Naturalistic Physical Adversarial Patch for Object Detectors

Yu-Chih-Tuan Hu[1,2] Bo-Han Kung[1] Daniel Stanley Tan[1]
Jun-Cheng Chen[1] Kai-Lung Hua[2] Wen-Huang Cheng[3]
[1]Research Center for Information Technology Innovation, Academia Sinica
[2]National Taiwan University of Science and Technology [3]National Yangming Jiaotong University

Abstract

*Most prior works on physical adversarial attacks mainly focus on the attack performance but seldom enforce any restrictions over the appearance of the generated adversarial patches. This leads to conspicuous and attention-grabbing patterns for the generated patches which can be easily identified by humans. To address this issue, we propose a method to craft physical adversarial patches for object detectors by leveraging the learned image manifold of a pretrained generative adversarial network (GAN) (*e.g.*, BigGAN and StyleGAN) upon real-world images. Through sampling the optimal image from the GAN, our method can generate natural looking adversarial patches while maintaining high attack performance. With extensive experiments on both digital and physical domains and several independent subjective surveys, the results show that our proposed method produces significantly more realistic and natural looking patches than several state-of-the-art baselines while achieving competitive attack performance.* [1]

1. Introduction

With the advancement of deep learning technologies, modern computer vision models can achieve comparable or even surpassing human performance on tasks such as face detection [10] and face recognition [10]. Although these technologies bring convenience to humans by automating daily routine tasks, they also significantly hurt our privacy since malicious people could easily utilize them to automatically collect private and sensitive personal information. To address this issue, some researchers propose to protect people from these threats by leveraging the adversarial examples which can be used to fool deep learning systems by adding small or imperceptible perturbations to system inputs. Adversarial attacks for deep learning systems can be categorized by two settings: (1) digital attacks, where deep learning models takes the digital attack images as inputs and (2) physical attacks, where the models take attack inputs that are retaken by a camera.

Figure 1. It shows the crafted adversarial patch generated by the proposed approach along with others by recent methods (a) [47] (b) [45] (c) [42] (d) [19] (e) ours. Our patch is more natural looking and less conspicuous than others so it is harder to human observers to identify it.

In this work, we focus on the second category due to its practical use in the real-world setting against the surveillance of various indoor and outdoor cameras around the world. Adversarial patch is one of most effective physical adversarial examples for this purpose. There are several works developed in this direction, including [19,42,45,47]. To the best of our knowledge, most of prior works on physical adversarial attacks mainly focus on the attack performance, and increasing adversarial strength of the perturbation is one of the most effective and direct ways for them. However, this usually leads to conspicuous and attention-grabbing patterns for the generated patches, which can be easily identified by human observers. To address this issue, we propose a method to craft physical adversarial patches for object detectors by leveraging the learned image manifold of generative adversarial networks (GANs) (*e.g.*, BigGAN [5] and StyleGAN [23, 24]) pretrained on real-world images. Through sampling images from GANs that minimizes detection score of a target object (*e.g.*, person), our

[1]Code is available at: https://github.com/aiiu-lab/Naturalistic-Adversarial-Patch

method can generate natural looking adversarial patches while maintaining acceptable attack performance. In addition, we also apply a clipping strategy to constrain the range of traversal from the initial starting point for optimization on the latent space of GAN for better image quality of generated patches. With extensive experiments on both digital and physical settings along with several independent subjective surveys, the results show that our proposed method produces significantly more realistic and natural looking patches than several state-of-the-art baselines while achieving competitive attack performance. A qualitative example is shown in Figure 1, where the patch generated by the proposed approach is more naturalistic and much less conspicuous than other compared methods.

From a security perspective, the existence of natural looking adversarial patches, which can not only fool detectors but also prevent suspicion from humans, is a potential issue. Thus, our work focuses on generating those natural-looking adversarial patches, verifying its existence and analyzing its properties.The main contributions of this work are summarized as follows:

- We leverage pretrained deep generative models (*i.e.,* StyleGAN, BigGAN) by traversing upon their latent spaces to craft more natural looking adversarial patches than other state-of-the-art baselines while maintaining the comparable attack capability.
- We conduct a thorough performance and naturalness analysis of the proposed method under different digital and physical settings in both indoors and outdoors.

2. Related Work

In this Section, we briefly review the recent relevant works on adversarial examples and deep image generation as follows:

2.1. Adversarial Examples

Adversarial examples are carefully crafted inputs to a model that will cause it to make mistakes. Szegedy et al. [41] first demonstrated that these adversarial examples can be easily made by adding small visually imperceptible noise towards the direction of an incorrect class. Their findings challenged the robustness and generalization of deep neural networks, sparking a whole new research field that follows a two-player game wherein attackers [6, 13, 19, 42, 45, 47] develop new ways to maliciously manipulate outputs of a model while defenders [18, 28, 37, 43, 49] try to develop ways to protect against them.

