Structure-Preserving Transformation: Generating Diverse and Transferable Adversarial Examples

Dan Peng[1], Zizhan Zheng[2], Xiaofeng Zhang[1]
[1]Harbin Institute of Technology (Shenzhen)
[2]Tulane University, USA
pengdan@stu.hit.edu.cn, zzheng3@tulane.edu, zhangxiaofeng@hit.edu.cn

Abstract

Adversarial examples are perturbed inputs designed to fool machine learning models. Most recent works on adversarial examples for image classification focus on directly modifying pixels with minor perturbations. A common requirement in all these works is that the malicious perturbations should be small enough (measured by an L_p norm for some p) so that they are imperceptible to humans. However, small perturbations can be unnecessarily restrictive and limit the diversity of adversarial examples generated. Further, an L_p norm based distance metric ignores important structure patterns hidden in images that are important to human perception. Consequently, even the minor perturbation introduced in recent works often makes the adversarial examples less natural to humans. More importantly, they often do not transfer well and are therefore less effective when attacking black-box models especially for those protected by a defense mechanism. In this paper, we propose a *structure-preserving transformation (SPT)* for generating natural and diverse adversarial examples with extremely high transferability. The key idea of our approach is to allow perceptible deviation in adversarial examples while keeping structure patterns that are central to a human classifier. Empirical results on the MNIST and the fashion-MNIST datasets show that adversarial examples generated by our approach can easily bypass strong adversarial training. Further, they transfer well to other target models with no loss or little loss of successful attack rate.

Introduction

Deep neural networks (DNNs) have achieved phenomenal success in computer vision by showing superber accuracy over traditional machine learning algorithms. However, recent works have demonstrated that DNNs are vulnerable to adversarial examples that are generated for malicious purposes by slightly twisting the original images (Szegedy et al. 2014; Goodfellow, Shlens, and Szegedy 2015). This observation has raised serious concerns on the robustness of the state-of-the-art DNNs and limited their applications in various security-sensitive applications. On the contrary, the adversarial examples can be aggregated to augment training datasets for learning a more robust DNN, known as adversarial training (Madry et al. 2018b).

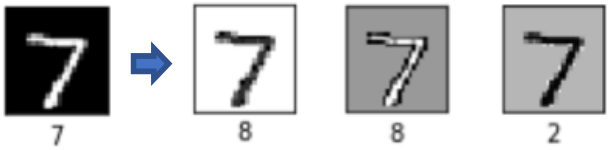

Figure 1: Adversarial examples generated by SPT. The first one is the original image selected from MNIST. The last three are adversarial examples crafted by SPT for attacking three different target classifiers, where they are classified as 8, 8, 2, respectively.

Generally speaking, adversarial examples can be any valid inputs to machine learning models that are intentionally designed to cause mistakes (Elsayed et al. 2018). For visual object recognition, we rely on human labelers to obtain the ground truth labels, which are unknown to the attacker before adversarial examples are generated. To bypass this dilemma, a common attack strategy is to start with a clean image where the ground truth label is already known and modify it so that the new image is visually similar to the original image while its output label differs from the ground truth label of the clean image. The intuition behind this approach is that by ensuring visual similarity, the two images are likely to share the same ground truth label.

A simple approach for ensuring visual similarity that has been intensively studied in the literature is to introduce small perturbations into pixels such that the distortion between the adversarial example and the original image is human imperceptible (Goodfellow, Shlens, and Szegedy 2015; Su, Vargas, and Sakurai 2018; Chen et al. 2018; Carlini and Wagner 2017b). However, the noisy and meaningless worst-case perturbations introduced by these approaches often make the generated adversarial images less natural (Goodfellow, Shlens, and Szegedy 2015). Moreover, perturbation-based adversarial examples exhibit low transferability, which makes them less effective for black-box attacks especially for those with a defense mechanism (Kurakin, Goodfellow, and Bengio 2016). There are some recent works (Zhao, Dua, and Singh 2018; Song et al. 2018) that relax the small perturbation requirement by utilizing generative networks (e.g., GANs). However, they attempt to generate adversarial examples with the identical distribu-

tion as the original dataset which limits their transferability and therefore the effectiveness in black-box attacks. To improve the robustness of DNNs against adversarial examples, a large number of defense mechanisms have been proposed in the literature (Kurakin, Goodfellow, and Bengio 2016; Madry et al. 2018b; Guo et al. 2018; Cisse et al. 2017). However, they are mainly designed to make classifiers more robust to small perturbations.

In this paper, we propose a *structure-preserving transformation (SPT)* for generating natural and diverse adversarial examples with extremely high transferability. Our approach is inspired by the fact that humans recognize object mainly from their shapes and structure patterns, supplemented by the color and brightness information. For example, consider the four images of digit 7 in Figure 1 where the first one is taken from MNIST while the next three are adversarial examples crafted by our algorithm against different target models. Despite the different background and foreground brightness in the four images, a human being can easily recognize the digit in all the cases since they share similar structure patterns. However, the last three images are mispredicted as digit 8 or 2 by DNNs. This inspires us to design adversarial examples that allow perceptible deviation (such as color and brightness) while keeping structure patterns that are central to a human classifier. We show that our structure-preserving adversarial examples are highly transferable with no loss or little loss of successful rate when applied to black-box attacks even when a defense mechanism is applied. This can be explained as follows. Traditional supervised learning models including DNNs all rely on the $i.i.d.$ assumption and become much less effective when test data and training data are drawn from different distributions. Current defense mechanisms do not improve that. Our approach generates adversarial examples that may have a totally different distribution from the original dataset and is therefore highly transferable.

An important limitation of previous attacks is that they can only generate a single type of adversarial examples. For perturbation-based attacks, a crafted adversarial example is a replica of the original image except for minor perturbation. That is, the attacker's action space (with respect to a single clean image) is a small neighborhood of the clean image under some metric. For generative-network-based attacks, they generate adversarial examples that obey the same distribution of the original images. That is, the attacker's action space is the set of images that follow the same distribution as the clean images. Although they have relaxed the small perturbation requirement, they still keep the same characteristic of the original images. For instance, the adversarial examples generated from MNIST images always have the same background color (black) and foreground color (white).

