDT attack from different initial points**

The defense targets the combination attack from multiple

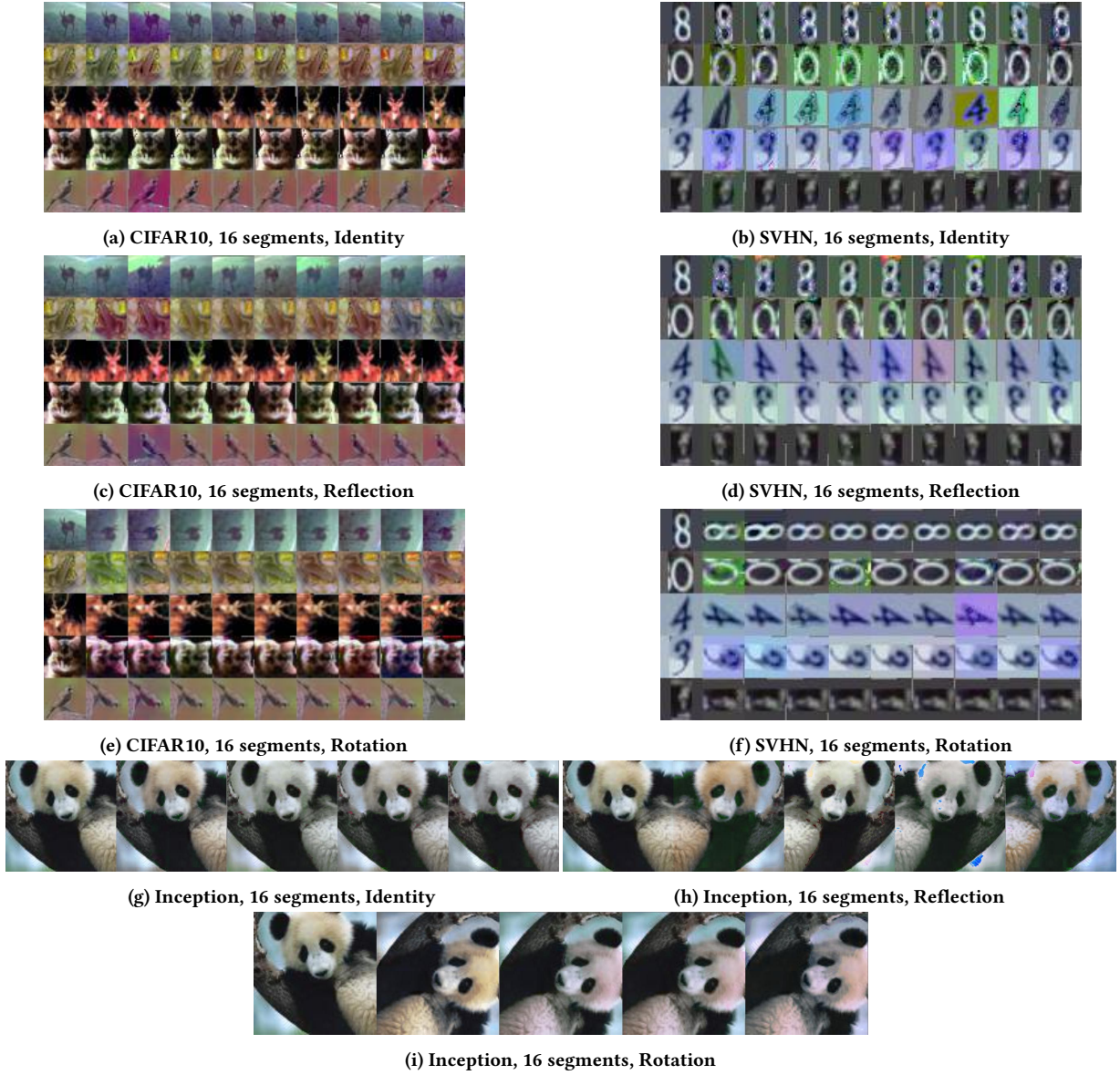


Figure 2: Adversarial examples of the combination attack. The leftmost column of images are the base images or the original images. The right columns are adversarial images whose targets are the other 9 classes of the dataset (or randomly chosen target classes for the Inception model). The result adversarial examples are of good human perception quality and are L^p -norm unrestricted.