Adversarial examples can be broadly grouped into two types: digital adversarial examples and physical adversarial examples. Our work focuses on creating physical adversarial examples for object detectors.

Digital Adversarial Examples are crafted with the assumption that they have access to the digital image and can directly manipulate any of the pixels right before it is fed into the model. While it is an unrealistic assumption for practical scenarios, it elucidates a crucial flaw common to all deep neural networks and provides a test bed to gain insights on why such attacks work [16] and how to defend against it. Earlier works such as Fast Gradient Sign Method (FGSM) [16] and Projected Gradient Descent (PGD) [32] focused on methods for generating adversarial examples that can be efficiently incorporated inside the training loop as a form of augmentation, making networks more robust as a result. However, these are white-box methods that require access to the target model parameters, which limits the applicability and generalizability of the generated adversarial examples to other deep neural network models that we do not have access to [44]. To circumvent this requirement, several black-box techniques [11, 12, 44, 46] relied instead on querying the model with controlled inputs and observing their predicted classes or predicted probabilities. This improved the generalizability to different models, albeit lesser attack success rate. Recently, there are also works proposing a no-box [7, 29] method wherein they assume a setting where they cannot query the target model. Instead, they rely on generating adversarial examples from substitute models trained on a similar domain as the target model.

Physical Adversarial Examples are crafted with the purpose of them being printed out and recaptured by a camera [27]. As a result, the attacker can only control a subset of the pixels that will be fed into the victim model. Moreover, the attacker has no control over perspective, scale, and other processing that cameras perform, making it more challenging since the adversarial examples need to be robust to these various transformations for the attack to succeed in a physical setting [4]. Physical adversarial examples are usually generated in relation to a physical object which can be dynamically moving (such as wearable t-shirts [19, 42, 45, 47], eye-glass frames [38–40], or car license plates [48]), or static with respect to the scene (such as stickers [6, 14, 30], posters [26], and traffic signs [8, 36, 42]).

While prior methods achieved reasonable attack success rates, they typically have no control over the appearance and produce bright saturated colors with uncanny patterns, making them look unnatural (flamboyant, and attention-grabbing). These properties are undesirable for attackers since they are likely to get caught before being able to carry out the attack. Therefore, we desire methods for generating discreet and natural looking adversarial patches. Some recent works [13, 31] that tries to address this by restricting the deformations to blend with its surroundings. In contrast to these works, we leverage the natural image manifold learned by generative adversarial networks to generate natural looking adversarial patches.

2.2. Deep Image Generation

Generative models for image generation advanced rapidly in terms of image quality and fidelity due to GANs [15]. It introduced the notion of a learnable loss function where the target of the image generator network is defined by a separate discriminator network that tries to distinguish real from fake images. Through this two-player game, the generator gradually learns to synthesize fake images that are indistinguishable from the real images. While GANs produce visually appealing images, it often suffers from instability, vanishing gradients, and mode collapse. Succeeding works addressed these issues by either modifying the loss function [3,21,33], imposing gradient penalties [17,25,34], or adding normalizations [35]. However, even with these improvements, it is still challenging for GANs to generate high resolution images. This is because at higher resolutions, it is much easier to differentiate fake from real, which reduces the overlap between the fake and real distributions [22], making it hard for the discriminator to provide meaningful gradients to the generator. Progressive GANs [22] solved this by first training on low resolution images and then gradually increase its resolution. BigGANs [5] further improved upon the image quality of generating high resolution images by introducing several tricks to scale the training of GANs to very large batch sizes. Different from these approaches that focused on training strategies, StyleGAN [23] focused on improving the generator by separating the representations of content and style, allowing for not only very high quality images but also control of the synthesized image. This is further improved [24] in their second version (StyleGAN2) by introducing weight modulation and demodulation which significantly reduced the artifacts produced by GANs.

3. The Proposed Method

Our goal is to generate physical adversarial patches that are natural looking while still maintaining their attack performance. To achieve this, we propose to use a pretrained GAN generator to restrict the space of generated adversarial patches. Figure 2 shows an overview of our framework. Given a pretrained generator (Section 3.1), we search for an input latent vector corresponding to a generated image that causes the victim object detector to fail. This involves an optimization procedure where we compute the adversarial gradient direction (Section 3.2) for a target object detector and iteratively perform gradient updates to the input latent vector until a suitable adversarial patch is found. Additionally, we impose a threshold on the norm of the input latent vector that allows us to control the trade-off between realism and attack performance (Section 3.3).