In contrast, the attacker's action space of our approach is much larger. It includes all the images that preserve the structure of the clean image. As shown in Figure 1, by preserving structure, our adversarial examples are natural, clean, legible to human beings. More importantly, for different target models or different initialization settings, the adversarial examples generated using our method are sufficiently diverse. They are distinct from the original images, and at the same time, they are also different from each other. For instance, the three adversarial examples shown in Figure 1 exhibit different background and foreground gray-level (intensity), which do not follow the distribution of the original dataset. This is why our approach can generate more diverse adversarial examples, which in turn makes our approach more difficult to defend. In particular, a defender that is adversarially trained with one type of adversarial examples is still vulnerable to other types of adversarial examples.

This work broadens the scope of adversarial machine learning by showing a new class of adversarial examples that follow different distributions from the training dataset while still being legible and natural to humans. Our study reveals the weakness of current defense mechanisms in the face of structure-preserving attacks that go beyond the small perturbation constraint.

Our main contributions can be summarized as follows.

- We propose a structure-preserving transformation (SPT) to generate natural and diverse adversarial examples. We show that an SPT can be easily trained with its learning parameters converging fast to the global optimum.
- We evaluate our approach on two datasets and demonstrate that our structure-preserving adversarial examples achieve extremely high transferability even when a defense mechanism is applied.
- We show that our structure-preserving adversarial examples can easily bypass strong adversarial training in the public MNIST challenge. The generated adversarial examples can dramatically reduce the accuracy of the white-box Madry network from 88.79% to 9.79% and the accuracy of the black-box Madry network from 92.76% to 9.80%.

Background

In this section, we briefly review recent studies on adversarial examples and defense mechanisms. We start with a simplified view of adversarial examples that has been broadly adopted in the literature and then discuss the more general definition used in this paper.

Traditional Adversarial Examples

In most previous works, an adversarial example is defined as a modified version of a clean image so that the two images are visually similar but the adversarial example is misclassified by the DNN (that is, its output label differs from the ground truth label of the clean image). A common requirement in many of existing studies is that the distortion between the adversarial example and the original image should be small enough so that the modification is human imperceptible. Also, the distortion is typically measured in an L_p norm for some p. In particular, the L_∞ norm measures the maximum variation of the distortions in pixel values (Goodfellow, Shlens, and Szegedy 2015), the L_1 and L_2 norms measure the total variations (Chen et al. 2018; Carlini and Wagner 2017b), while the L_0 norm corresponds to the total number of modified pixels (Su, Vargas, and

Sakurai 2018). Moreover, psychometric perceptual similarity measures that are more consistent with human perception have also been studied (Rozsa, Rudd, and Boult 2016).

Various techniques for crafting adversarial examples have been proposed. Most of them focus on adding small perturbations to the inputs, where several techniques have been studied. In gradient-based approaches such as the popular Fast Gradient Sign Method (FGSM) (Goodfellow, Shlens, and Szegedy 2015) and many of its variants, adversarial examples are generated by superimposing small perturbations along the gradient direction (with respect to the input image). More powerful attacks can be obtained by considering multi-step variants of FGSM such as Projected Gradient Descent (PGD) (Madry et al. 2018b). In optimization based approaches, an optimization problem is solved to identify the optimal perturbation over the original image that can lead to misclassification (Carlini and Wagner 2017b; Szegedy et al. 2014). More recently, generative networks including auto-encoders (Baluja and Fischer 2018) and generative adversarial networks (GANs) (Xiao et al. 2018) have been used to generate adversarial examples, where the small perturbation requirement is imposed on the image space and is included in the objective function to be trained.

General Adversarial Examples

Essentially, adversarial examples are inputs to machine learning models that are intentionally designed to make outputs differ from human labeled ground truth (Elsayed et al. 2018). There are two important points to be clarified in this definition. First, adversarial examples should be meaningful and recognizable to humans (Song et al. 2018). Although unrecognizable images can be generated to mislead classifiers (Nguyen, Yosinski, and Clune 2015), they are hard to label by humans to get the ground truth. Second, the small perturbation requirement is not a necessity. Instead, it is introduced to allow the adversarial examples share the ground truth labels of the original images to simplify the design of attacks. To cause a mistake in learning models, we need to ensure that adversarial examples are classified into labels that are different from their ground truth labels, while the latter is unknown until the adversarial examples have been generated and labelled by humans. To avoid this dilemma, traditional approaches introduced the small perturbation requirement to ensure the *visual similarity* between adversarial examples and clean images so that they are likely to share the ground truth labels. In contrast, we propose to use structure-similarity to model *semantic similarity* in this paper. Both perturbation-based and structure-preserving adversarial examples are consistent with the general definition of adversarial examples.

Defense Techniques

To improve the robustness of machine learning models against adversarial examples, a number of defense mechanisms have been proposed in the literature. Some approaches try to merge adversarial examples into the training dataset to decrease the model susceptibility to malicious attacks (Kurakin, Goodfellow, and Bengio 2016; Madry et al. 2018b). Researcher have also considered techniques for removing adversarial perturbations during the test stage (Guo et al. 2018) and modifying network structures to improve the model robustness (Cisse et al. 2017). However, most defense methods are ineffective to the newly proposed attacks (Athalye, Carlini, and Wagner 2018; Carlini and Wagner 2017a). Only a few state-of-the-art defense models have demonstrated their robustness to adversarial examples (Madry et al. 2018b; Samangouei, Kabkab, and Chellappa 2018), which, however, are mainly designed for perturbation-based attacks. To shed light on the weakness of existing defense mechanisms, we have evaluated our structure-preserving attacks against adversarial training with PGD (Madry et al. 2018b), which has been demonstrated to be the most effective defense method on the MNIST dataset (Athalye, Carlini, and Wagner 2018).

Structure-Preserving Attacks to DNNs

The proposed approach is called **S**tructure-**P**reserving **T**ransformation (SPT), which attempts to transform the input images into adversarial examples against a deep neural network while keeping the structure patterns of the original images. SPT can be either untargeted or targeted and can be used for both white-box and black-box attacks. In this paper, we focus on untargeted white-box and black-box attacks.