Table 5: Quality and amount of transformation required of adversarial images under different confidence levels (16 segments, identity transformation matrix as the initial point).

Metric \ Confidence						
	0	10	20	30	40	50
SSIM	0.9648	0.9195	0.8513	0.7650	0.6664	0.5825
D_c	136.8897	201.0818	310.2363	429.6003	560.2861	718.0401
D_a	0.0046	0.0056	0.0075	0.0122	0.0225	0.0232

Table 6: Comparison of the adversarial images from the combination attack and C&W’s L^2 attack. For the combination attack, we use 16 segments and identity transformation matrix as the initial point.

(a) Average L^2 distances between adversarial and base images

Attack \ Model	CIFAR10	SVHN	Inception
Combine attack	3.73	5.69	20.08
C&W’s L^2 attack	0.86	0.92	2.55

(b) Average quality of human perception

Attack \ Model	CIFAR10	SVHN	Inception
Combine attack	0.9648	0.8616	0.9786
C&W’s L^2 attack	0.9713	0.9620	0.9927

initial points. We retrained the model with adversarial examples generated by PGDT attack from different initial points.

- **Retrain with different attack spaces**

The defense’s targets are adversarial examples from both the image transformation space and the L^∞ -norm ball. We retrain the model with adversarial examples generated by both PGDT attack (from identity) and L^∞ PGD attack.

We chose CIFAR-10 and its classifier as our target. We performed one-step retraining, in which we retrained the model from scratch by adding adversarial examples to the training data. All the training configurations, such as the amount of training data and training epochs, are shared among all the retraining processes.

Table 7 shows the result of the evaluation of the retrained models. For simplicity, we write Eval_D^E to represent evaluating model retrained with set D on evaluation set E. For example $\text{Eval}_{3,5}^E$ represent evaluating model retrained with both 3 and 5 on the evaluation data **e**. We make the following observations:

Random augmentation provides limited robustness. From columns of **1** and **2**, we find that random augmentation provides nearly no robustness improvement compared with directly retraining the model with no augmentation.

Transformation attacks has higher transferability. From $(\text{Eval}_1^a, \text{Eval}_1^b, \text{Eval}_1^c)$ and $(\text{Eval}_1^d, \text{Eval}_1^e, \text{Eval}_1^f)$, by evaluating the model retrained directly without any extra data, we show that our transformation attack is more likely to transfer compared with C&W’s L^∞ and L^2 attack when increasing the confidence level.

Transformation attacks are powerful. From the rows of **a** and **d** show that high confidence transformation adversarial images are hard to defend against. Even if we run the PGDT attack with 20 times larger projection radius, the accuracy of the high confidence adversarial examples has no significant improvement (from $\text{Eval}_3^d = 50.89\%$ to $\text{Eval}_4^d = 52.33\%$). Meanwhile, the C&W’s L^∞ attack and L^2 attack with the same high confidence levels can be defended against more effectively with retraining ($\text{Eval}_6^e = 77.22\%$, $\text{Eval}_6^f = 76.66\%$).

Training a universally robust model is hard. From the entire Table 7, we find that all the best accuracy values are acquired by the models retrained with only one initial point or single attack space. The models trained on multiple initial points or attack spaces cannot perform as good as training on each of the single spaces.

Retraining reduces model accuracy. From the row of **o**, we can see retraining with adversarial examples reduces model accuracy on the original test data. The result is unsurprising since adversarial examples introduce more noises and biases in the training data.

In conclusion, exploring the transformation space for adversarial examples has the following advantages. First, random augmentation contributes little to the robustness against our transformation attack. Second, high confidence transformation attack is more likely to transfer than C&W’s L^∞ attack and L^2 attack. Third, it is harder to train a universally robust model against different attack spaces or different initial points than only considering one attack space.