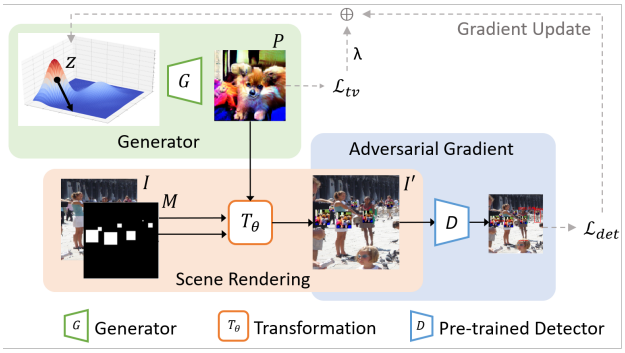


Figure 2. Overview of our naturalistic adversarial patch generation framework which crafts the patches for object detectors by leveraging the learned image manifold of a pretrained GAN upon real-world images and sampling the optimal image from the GAN through the iterative optimization process for the final patch.

3.1. Generating Adversarial Patches

Prior works optimize for an adversarial patch in the pixel space. In contrast, we optimize for an adversarial patch in the latent space of a GAN generator. Since GANs learn a latent space that approximates the manifold of natural images, our resulting adversarial patch would then be closer to the manifold of natural images, and thus, look more naturalistic.

We use a generator G pretrained on a set of natural images using a GAN framework and traverse its learned natural image manifold. We start with a latent vector $z \in \mathbb{R}^d$ randomly drawn from a standard normal distribution to generate an initial adversarial patch $P = G(z) \in \mathbb{R}^{H \times W \times 3}$. Next, we iteratively perform gradient updates on the latent vector z to look for a suitable z that optimizes our objective defined as follows:

$$L_{total} = L_{det} + \lambda_{tv} L_{tv}. \quad (1)$$

The first term L_{det} is the adversarial detection loss coming from the target object detector (discussed in Section 3.2). The second term L_{tv} is a total variation loss on the generated image to encourage smoothness. It is defined as:

$$L_{tv} = \sum_{i,j} \sqrt{(P_{i+1,j} - P_{i,j})^2 + (P_{i,j+1} - P_{i,j})^2}, \quad (2)$$

where the subindices i and j refer to the pixel coordinate of the patch P. We used $\lambda_{tv} = 0.1$ in all the experiments of this paper.

3.2. Adversarial Gradient

The generator relies on adversarial gradients to guide it in synthesizing images that can fool the target object detector. To get these adversarial gradients, we first render our adversarial patch onto a scene. Then, we feed it to an object

detector and compute an adversarial loss for the box detections.

Scene Rendering. For physical attacks, we have no control over the perspective, position, and scale of the adversarial patch with respect to the captured image. Thus, to make our adversarial patch robust to a wide range of possibilities, we render it on top of the clothes of humans and simulate different scenes at different settings. We also perform several transformations on our generated adversarial patch P such as rotation and occlusions to simulate different appearances that our adversarial patch may take in a practical scenario. Next, we use the object detector to get the locations of persons in a given image I. We create a mask around the clothing area of each person and place our adversarial patch P on these masks. We denote the new rendered image containing our adversarial patch as I'.

Adversarial Detection Loss. Object detectors, such as YOLO, output an arbitrary number of boxes or detections. For each detection j, we are interested in attacking two quantities: its objectness probability D^j_{obj} and its class probability D^j_{cls}. Minimizing the objectness probability D^j_{obj} causes the j-th object not to get detected. Minimizing class probability D^j_{cls} causes the j-th object to get classified into a wrong class (*e.g.* person gets classified as a dog.). In this paper, we focused on targeting the person class. Thus, we minimized both the objectness D^j_{obj} and class probabilities D^j_{cls} pertaining the the person class. For faster iterations, we do not compute the loss over all detected boxes. Instead, we only use the detected box having the highest objectness and class probabilities [42]. Our adversarial detection loss is defined below:

$$L_{det} = \frac{1}{N}\sum_{i=1}^{N} \max_j \left[D^j_{obj}(I'_i) D^j_{cls}(I'_i)\right], \tag{3}$$

where I'_i is the $i\text{-}th$ image in a batch with size N. Iteratively optimizing for Eq. 3 pushes the highest scoring detection to be low, thus, encouraging all the target objects to be either invisible or misclassified by the detector.

3.3. Realism vs Attack Performance Trade-off

Without any constraints, the model can optimize for latent vector z that is not contained within the high density region learned by the generator, therefore we can no longer expect the generated images to look realistic. Since the generator is trained by sampling random vectors from a standard normal distribution, we expect the high density region to be centered around the origin. Therefore, there is a higher probability of generating realistic images if z is closer to the origin.

To preserve realism, we ensure that the latent vector z will not have a norm greater than a threshold τ. Adjusting the norm threshold τ allows us to trade-off realism for attack performance. More details can be referred to the experimental section.