Let $\mathbf{x} \in \mathcal{X}$ be an unlabeled image and $\mathcal{Y}$ the set of class labels. Let f be the target network that outputs a probability distribution across the class labels. The target model assigns the most possible class $C(\mathbf{x}) = \arg\max_{k \in \mathcal{Y}}[f(\mathbf{x})]_k$ to the input image $\mathbf{x}$. Let $\mathbf{x}'$ denote the crafted adversarial image of $\mathbf{x}$ using the proposed transformation. The general transformation function $g_{f,\boldsymbol{\theta}}$ can be defined as

$$g_{f,\boldsymbol{\theta}} : \mathbf{x} \in \mathcal{X} \rightarrow \mathbf{x}', \tag{1}$$

where $\boldsymbol{\theta}$ is the parameter vector of function $g_{f,\boldsymbol{\theta}}$. Given the ground truth class label of the original image $\mathbf{x}$, $C_{\text{true}}(\mathbf{x})$, our approach tries to craft an adversarial image (transformed image $\mathbf{x}'$) to mislead classifier f so that $C(\mathbf{x}') \neq C_{\text{true}}(\mathbf{x})$.

Transformations in spatial domains are commonly used in image enhancement where an image is transformed to a new image for better visual quality by directly manipulating pixels, which can be formalized as follows.

$$\mathbf{x}'(m,n) = T_{m,n}[\mathbf{x}(m,n)] \tag{2}$$

where $\mathbf{x}(m,n)$ (resp. $\mathbf{x}'(m,n)$) are the intensity of the input (resp. output) image at the coordinate (m,n), and $T_{m,n}$ is a transformation on $\mathbf{x}$ that depends on (m,n). Note that in general, the transformed value of any pixel may depend on all the pixels in some neighborhood of it, e.g., considering the convolution and pooling operations in CNN (Krizhevsky, Sutskever, and Hinton 2012). The simplest form of T is when the neighborhood of a pixel is a singleton, that is, it only contains the pixel itself. In this case, $\mathbf{x}'(m,n)$ depends only on $\mathbf{x}(m,n)$ and T becomes a gray-level transformation that depends only on pixel values and is independent of their coordinates. As a special case of such singleton-based transformations, power functions are commonly used in image enhancement to allow a variety of devices to print and display images (Kim 1999). In this paper, we define the transformation function $g_{f,\boldsymbol{\theta}}$ as a linear combination of multiple

power functions. For simplicity, we use $g(.)$ to denote $g_{f,\boldsymbol{\theta}}$ in the rest of this paper. Accordingly, the general form of $g(.)$ is given as,

$$g(\mathbf{x}) = \text{sigmoid}(\sum_i w_i \mathbf{x}^{\gamma_i}) \tag{3}$$

where γ_i is the exponent of the i-th basis power function. In general, these exponents can be either learned from data or chosen based on domain knowledge and fixed in advance. For simplicity, we manually choose the values of γ_i's in this paper. The weight w_i is a scalar shared with all the elements in $\mathbf{x}^{\gamma_i}$. These weights $\{w_i\}$ are the parameters to be learned. That is, $\boldsymbol{\theta} = \{w_i\}$. The sigmoid function is adopted to further guarantee the range of output falls into [0,1].

A notable property of any singleton-based transformation including ours is that all the pixels with the same gray level in the original image have the same gray level in the transformed image. Further, our transformation is typically injective, i.e., it ensures that pixels with different gray levels in the original image typically have different gray levels in the transformed image. That is, such a transformation keeps the *structure patterns* of images, which is formally defined as follows.

Definition 1. ***Structure Pattern*** *The set of all pixels with the same grey-level in an image is called a structure pattern of the image. Symbolically, given an image* $\mathbf{x}$*, the set* $\{(m,n)|\mathbf{x}(m,n) = c\}$ *is called a structure pattern with gray level* c*.*

Training

SPT is trained together with the target model by adding an extra layer in front of the target network. For untargetd attacks, our objective is to find the parameters $\boldsymbol{\theta}$ so that $C(\mathbf{x}') \neq C_{\text{true}}(\mathbf{x})$. To improve the chance of successful attacks, we aim to find $\boldsymbol{\theta}$ so that the distance between the predicted logits (that is, the output of the classifier $f(\mathbf{x})$) and the one-hot encoding of ground truth class on all the training data is maximized. We consider the following objective function:

$$\underset{\boldsymbol{\theta}}{\text{argmax}} \sum_{\mathbf{x}\in\mathcal{X}} L(f(g(\mathbf{x})), l_{\text{true}}(\mathbf{x}))) - \alpha \sum_i w_i^2, \tag{4}$$

where L is a loss function that measures the difference between the output logit $f(g(\mathbf{x}))$ when the target model f is applied to the crafted adversarial example $g(\mathbf{x})$ and the one-hot encoding of ground truth class of the original image. Moreover, an L_2 regularization term is introduced in (4), which is used to penalize intensely dramatic image transformations that may deteriorate the visual quality of images. α is a scalar parameter that controls the strength of regularization. Unlike most other approaches, the generated $\mathbf{x}'$ is not restricted to be similar enough to the original $\mathbf{x}$. Instead, our approach guarantees the perceptual similarity in human vision in terms of structure patterns. Hence, the corresponding objective function could be defined in a concise manner and is thus easy to converge.