5 RELATED WORK

After proposals of various L^p -norm restricted adversarial attacks in previous years, researchers begin to consider the necessity of L^p restrictions and try to find alternative metrics for measuring human perception quality. For example, Rozsa et al. [?] propose to apply SSIM as the similarity metric. SSIM qualifies an image by its illuminance, contrast, and structural similarity to the original image, which makes more sense than merely L^p distances. Later, Sharif et al. [?] have shown that L^p -norms are neither necessary nor sufficient for measuring image perceptual similarity by generating both high perceptual quality adversarial examples with large L^p distances and human-misclassified examples with small L^p distances. The result indicates that adversarial attacks can also be unrestricted.

Many unrestricted attacks have been proposed. For example, in paper [?], Brown et al. proposed that people can generate a universal, robust, targeted patch that fools classifiers regardless of the scale or location of the patch, and does not require knowledge of the other items in the scene that it is attacking. In paper [?], Yang et al. proposed a method that modifies the variable of the latent space of a GAN to generate adversarial images from scratch, which means there is no requirement for any base image. The main difficulty of these unrestricted attacks is that there is no proper metric for measuring human perception quality, so the modifications are either too obvious or need human validation. Our attacks are also unrestricted attacks, meanwhile having well-defined quality metrics.

Spatial transformations on images are also studied broadly in previous works. Engstrom et al. [?] proposed that very simple affine transformation with only rotations and translations can lead to adversarial examples. We generalize his First-Order method which only optimizes over the latent space of rotation and translation to optimizing over the entire transformation matrix to find the adversarial transformation that could be the combination of scalings, shearings, rotation, and translations. There are more sophisticated spatial transformations such as [?] and [?], however, since our

Table 7: Accuracy of the retrained model on the adversarial images generated for the original CIFAR-10 classifier. There are 90 000 retraining images, and the training epochs is set to be 10. We randomly drew 100 examples from the original test set and, for each test example, generated 9 adversarial examples targeting each class.

Data for retraining

0. No retraining (the original model)
1. Retraining using the original training set
2. Randomly augmented the original training set
3. PGDT attack on the original training set. The transformation matrices are projected onto the L^∞ -norm ball of identity matrix with radius 0.25, step length 0.025, and number of steps 10.
4. PGDT attack on the original training set with a larger projection radius. The transformation matrices are projected onto the L^∞ -norm ball of identity matrix with radius 5, step length 0.5, and number of steps 10.
5. PGDT attack on the original training set rotated by 90°. The transformation matrices are projected onto the L^∞ -norm ball of identity matrix with radius 0.25, step length 0.025, and 10 steps.

6. PGD attack on the original training set. Images are projected onto the L^∞ -norm ball of original image with radius 0.05.
7. PGD attack on the original training set. Images are projected onto the L^2 -norm ball of original image with radius 0.2.

Data for evaluation

- a. Original test data
- a. Combine transformation attack, confidence level 0
- b. C&W's L^∞ attack, confidence level 0
- c. C&W's L^2 attack, confidence level 0
- d. Combine transformation attack, confidence level 30
- e. C&W's L^∞ attack, confidence level 30
- f. C&W's L^2 attack, confidence level 30
- g. Combine transformation attack, confidence level 0, rotated 90°

Retrain \ Eval	0	1	2	3	4	5	6	7	3,5	3,6	3,5,6
o	77.82 %	77.08 %	78.67 %	75.62 %	77.18 %	75.33 %	73.83 %	77.34 %	76.38 %	74.00 %	74.78 %
a	0.0 %	68.56 %	68.67 %	79.67 %	73.89 %	71.11 %	65.89 %	71.33 %	69.22 %	71.78 %	72.22 %
b	0.0 %	82.78 %	76.33 %	76.33 %	74.55 %	72.22 %	79.11 %	81.77 %	69.44 %	77.00 %	78.78 %
c	0.0 %	82.78 %	81.67 %	77.78 %	74.89 %	71.99 %	78.44 %	82.00 %	68.56 %	76.89 %	79.44 %
d	0.0 %	29.44 %	30.56 %	50.89 %	52.33 %	40.89 %	34.89 %	26.00 %	46.33 %	48.22 %	45.00 %
e	0.0 %	70.78 %	63.44 %	58.11 %	58.00 %	61.89 %	77.22 %	63.88 %	59.00 %	74.22 %	74.33 %
f	0.0 %	70.67 %	70.89 %	60.00 %	59.11 %	62.78 %	76.66 %	63.55 %	60.56 %	72.78 %	73.89 %
g	0.0 %	25.22 %	27.33 %	19.44 %	13.89 %	71.44 %	25.22 %	21.33 %	61.11 %	19.11 %	54.78 %

primary goal of leveraging spatial transformation is to enhance the color transformation and explore the combination attack space, affine transformations are enough for our work.