Similar to PGD, we adopt ℓ_∞ norm to constrain z. We update z using equations below:

$$z^t = \kappa(z^{t-1} + \eta \nabla L_{total}), \tag{4}$$

$$\kappa(z) = \{z_i | z_i \leftarrow \min(\max(z_i, -\tau), \tau), z_i \sim z\}, \tag{5}$$

where t is the time step, η is the step size, ∇L_{total} is the gradient of the objective, and κ is the clipping function defined in Eq. 5, where z_i is the i-th element of z.

4. Experimental Results

In this Section, we first describe the implementation details of the proposed approach followed by various qualitative and quantitative experiments for the proposed adversarial patch in the digital (*i.e.*, INRIA person dataset [9] and MPII Human Pose dataset [2]) and physical (*i.e.*, our recorded videos in different scenes) environments. In addition, we also provide various ablation studies for different parameters to get a more natural looking physical adversarial patch with comparable attack performance along with subjective evaluation of the naturalness of the generated patches.

4.1. Implementation Details

The whole optimization is performed using Adam optimizer with a learning rate of 0.01, $\beta_1 = 0.5$ and $\beta_2 = 0.999$. We reduce the learning rate if the change in losses is consistently below $1e^{-4}$ for at least 50 epochs. The batch-size is 8, and the total epoch is 1,000. Unless otherwise specified, we set $\tau = 50$. We used BigGAN[2] with an output resolution of 128×128 pretrained on the ImageNet-1k dataset, and StyleGAN2 with an output resolution of 512×512 pretrained on an anime character dataset as our generators[3]. In addition, since the BigGAN generator we used is a class conditional generator, it can guarantee that the generated patch is a certain class. We simply set the class vector to the dog and will test more classes in the next section. For the dimension d of the latent vector z, we used 120 for BigGAN, and 512 for StyleGAN2. For object detectors, we used YOLOv2[4], YOLOv3[5], YOLOv3tiny[5], YOLOv4[6], YOLOv4tiny[6], and FasterRCNN with an input image with the resolution of 416×416. Without explicitly stating, we use YOLOv4tiny to generate adversarial patches for the experiments. For the evaluation of the physical attack, we print the generated patches in the size of 40 cm $\times$ 30 cm onto a T-shirt.

[2]Pretrained model: https://github.com/anvoynov/GANLatentDiscovery

[3]Pretrained model: https://github.com/justinpinkney/awesome-pretrained-stylegan2

[4]https://gitlab.com/EAVISE/adversarial-yolo

[5]https://github.com/eriklindernoren/PyTorch-YOLOv3

[6]https://github.com/Tianxiaomo/pytorch-YOLOv4

Trained on \ Victim	YOLOv2	YOLOv3	YOLO3tiny	YOLOv4	YOLOv4tiny	FasterRCNN
$^{(P1)}$Ours-YOLOv2	**12.06**	43.50	32.12	50.56	24.89	52.54
$^{(P2)}$Ours-YOLOv3	56.67	**34.93**	41.46	56.29	37.46	61.78
$^{(P3)}$Ours-YOLOv3tiny	31.61	28.81	**10.02**	65.13	18.61	55.08
$^{(P4)}$Ours-YOLOv4	44.27	56.59	56.61	**22.63**	50.04	59.42
$^{(P5)}$Ours-YOLOv4tiny	34.68	37.79	21.69	46.80	**8.67**	59.97
$^{(P6)}$Ours-FasterRCNN	28.26	39.05	37.06	51.46	29.06	**42.47**
$^{(P7)}$Ours-ensemble†	49.42	35.46	25.29	51.71	18.51	61.28
Gray	72.66	74.17	67.52	66.52	64.74	61.54
$^{(P8)}$Random	75.03	73.75	78.91	76.71	75.74	73.00
White	69.63	74.93	66.45	72.48	59.66	65.40
$^{(P9)}$Adversarial Patches* [42]	2.13	22.51	8.74	12.89	3.25	39.41
$^{(P10)}$UPC** [19]	48.62	54.40	63.82	64.21	57.93	61.87

†trained on YOLOv2+YOLOv3+YOLOv4tiny * trained on YOLO ** trained on FasterRCNN

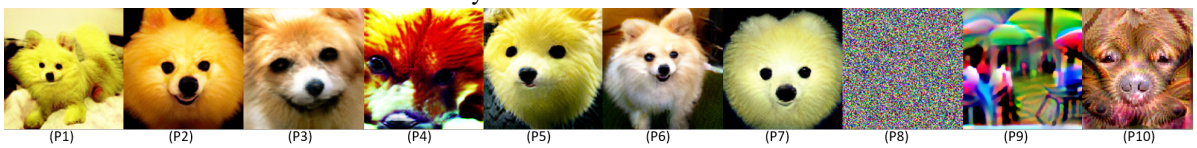
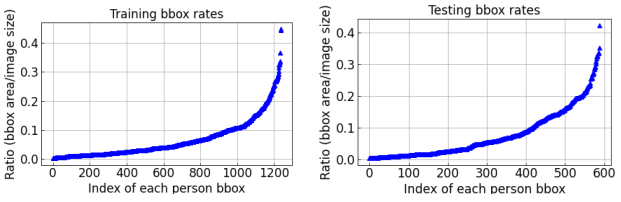


Table 1. Different patches evaluations in mAP(%) for the INRIA dataset using BigGAN. $P1$ to $P10$ indicate the corresponding patches, which are shown at the bottom of this table. The proposed patches not only attack detectors effectively but also are natural looking.