In this paper, we define the loss function L as the cross entropy between the one-hot encoding ground truth class $l_{\text{true}}(\mathbf{x})$ of the original image and the predicted logit $f(g(\mathbf{x}))$ of the corresponding adversarial image. Formally,

$$L(f(g(\mathbf{x})), l_{\text{true}}(\mathbf{x})) = -l_{\text{true}}(\mathbf{x})^T \log(f(g(\mathbf{x}))). \tag{5}$$

where T denotes the transpose of a vector. For untargted attacks, we then solve the following minimization problem to find the parameters θ of $g(\cdot)$:

$$\underset{\boldsymbol{\theta}}{\text{argmin}} \sum_{\mathbf{x}\in\mathcal{X}} l_{\text{true}}(\mathbf{x})^T \log(f(g(\mathbf{x}))) + \alpha \sum_i w_i^2. \tag{6}$$

Although we focus on untargeted attacks in this work, our approach extends to targeted attacks by considering a slightly different objective function:

$$\underset{\boldsymbol{\theta}}{\text{argmin}} \sum_{\mathbf{x}\in\mathcal{X}} L(f(g(\mathbf{x})), l_{\text{target}}(\mathbf{x}'))) + \alpha \sum_i w_i^2, \tag{7}$$

where $l_{\text{target}}(\mathbf{x}')$ is the vectorized representation of target class we want to mispredict, which can be simply defined as the one-hot encoding of the target class similar to the untargeted case. In addition to this simple encoding scheme and the cross-entropy based loss function, other encoding techniques and loss functions (Baluja and Fischer 2018; Carlini and Wagner 2017b) can also be applied to our framework for both untargeted attacks and target attacks, which will be studied in the future.

To find the learning parameters $\{w_i\}$, we solve the above optimization problem using the Adam method (Kingma and Ba 2014) with a learning rate of 10^{-4}. Due to the good convergence property of the cross entropy loss function, one epoch training is sufficient for all the training data. Moreover, because a small number of parameters $\{w_i\}$ (about 10 parameters) need to be optimized, it only takes a few minutes to train.

Inference

After training a Structure-Preserving Transformation, we generate adversarial examples by performing the trained SPT on the original images. The process is fast and only involves computing a linear combination of power functions and does not require any information of the target models.

Experiment Results

To evaluate the performance of SPT, we compare it with three baseline attack algorithms, FGSM (Goodfellow, Shlens, and Szegedy 2015), PGD (Madry et al. 2018b), and C&W (Carlini and Wagner 2017b) on two popular image classification datasets, MNIST (LeCun et al. 1998) and Fashion-MNIST (F-MNIST) (Xiao, Rasul, and Vollgraf 2017). Both datasets consist of 60,000 training images and 10,000 testing images from 10 classes. As in most related works, classification accuracy is used as the evaluation criterion. A stronger attack method has a lower classification accuracy. Moreover, the attack methods are evaluated both on the original models as well as when a defense mechanism is applied. In this paper, adversarial training with PGD is chosen as the defense method, which has been shown to be the

Table 1: **White-box attacks**. Classification accuracy of different models on the MNIST (top) and F-MNIST (bottom) datasets. In each row, the best result is highlighted in bold and the second-best result is underlined.

Defense	Target	No Attack	FGSM	PGD	C&W	**SPT**
No Defense	C_p	99.02%	9.28%	**0.00%**	**0.00%**	9.74%
	C_{a0}	98.83%	5.55%	**0.00%**	**0.00%**	9.75%
	C_{a1}	98.73%	7.18%	0.03%	**0.00%**	10.52%
	C_{a2}	98.33%	8.25%	0.09%	**0.00%**	8.93%
	C_{a3}	98.58%	11.44%	**0.00%**	**0.00%**	9.91%
PGD Adv. Tr. (Tr. Acc) $\epsilon = 0.3$ $\alpha = 0.01$	C_p	98.08%	93.24%	88.14%	32.00%	**13.47%**
	C_{a0}	97.65%	84.51%	68.07%	**5.00%**	5.10%
	C_{a1}	98.13%	89.01%	73.92%	6.00%	**4.39%**
	C_{a2}	98.20%	87.78%	73.16%	**4.00%**	8.98%
	C_{a3}	96.90%	92.54%	87.15%	45.00%	**7.32%**
No Defense	C_p	91.18%	7.73%	**0.00%**	**0.00%**	10.00%
	C_{a0}	91.35%	7.42%	**0.00%**	**0.00%**	9.18%
	C_{a1}	90.67%	7.88%	**0.00%**	**0.00%**	10.45%
	C_{a2}	89.79%	9.61%	0.08%	**0.00%**	10.04%
	C_{a3}	91.24%	4.42%	**0.00%**	**0.00%**	10.06%
PGD Adv. Tr. (Tr. Acc) $\epsilon = 0.3$ $\alpha = 0.01$	C_p	74.80%	68.92%	55.40%	69.00%	**10.00%**
	C_{a0}	72.40%	63.17%	50.54%	78.00%	**7.24%**
	C_{a1}	73.27%	65.18%	51.75%	77.00%	**9.99%**
	C_{a2}	77.29%	74.52%	66.41%	78.00%	**5.71%**
	C_{a3}	71.47%	64.04%	49.34%	66.00%	**9.45%**

most effective defense method on the MNIST dataset (Athalye, Carlini, and Wagner 2018). Evaluations in both white-box and black-box attack settings show that the proposed SPT can generate more diverse and naturally distorted adversarial images with high transferability.

Experiment Settings

Baseline Attacks: As aforementioned, three baseline attack approaches are implemented for performance comparison. The Fast Gradient Sign Method (**FGSM**) superimposes small gradient-based perturbation to the original images to mislead the classification models (Goodfellow, Shlens, and Szegedy 2015); The Projected Gradient Descent (**PGD**) attack (Madry et al. 2018b) is a variant of iterative FGSM; **C&W** (Carlini and Wagner 2017b) generates adversarial examples by solving L_2 norm based optimization problem to find the adversarial examples that drop the accuracy of target model as much as possible.

Parameter settings: For SPT, we consider a linear combination of 11 power functions with exponents γ equal to $0.04, 0.10, 0.20, 0.40, 0.67, 1.0, 1.5, 2.5, 5.0, 10.0, 25.0$, respectively. The larger the γ is, the darker the transformed image will be. The intuition behind choosing these exponents is further explained in the Appendix. Since most pixel values in the MNIST dataset are close to black or white, even a large γ will not significantly distort an image, the coefficient α of the penalty term in (3) is to set to 0 for experiments on the MNIST dataset. We set $\alpha = 0.6$ for experiments on fashion-MNIST dataset to avoid low image contrast for large γ values.