Some previous works also studied making transformations to image colors. Hosseini et al. proposed to look for semantic adversarial examples in the HSV space. Their method is based on applying random search in the HSV space to find misclassified examples, which requires no gradient information, and has no restrictions on the transformed images. As a result, many of their adversarial examples are modified too much in color (e.g., a green bird) thus do not maintain the same semantic information as the original images. Zhang et al. proposed the blind-spot attack in [?]. They apply very simple linear transformation on the entire image by $x' = \alpha x + \beta$, where α and β are two constants. Compared with their work, we formally define color transformation in the RGB space, thus do not require any color-space translations. Our attacks are gradient-based, can find either high confidence adversarial images or efficiently perform a large number of attacks. Though not bounded by L^p -norms, our adversarial images are still restricted by some metrics for human perception quality, so that they can maintain the semantics of the original image.

6 DISCUSSIONS

In this section, we make discussions upon the limitations and possible improvement of our attacks, and some general problems in the robust machine learning area.

The speed of our attack. One limitation of our optimization-based attack is the attack speed. It may take an hour to find one good adversarial example for large models like the Inception model. The limitation is from the large size of the image, the complex model architecture, the large number of matrix multiplications, and tuning hyper-parameters. How to speed up the optimization-based attacks will be one of our future directions.

Quality metrics for unrestricted attacks. The transformations we used in our attacks are all linear transformations. In the future, we can explore more general image transformations, including non-linear transformations. However, compared with increasing the variety of attack spaces, it is more important to find proper quality metrics for a new attack space.

Currently, some researchers propose using human validation for unrestricted adversarial examples. This approach requires a lot of human resources, and we cannot perform large-scale evaluations. The metrics we propose, matrix distance and SSIM value, are also not panaceas. As shown in [?], the SSIM can be unintendedly high when two images contain entirely different objects but similar backgrounds. In the world of unrestricted adversarial examples, attackers are not even required to generate adversarial examples from base images (e.g., by using GAN), so that we cannot apply full-referenced image quality assessment methods anymore. Thus studying non-reference image quality assessment methods is

an important research direction for measuring the quality of unrestricted adversarial images.

Why it is hard to build a robust model. In the real scenarios, the difficulty of building a robust model is from the worst-case attacks. From the attacker’s perspective, the attack is successful even if the attacker only finds one available adversarial example that compromises the model. Take our combination attack as an example, the attackers only need to find one available adversarial example from one of the multiple initial points. However, from the model designers’ perspective, they need to consider all the possible initial points and even all possible attack spaces.

7 CONCLUSION

Motivated by the fact that adversarial images are not necessarily close to the base images in L^p -norm distances, we explore adversarial attacks in another space—the image transformation space—instead of L^p -norm balls, with the help of color transformations and affine transformations. Also, we apply matrix distance and SSIM as alternative quality metrics to L^p -norms.

We evaluate our proposed attacks in a multiple configuration manner: three image segment settings for color transformation, three initial points for affine transformation, three datasets that are of different sizes and contents. With these parameters, we can explore an enormous transformation space. To find the best parameters, we thoroughly analyze and report how each parameter can affect the performance of our attack.

By evaluating our adversarial examples under different retraining defenses, we find several advantages of our transformation attack compared with C&W’s L^p attacks: it is harder to defend against by retraining and has higher transferability. We also show the significance of exploring different attack spaces: exploring different attack spaces makes it much more challenging for model designers to build a universally robust model that takes into account all the attack spaces. As a result, it would be necessary for model designers to define the model boundary carefully in the future.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. 1801751.

This research was partially sponsored by the Combat Capabilities Development Command Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-13-2-0045 (ARL Cyber Security CRA). The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Combat Capabilities Development Command Army Research Laboratory or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation here on.