(a) training-data bbox-rate (b) testing-data bbox-rate

Figure 3. The ratios of the area per person bounding box with respect to the image size for the INRIA dataset, where y-axis represents the ratio and x-axis indicate the index of each person bounding box sorted in ascending order of the ratios. We can find there is a large size variations of the INRIA person dataset.

4.2. Datasets

For the training and evaluation of the proposed approach, we mainly use the INRIA person dataset. It consists of 614 training images and 288 test images. All the images are first resized to the resolution of 416×416. Since the outputs (*i.e.*, the size and tightness of the detected bounding boxes.) of various object detectors are slightly different from each other, we re-normalize the ground-truth bounding boxes by replacing them with the corresponding outputs of each detector with the largest Intersection-over-Union (IoU) score on the clean images for fair comparisons. As shown in Figure 3, the ratios for the area of each person's bounding box with respect to the whole image range from 0.002 to 0.448, and this makes the dataset suitable for training and test under the situation of different scales.

We also performed cross-dataset evaluations using the MPII Human Pose dataset to test the generalization of the proposed approach. We selected pictures in categories "running", "walking", and "dancing" classes. This results to a total of 1,646 images, which we split into 1,317 training and 329 testing images. Similarly, all the images are scaled to 416×416.

4.3. Evaluations

4.3.1 In-dataset Evaluation: INRIA

We evaluate our naturalistic adversarial patch in comparison with three standard patches and two state-of-the-art baselines (Adversarial Patch [42] and UPC [19]). We use mean average precision (mAP) as our main evaluation metric. Following the experimental settings of [42], we use the box detections of each detector on the clean dataset as the ground truth boxes (*i.e.,* the mAP of detectors will be 100% if there is no adversarial patch.) and report the average precision (AP) when evaluated with the adversarial patches. We evaluate on the setting where the adversarial patches are facing front, *i.e.,* no additional transformations on both training and testing. Table 1[7] shows the evaluation results on INRIA dataset. We utilize six different detectors to train the patches. In addition, we also ensemble some of them to jointly train the patch. $P1$ to $P10$ indicate the detectors and the corresponding patches.

[7]Different from initial version, we added two new detectors (FasterRCNN and ensemble YOLO). We also further optimized each patch for better performance.

Naturally, we achieve a very good attack performance when the victim detector during training matches the victim detector during testing. But surprisingly, our method can also perform well to other detectors despite not having any direct supervision from them. Note that although Adversarial Patch [42] achieves the best attack performance, its generated patches does not look naturalistic and is very attention-grabbing. On the other hand, our proposed method can generate very natural looking adversarial patches while still achieving reasonable attack performance.

4.3.2 Cross-dataset Evaluation: MPII Dataset

Furthermore, we also use the MPII dataset to evaluate the attack performance. The YOLOv4tiny results are shown in Table 2. Due to different nature of each dataset, we use different initial points to optimize each patch. It can be seen that the proposed method can also work in different datasets. However, we observe that if we use both INRIA and MPII to train the patch, it does not achieve better performance. We conjecture that it is due to the distribution difference between these two datasets.

Train on \ Test on	INRIA	MPII	Mix
INRIA	8.67	0.51	2.69
MPII	22.05	7.92	14.12
Mix	18.45	6.32	11.68

Table 2. Attack performance (mAP%) in different datasets using YOLOv4tiny.

4.3.3 Subjective Evaluation for the Naturalness of Different Adversarial Patches

The focus of the proposed approach is the naturalness and conspicuousness of the generated adversarial patch to humans. Therefore, we conduct a formal set of subjective evaluations to estimate the naturalness of our proposed patches as compared with baselines and real images. We conducted two subjective surveys, each with their own independent set of 24 participants. In the first subjective survey, we showed the patches in random order to the participants. For comparison, we generated 3 adversarial patches and gathered 12 off-the-shelf adversarial patches generated by [42] and [45]. We asked the participants to place a vote on each patch that looks natural for them. We compute the naturalness score as the percentage of votes for each patch respectively. As shown in Table 3 (Part 1), our proposed patches have higher naturalness scores than other baselines.