Both C&W and FGSM are implemented using the public CleverHans package with the same configurations (Papernot et al. 2016). PGD is implemented using the source code from the MNIST challenge (Madry et al. 2018a).

Attack modes: Two attack modes are considered in the evaluations. i.e., ***white-box attack*** and ***black-box attack***. For ***white-box attack***, details of the target model are publicly known to the malicious parties including model structure and learned weights, datasets, and the adopted defense mechanisms. For ***black-box attack***, only the training and testing datasets could be manipulated but the rest keeps unknown to the adversaries.

Evaluation Results for White-Box Attacks

In this section, we present the experimental results in the white-box setting. Following the same network architecture in (Baluja and Fischer 2018), we train five networks as target models on the MNIST training dataset and F-MNIST training dataset, respectively. Each network is a combination of convolutional and fully connected layers and details are provided in the Appendix. In Table 1, we report the classification accuracy of these models under various attack-defense configurations. On the attacker side, we consider five cases: when there is no attack and when one of the four attack methods is applied. On the defender side, two cases are considered: when there is no defense and when PGD-based adversarial training is applied. We observe that SPT achieves low accuracy (about 10% or lower) across all the scenarios. When no defense is applied, both C&W and PGD achieve very low accuracy (close to 0), while SPT is still comparable to FGSM. When PGD-based adversarial training is applied, SPT obtains overwhelming low accuracy than all the baseline attacks. These results demonstrate the effectiveness of SPT even in the presence of strong defense.

We further evaluate SPT adversarial examples on the most

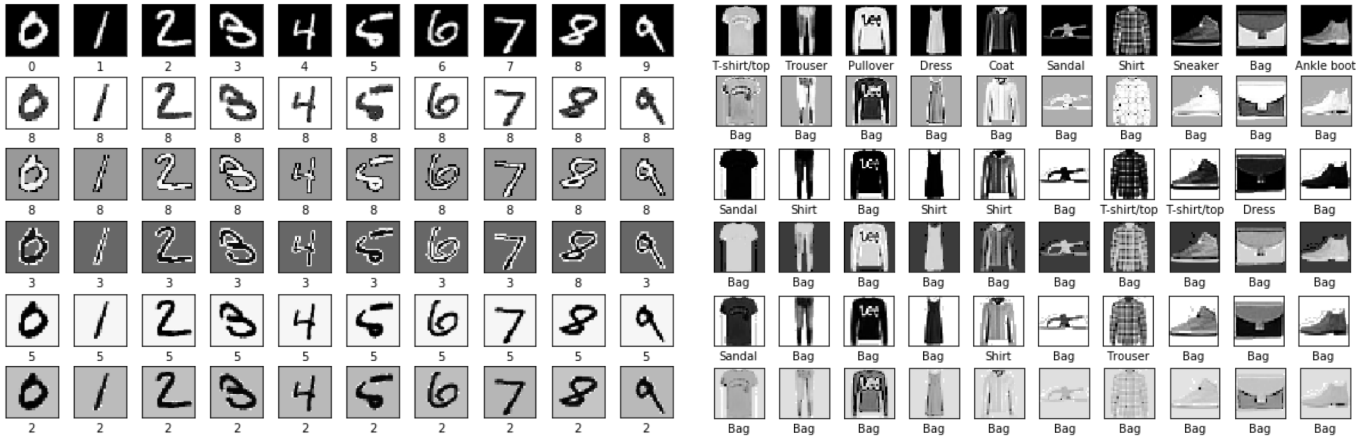

Figure 2: Visual illustration of adversarial examples generated by SPT. The first row shows the original images selected from MNIST (left) and F-MNIST (right), respectively. The other five rows show the adversarial examples crafted by SPT for each of the five classifiers, respectively.

competitive public MNIST challenge where researchers try their best to attack the well-trained robust Madry networks (Madry et al. 2018a). Remarkably, SPT drops the accuracy of the white-box Madry network to 9.79% , which is far below the current best result of 88.79% on the leaderboard (Zheng, Chen, and Ren 2018). We acknowledge that the comparison is a bit unfair since SPT does not have the small perturbation requirement while the public results in the leaderboard do. However, the observed big gap does indicate the strength of SPT.

Black-Box Attacks and Transferability

In the section, we present experimental results in the black-box setting. The same five target networks as in the white-box setting are used. Among them, Classifier C_p is used as the substitute model. We first generate adversarial examples against C_p and then test them on the other four target networks. As expected, SPT consistently obtains low accuracy with or without defense. Thus, it has excellent transferability and is extremely effective in the black-box setting. FGSM and PGD have rather low accuracy when no defense is applied due to their moderate transferability. However, when defense is applied, both of them exhibit poor transferability. C&W exhibits poor transferability in all the scenarios due to the excessive optimization applied (Kurakin, Goodfellow, and Bengio 2016). Further, SPT reduces the accuracy of the black-box Madry network to 9.80% in the MNIST challenge, while the current best result on the leaderboard is 92.76% (Xiao et al. 2018).

Illustration of Adversarial Examples

Figure 2 shows the adversarial examples generated by SPT where the first row gives the original images and each other row shows the generated adversarial examples against C_p, C_{a0}, C_{a1}, C_{a2}, and C_{a3}, respectively. We observe that adversarial examples on the same column (corresponding to different target models) have distinct colors (brightness) but they all keep the same structure as the original image. In fact, SPT generates diverse adversarial examples not only for different target classifiers but also for different initialization settings (see Figure 4 in the in the Appendix).

The diversity of adversarial examples poses a challenge to defense against SPT as it is difficult to resist all kinds of SPT adversarial examples. Moreover, all the adversarial examples are clean and legible to humans. For MNIST, the generated digits keep the same handwriting characteristics, like the same person writing with different-colored pens and papers. For F-MNIST, the generated clothes are of the same style while they have different colors and textures.

Observations and Discussions

We make the following observations from the above results.

First, DNNs even with defense are vulnerable to adversarial examples drawn from a distribution that is different from the training dataset. That is the main reason why the proposed SPT adversarial examples exhibit high attack success rate and high transferability.

Second, as shown in Figure 2, adversarial examples in the same column preserve the same structure pattern, but they are classified into different classes by the target DNNs. This indicates that DNNs are not very good at learning shapes and structures especially when images are not drawn from the same distribution as the training data.

Third, Figure 2 shows that adversarial examples on the same row are often classified into the same class. The statistics of the prediction results across labels given in Table 3 (in the Appendix) further confirms this observation. We note that these examples correspond to different original images against the same target model. Moreover, they exhibit different structures but have similar brightness. We therefore speculate that in make predictions for image drawing from different distribution with training data, DNNs mainly rely on the distribution of pixel values rather than detailed structure patterns.

We hope our observations bring new insights into how DNNs work.

Table 2: **Black-box attacks**. Classification accuracy on the MNIST (top) and F-MNIST (bottom) datasets where C_p is used as the substitute model and the other models are target models. In each row, the best result is highlighted in bold and the second-best result is underlined.

Defense	Target	No Attack	FGSM	PGD	C&W	**SPT**
No Defense	C_p	99.02%*	9.42%*	0.00%*	0.00%*	9.74%*
	C_{a0}	98.83%	<u>27.95%</u>	29.24%	89.00%	**9.74%**
	C_{a1}	98.73%	<u>41.57%</u>	45.81%	88.00%	**11.72%**
	C_{a2}	98.33%	<u>30.35%</u>	35.74%	89.00%	**8.92%**
	C_{a3}	98.58%	<u>28.78%</u>	29.51%	87.00%	**10.17%**
PGD Adv. Tr. (Tr. Acc) $\epsilon = 0.3$ $\alpha = 0.01$	C_p	98.08%	94.50%	95.41%	<u>90.00%</u>	**5.23%**
	C_{a0}	97.65%	<u>87.38%</u>	90.33%	89.00%	**6.97%**
	C_{a1}	98.13%	90.19%	92.79%	<u>90.00%</u>	**7.58%**
	C_{a2}	98.20%	<u>85.67%</u>	91.30%	90.00%	**8.60%**
	C_{a3}	96.90%	92.86%	93.77%	<u>90.00%</u>	**6.45%**
No Defense	C_p	91.18%*	7.73%*	0.00%*	0.00%*	10.00%*
	C_{a0}	91.35%	11.19%	**0.04%**	88.00%	<u>10.02%</u>
	C_{a1}	90.67%	11.23%	**0.58%**	90.00%	<u>10.00%</u>
	C_{a2}	89.79%	15.67%	**3.01%**	89.00%	<u>13.36%</u>
	C_{a3}	91.24%	11.72%	**1.68%**	90.00%	<u>10.00%</u>
PGD Adv. Tr. (Tr. Acc) $\epsilon = 0.3$ $\alpha = 0.01$	C_p	74.80%	<u>71.65%</u>	71.69%	90.00%	**9.99%**
	C_{a0}	72.40%	<u>67.58%</u>	67.78%	90.00%	**10.00%**
	C_{a1}	73.27%	<u>68.54%</u>	68.69%	90.00%	**9.99%**
	C_{a2}	77.29%	<u>72.77%</u>	73.28%	90.00%	**10.00%**
	C_{a3}	71.47%	69.50%	<u>69.36%</u>	90.00%	**10.00%**

Related Works

There are a few existing works that consider generating general adversarial example beyond the small perturbation requirement. Most of them are built upon the idea that the ground truth of an original image can propagate to adversarial examples by preserving the visual similarity between them. In (Zhao, Dua, and Singh 2018; Song et al. 2018), this is achieved by limiting the change of a latent representation of images learned using GANs. Another approach is to make perturbations resemble real yet inconspicuous objects to hide them from humans (Evtimov et al. 2017; Brown et al. 2017).

The intuition behind the recent work (Hosseini and Poovendran 2018) is somewhat similar to ours, where they propose the concept of "shape bias" and state that humans prefer to categorize objects according to their shapes rather than colors (Landau, Smith, and Jones 1988). However, the technical approaches of the two works are very different. In particular, the approach in (Hosseini and Poovendran 2018) focuses on color images and the main idea is convert the color space from RGB to HSV. To preserve the shape information, they keep the brightness component unchanged and change color components (i.e., saturation and hue) with the same amount for all the pixels. In contrast, we change pixel values based on structure patterns, where the amount of change is different for pixels in different structure patterns. Moreover, our transformation is controlled by multiple trainable parameters. These two differences are the key to obtain more diverse adversarial examples compared to the approach in (Hosseini and Poovendran 2018). We also note that although our approach focuses on grey-scale images, the key idea applies to color images as well.

Conclusion and Further Work

In this paper, we propose the technique of structure-preserving transformation (SPT) to generate natural and diverse adversarial examples with high transferability. SPT keeps the semantic similarity between adversarial examples and original images by preserving the structure patterns between them. This new approach allows the ground truth of original images to be shared with adversarial examples without imposing the small perturbation requirement. Empirical results on the MNIST and fashion-MNIST datasets show that SPT can generate adversarial examples that are natural to humans while being sufficiently diverse. Further, the adversarial examples significantly reduce the classification accuracy of target models whether defenses are applied or not. In particular, we show that SPT can easily bypass PGD-based adversarial training. Moreover, SPT adversarial examples can transfer across different models with little or no loss of successful attack rates. The high successful attack rates and outstanding transferability stem from the key property that SPT adversarial examples follow different distributions from the training data of target models.

Although we focus on untargeted attacks and gray-scale images in the paper, the core idea of SPT can extend to targeted attacks and color images. In our preliminary experiments for color images, the high successful attack rate is at the expense of images being less discriminable and authentic. How to ensure the quality of images while maintaining a high successful attack rate for color images is an important challenge that we will investigate in our future work. In addition, we will test the performance of SPT-based adversarial training and study the performance of SPT attacks under other defense methods that do not rely on ℓ_p attacks.

References

[Athalye, Carlini, and Wagner 2018] Athalye, A.; Carlini, N.; and Wagner, D. A. 2018. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *ICML Workshop*.

[Baluja and Fischer 2018] Baluja, S., and Fischer, I. 2018. Learning to attack: Adversarial transformation networks. In *AAAI*.