In the second subjective survey, we showed 6 of our generated patches together with 6 real images and ask them to vote for naturalness. As shown in Table 3 (Part 2), our generated adversarial patch achieves promising results relative to real images. There is still a quality gap between ours and real ones, which we conjecture the reason is due to the limited generation power of current GANs. We expect this can be bridged in the future by leveraging more advanced GANs or other deep generative models for more photo-realistic and higher-fidelity generated patches.

4.3.4 Physical Attack Evaluations

We performed physical attack evaluations on both indoor and outdoor settings by taking a video of one person wearing an adversarial shirt side-by-side with another person wearing an ordinary shirt as a baseline for comparison. We used the adversarial patch shown in Table 1 ($P5$). We selected YOLOv4tiny trained patch and utilized YOLOv4tiny as the evaluation detector. We quantified the attack performance based on recall. The two participants stand beside each other and are approximately two meters away from the camera. We asked the participants to move one step back and forth as well as side-to-side within the duration of the video. As shown in Table 4, our adversarial shirt can reduce the detection recall to approximately 23.80% in indoor settings (lab, living room, hallway), and down to approximately 43.14% in the much more challenging outdoor settings (balcony, grass field).

4.4. Discussions

4.4.1 Trade-off between Naturalness and Attack Performance

There is inevitably a trade-off between naturalness and attack performance. Adversarial attacks rely on finding perturbations or distortions that object detection models unintentionally have strong responses to [20, 50]. On the other hand, object detection models desire to ignore these distortions since they should not affect the class of the object. By training on large scale datasets, these models can already learn to ignore most naturally occurring distortions. This means that for adversarial patches, increasing naturalness will lead to a decrease in attack performance since the generated patches become closer to the space of distortions that the detector already learned to ignore, thus necessitating a sacrifice in either naturalness or attack performance. We would like to note that our model gives users the freedom to control this trade-off based on their preferences and requirements by adjusting the norm threshold τ.

To further illustrate this trade-off, we generated five different adversarial patches having different infinity norm values for their latent codes. In addition, we conducted a subjective survey to rank the naturalness of each patch. We ask ten people to rank these five patches, and sort them based on average the ranking. Figure 4 shows the average rank of each patch, its corresponding mAP, and the infinity norm of the latent code . It can be seen that the mAP decreases (*i.e.,* better attack performance), as the patch becomes less

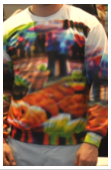
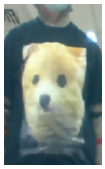
	Part 1				Part 2	
Images						
Naturalness Scores (%)	8.3	66.7	50.0	75.0	100.0	62.5
Source	[42]	ours	[45]	ours	[1]	ours

Table 3. Subjective tests for the naturalness evaluations of our adversarial patches with other baselines. The Naturalness scores are the ratios of votes for each test image over the whole group of participants. As shown in the results, ours get more votes than others, demonstrating its effectiveness. The complete set of patches used in survey are shown in the supplementary materials.

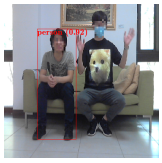
Image					
Setting	Lab	Living room	Hallway	Balcony	Grass
	Detection Recall				
w/o Adversarial Shirt	100%	100%	100%	100%	100%
w/ Adversarial Shirt	23.80%	24.49%	38.46%	44.33%	43.14%

Table 4. Percentage of detections from YOLOv4tiny with and without adversarial shirts at different physical settings.

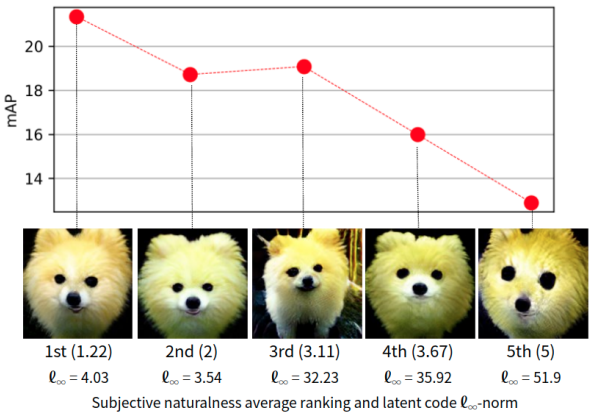


Figure 4. Naturalness test average score against attack performance using YOLOv4tiny.

natural looking. In addition, it can be seen that the infinity norm of the latent code is correlated with the naturalness and attack performance.

4.4.2 The Influence of Different Transformations

We explored various transformation methods for our scene rendering module to find suitable transformations that can enhance the attack performance of our adversarial patches. Table 5 shows the attack performance for each transformation under different settings, with or without using each transformation during the patch generation versus with or without using the corresponding transformation during evaluation. Surprisingly, using many of the transformations resulted in unsatisfactory performance. We hypothesize that the GAN latent space might be too limited to find a patch that is robust to all the transformations. Among the transformations considered, we observe that training with in-plane rotation can improve the attack performance from 13.42% to 10.16% when evaluated with in-plane rotation. We also observed that using blur and occlusions often leads to instability during training where the generated patches look unrealistic, therefore, we did not use them in other experiments. More results and illustration of the transformation can be found in the supplementary materials.