[Brown et al. 2017] Brown, T. B.; Mané, D.; Roy, A.; Abadi, M.; and Gilmer, J. 2017. Adversarial patch. *arXiv preprint arXiv:1712.09665*.

[Carlini and Wagner 2017a] Carlini, N., and Wagner, D. 2017a. Adversarial examples are not easily detected: Bypassing ten detection methods. In *AISec Workshop*.

[Carlini and Wagner 2017b] Carlini, N., and Wagner, D. 2017b. Towards evaluating the robustness of neural networks. In *S&P*.

[Chen et al. 2018] Chen, P.; Sharma, Y.; Zhang, H.; Yi, J.; and Hsieh, C. 2018. EAD: elastic-net attacks to deep neural networks via adversarial examples. In *AAAI*.

[Cisse et al. 2017] Cisse, M.; Bojanowski, P.; Grave, E.; Dauphin, Y.; and Usunier, N. 2017. Parseval networks: Improving robustness to adversarial examples. In *ICML*.

[Elsayed et al. 2018] Elsayed, G. F.; Shankar, S.; Cheung, B.; Papernot, N.; Kurakin, A.; Goodfellow, I.; and Sohl-Dickstein, J. 2018. Adversarial examples that fool both human and computer vision. *arXiv preprint arXiv:1802.08195*.

[Evtimov et al. 2017] Evtimov, I.; Eykholt, K.; Fernandes, E.; Kohno, T.; Li, B.; Prakash, A.; Rahmati, A.; and Song, D. 2017. Robust physical-world attacks on deep learning models. *arXiv preprint arXiv:1707.08945* 1.

[Goodfellow, Shlens, and Szegedy 2015] Goodfellow, I.; Shlens, J.; and Szegedy, C. 2015. Explaining and harnessing adversarial examples. In *ICLR*.

[Guo et al. 2018] Guo, C.; Rana, M.; Cisse, M.; and van der Maaten, L. 2018. Countering adversarial images using input transformations. In *ICLR*.

[Hosseini and Poovendran 2018] Hosseini, H., and Poovendran, R. 2018. Semantic adversarial examples. *arXiv preprint arXiv:1804.00499*.

[Kim 1999] Kim, J.-g. 1999. Color correction device for correcting color distortion and gamma characteristic. US Patent 5,949,496.

[Kingma and Ba 2014] Kingma, D. P., and Ba, J. L. 2014. Adam: Amethod for stochastic optimization. In *ICLR*.

[Krizhevsky, Sutskever, and Hinton 2012] Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NIPS*.

[Kurakin, Goodfellow, and Bengio 2016] Kurakin, A.; Goodfellow, I.; and Bengio, S. 2016. Adversarial machine learning at scale. *arXiv preprint arXiv:1611.01236*.

[Landau, Smith, and Jones 1988] Landau, B.; Smith, L. B.; and Jones, S. S. 1988. The importance of shape in early lexical learning. *Cognitive Development* 3:299–321.

[LeCun et al. 1998] LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

[Madry et al. 2018a] Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018a. MNIST Adversarial Examples Challenge. `https://github.com/MadryLab/mnist_challenge`. Accessed: 2018-09-01.

[Madry et al. 2018b] Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018b. Towards deep learning models resistant to adversarial attacks. In *ICLR*.

[Nguyen, Yosinski, and Clune 2015] Nguyen, A.; Yosinski, J.; and Clune, J. 2015. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*.

[Papernot et al. 2016] Papernot, N.; Carlini, N.; Goodfellow, I.; Feinman, R.; Faghri, F.; Matyasko, A.; Hambardzumyan, K.; Juang, Y.-L.; Kurakin, A.; Sheatsley, R.; et al. 2016. cleverhans v2. 0.0: an adversarial machine learning library. *arXiv preprint arXiv:1610.00768*.

[Rozsa, Rudd, and Boult 2016] Rozsa, A.; Rudd, E. M.; and Boult, T. E. 2016. Adversarial diversity and hard positive generation. In *CVPR Workshop*.

[Samangouei, Kabkab, and Chellappa 2018] Samangouei, P.; Kabkab, M.; and Chellappa, R. 2018. Defense-GAN: Protecting classifiers against adversarial attacks using generative models. In *ICLR*.

[Song et al. 2018] Song, Y.; Shu, R.; Kushman, N.; and Ermon, S. 2018. Constructing unrestricted adversarial examples with generative models. In *NeurIPS*.

[Su, Vargas, and Sakurai 2018] Su, J.; Vargas, D. V.; and Sakurai, K. 2018. Attacking convolutional neural network using differential evolution. *arXiv preprint arXiv:1804.07062*.

[Szegedy et al. 2014] Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; and Fergus, R. 2014. Intriguing properties of neural networks. In *ICLR*.

[Xiao et al. 2018] Xiao, C.; Li, B.; Zhu, J.-Y.; He, W.; Liu, M.; and Song, D. 2018. Generating adversarial examples with adversarial networks. *arXiv preprint arXiv:1801.02610*.

[Xiao, Rasul, and Vollgraf 2017] Xiao, H.; Rasul, K.; and Vollgraf, R. 2017. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*.

[Zhao, Dua, and Singh 2018] Zhao, Z.; Dua, D.; and Singh, S. 2018. Generating natural adversarial examples. In *ICLR*.

[Zheng, Chen, and Ren 2018] Zheng, T.; Chen, C.; and Ren, K. 2018. Distributionally adversarial attack. *arXiv preprint arXiv:1808.05537*.

Appendix

The Choice of Exponents of Power Functions

In the experiments, the exponents γ of the power functions in SPT are empirically set to $\{0.04, 0.10, 0.20, 0.40, 0.67, 1.0, 1.5, 2.5, 5.0, 10.0, 25.0\}$. The underlying reason is that these basic functions could well reflect different transformation intensities. (see Figure 3). A power function with a larger γ will generate a darker image and vise versa.