4.4.3 The Influence of Patch Size

To evaluate the performance influence of the patch size with respect to the size of the pedestrian, we conduct several digital qualitative and quantitative experiments on the INRIA dataset. As shown in the subfigure of Table 6, the larger the size of the patch is, the stronger its attack performance, which follows our expectations. Besides the qualitative samples, the quantitative results shown in Table 6 also demonstrate consistent findings. In addition, for a more precise evaluation, we further divide the test set of the INRIA dataset into two sets based on the ratios of the size of pedes-

Trans.($\mathcal{T}$)	Trained / Test	w/ $\mathcal{T}$ w/o any $\mathcal{T}$	w/ $\mathcal{T}$ w/ $\mathcal{T}$	w/o any $\mathcal{T}$ w/ $\mathcal{T}$
No trans.		8.67	8.67	8.67
In-plane rotation		15.11	10.16	13.42
Random translation		10.77	31.63	30.15
Crease		8.65	15.03	12.77
Out-of-plane rotation		10.4	27.77	26.25
Random occlusion		15.85	37.25	34.92
Blur		13.18	15.07	12.92
All		14.99	60.44	63.37

Table 5. The patches are trained with different transformations ($\mathcal{T}$) and then tested without or with the corresponding $\mathcal{T}$. It illustrates the performance influences (mAP%) under different transformations on the INRIA dataset and YOLOv4tiny.

Patch Scale	INRIA 0.0-1.0	INRIA 0.0-0.2	INRIA 0.2-1.0
0.3	1.64	1.41	13.50
0.25	5.92	5.39	21.98
0.2	17.27	16.91	27.59
0.15	36.82	36.50	44.83
0.1	74.60	75.48	55.17
0.05	95.04	94.82	100.0

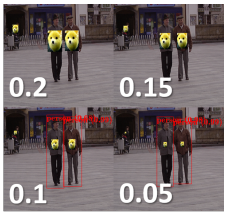


Table 6. The AP(%) of YOLOv4tiny with adversarial patches in different size settings for the INRIA dataset. "INRIA 0.0-0.2" represents the test data of the bbox-rate 0.2 to 1.0 of INRIA.

Victim \ Patches trained on	YOLOv2 Bulbul	YOLOv3tiny Penguin	YOLO v4 tiny Peacock
			
YOLOv2	23.4	38.42	44.47
YOLOv3tiny	22.18	12.57	30.79
YOLOv4tiny	24.36	19.1	25.16

Table 7. Evaluations (mAP%) with different classes generated by BigGAN (comparable with Table 1).

trians with respect to the whole image. The first set consists of samples with the ratios from 0 to 0.2 and the second is from 0.2 to 1. We can observe from Table 6 that targets with larger size are harder to attack.

4.4.4 Adversarial Patches in Different Classes

With the proposed approach, we can easily generate various physical adversarial patches with different classes. For example, with the conditional BigGAN, we can freely generate the adversarial patches with our specified class. We generate adversarial patches for bulbul, penguin, and peacock. The results in Table 7 demonstrate satisfactory attack performance on the INRIA dataset.

Patch			
mAP	36.1	34.34	40.24

Table 8. Evaluations with patches generated by StyleGAN2 and YOLOv4tiny (comparable with Table 1).

4.4.5 Adversarial Patches using Different GAN

In the previous experiments, we mainly utilize BigGAN to generate the adversarial patch and analyze the proposed method. To further evaluate the proposed method and confirm it can generate adversarial patches with different GAN, we replace BigGAN with StyleGAN2 as the generator. Since we consider printing the patch on the clothes, we use the anime face pretrained weight to generate anime face patches. Table 8 shows the results. We use YOLOv4tiny to train three patches, and use YOLOv4tiny to evaluate. It can be seen that the adversarial patches generated by different GANs are also effective in attacking detectors. In addition, the trade-off between naturalness and attack performance can also be seen in this table.

5. Conclusion

In this work, we propose a method to craft naturalistic physical adversarial patches for object detectors by leveraging the learned image manifold of pretrained GAN models. With the astonishing image generation capability of state-of-the-art GAN models, our method can successfully generate more natural looking adversarial patches while maintaining competitive attack performance than other similar methods from extensive qualitative and quantitative experiments in both digital and physical domains along with the subjective naturalness evaluation. Although there is still a quality gap between ours and real images from the results of the subjective evaluation, we expect this can be bridged in the future by leveraging more advanced GANs or other deep generative models for more photo-realistic and higher-fidelity generated patches. Meanwhile, a better non-reference perceptual quality assessment method could further help enhance the quality of the generated patches beyond our proposed clipping strategies and will be explored as the future work.