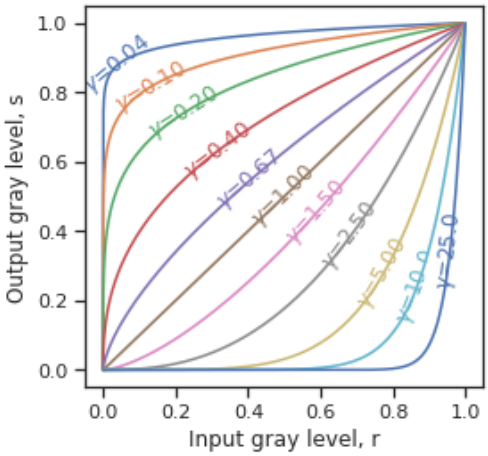

Figure 3: the equation $s = cr^{\gamma}$ for various values of γ

Neural Network Architectures

Details of the five network architectures used in the paper can be found in Table 4. In the table, Conv(m, k, k, s) denotes a convolutional layer with m feature maps, filter size $k \times k$, and stride s, Max Pool(n, n) denotes a max pooling layer with filters of size $n \times n$, FC(m) denotes a fully-connected layer with m outputs, and ReLU denotes the Rectified Linear Unit activation function.

Prediction Statistics for Adversarial Examples

The prediction statistics for white-box SPT adversarial examples are reported in Table 3. Each row in the table records the percentage of SPT adversarial examples that are classified into each label by a particular target model. The result shows that most adversarial examples are classified into the same class by any target model.

SPT Adversarial Examples for Different Initialization Settings

Figure 4 plots the generated SPT adversarial examples against white-box C_p with different initial weights. Obviously, the generated SPT adversarial examples vary a lot with different initialization settings.

Table 3: Prediction statistics of different models for SPT adversarial examples on the MNIST (top) and F-MNIST (bottom) datasets

Target	0	1	2	3	4	5	6	7	8	9
C_p	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	**100%**	0.00%
C_{a0}	0.00%	0.00%	0.00%	0.00%	0.00%	0.01%	0.00%	0.00%	**99.99%**	0.00%
C_{a1}	0.00%	0.00%	0.00%	**89.15%**	0.00%	0.01%	0.00%	0.00%	**10.85%**	0.00%
C_{a2}	0.06%	0.00%	0.00%	0.00%	0.00%	**99.93%**	0.00%	0.00%	0.01%	0.00%
C_{a3}	0.00%	0.00%	**99.13%**	0.87%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%

Target	T-shirt/top	Trouser	Pullover	Dress	Coat	Sandal	Shirt	Sneaker	Bag	Ankle boot
C_p	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	**100%**	0.00%
C_{a0}	**37.03%**	4.54%	1.93%	0.71%	0.00%	9.28%	15.3%	0.01%	**26.11%**	5.09%
C_{a1}	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.02%	0.00%	**99.49%**	0.49%
C_{a2}	2.93%	15.49%	0.57%	0.08%	0.00%	2.94%	11.89%	0.00%	**64.63%**	1.47%
C_{a3}	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.26%	0.00%	**99.74%**	0.00%

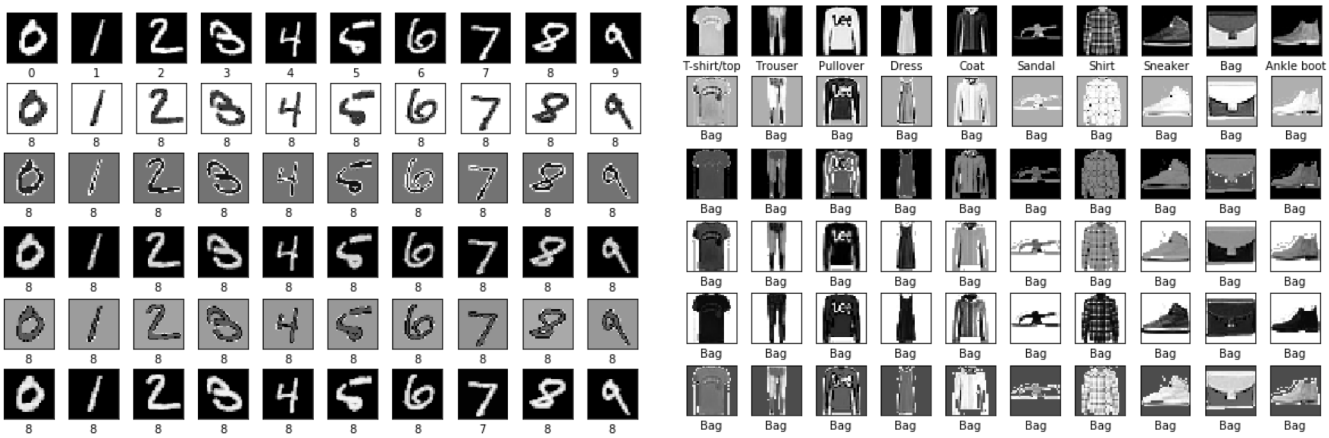

Figure 4: Visual illustration of adversarial examples generated by SPT with different initialization settings. The first row shows the original images selected from MNIST (left) and F-MNIST (right), respectively. The other five rows show the adversarial examples crafted by SPT for different initialization settings, respectively.

Table 4: Neural network architectures for classifiers used in experiments. C_{a0} shares the same architecture as C_p, and only differs in the initialization of the weights.

C_p , C_{a0}	C_{a1}	C_{a2}	C_{a3}
Conv(32, 5, 5, 1)	Conv(32, 4, 4, 1)	Conv(32, 3, 3, 1)	Conv(32, 3, 3, 1)
Relu	Relu	Relu	Relu
Max Pool(2, 2)	Max Pool(2, 2)	Max Pool(2, 2)	Max Pool(2, 2)
Conv(64, 5, 5, 1)	Conv(32, 4, 4, 1)	Conv(32, 3, 3, 1)	FC(1024)
Relu	Relu	Relu	Relu
Max Pool(2, 2)	Max Pool(2, 2)	Max Pool(2, 2)	Fc(512)
FC(1024)	Conv(64, 4, 4, 1)	Conv(64, 3, 3, 1)	Relu
FC(10) + Softmax	FC(1024)	FC(1024)	Fc(10) + Softmax
	FC(10) + Softmax	FC(10) + Softmax	