Acknowledgement This work was supported in part by the Ministry of Science and Technology of Taiwan (R.O.C) under Grants MOST-110-2221-E-001-009-MY2, MOST-110-2221-E-001-002, MOST-109-2221-E-001-020, MOST-108-2218-E-001-004-MY2, MOST-109-2223-E-009-002-MY3, MOST-110-2218-E-A49-018, MOST-110-2634-F-007-015, and MOST-109-2221-E-011-125-MY3.

References

[1] TotallyHer Media LLC an Evolve Media LLC company. dogtime. https://dogtime.com/dog-breeds/akita-chow, 2021. 7

[2] Mykhaylo Andriluka, Leonid Pishchulin, Peter Gehler, and Bernt Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, June 2014. 4

[3] Martin Arjovsky, Soumith Chintala, and Léon Bottou. Wasserstein generative adversarial networks. In *International conference on machine learning*, pages 214–223. PMLR, 2017. 3

[4] Anish Athalye, Logan Engstrom, Andrew Ilyas, and Kevin Kwok. Synthesizing robust adversarial examples. In *International conference on machine learning*, pages 284–293. PMLR, 2018. 2

[5] Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *International Conference on Learning Representations*, 2019. 1, 3

[6] Tom B Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *arXiv preprint arXiv:1712.09665*, 2017. 2

[7] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 15–26, 2017. 2

[8] Shang-Tse Chen, Cory Cornelius, Jason Martin, and Duen Horng Polo Chau. Shapeshifter: Robust physical adversarial attack on faster r-cnn object detector. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 52–68. Springer, 2018. 2

[9] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7263–7271, 2005. 4

[10] Jiankang Deng, Jia Guo, Yuxiang Zhou, Jinke Yu, Irene Kotsia, and Stefanos Zafeiriou. Retinaface: Single-stage dense face localisation in the wild. *arXiv preprint arXiv:1905.00641*, 2019. 1

[11] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018. 2

[12] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4312–4321, 2019. 2

[13] Ranjie Duan, Xingjun Ma, Yisen Wang, James Bailey, A Kai Qin, and Yun Yang. Adversarial camouflage: Hiding physical-world attacks with natural styles. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1000–1008, 2020. 2

[14] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust physical-world attacks on deep learning visual classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018. 2

[15] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Advances in Neural Information Processing Systems*, volume 27, 2014. 3

[16] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014. 2

[17] Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. Improved training of wasserstein gans. In *Advances in Neural Information Processing Systems*, volume 30, 2017. 3

[18] Jamie Hayes. On visible adversarial perturbations & digital watermarking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2018. 2

[19] Lifeng Huang, Chengying Gao, Yuyin Zhou, Cihang Xie, Alan L Yuille, Changqing Zou, and Ning Liu. Universal physical camouflage attacks on object detectors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 720–729, 2020. 1, 2, 5

[20] Andrew Ilyas, Shibani Santurkar, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. *Advances in neural information processing systems*, 32, 2019. 6

[21] Alexia Jolicoeur-Martineau. The relativistic discriminator: a key element missing from standard GAN. In *International Conference on Learning Representations*, 2019. 3

[22] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of GANs for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. 3

[23] Tero Karras, Samuli Laine, and Timo Aila. A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4401–4410, 2019. 1, 3

[24] Tero Karras, Samuli Laine, Miika Aittala, Janne Hellsten, Jaakko Lehtinen, and Timo Aila. Analyzing and improving the image quality of stylegan. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8110–8119, 2020. 1, 3

[25] Naveen Kodali, James Hays, Jacob Abernethy, and Zsolt Kira. On convergence and stability of GANs, 2018. 3

[26] Zelun Kong, Junfeng Guo, Ang Li, and Cong Liu. Physgan: Generating physical-world-resilient adversarial examples for autonomous driving. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 14254–14263, 2020. 2

[27] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016. 2

[28] Cassidy Laidlaw, Sahil Singla, and Soheil Feizi. Perceptual adversarial robustness: Defense against unseen threat models. In *International Conference on Learning Representations*, 2021. 2

[29] Qizhang Li, Yiwen Guo, and Hao Chen. Practical no-box adversarial attacks against dnns. *arXiv preprint arXiv:2012.02525*, 2020. 2

[30] Aishan Liu, Xianglong Liu, Jiaxin Fan, Yuqing Ma, Anlan Zhang, Huiyuan Xie, and Dacheng Tao. Perceptual-sensitive gan for generating adversarial patches. In *Proceedings of the AAAI conference on artificial intelligence*, pages 1028–1035, 2019. 2

[31] Jinqi Luo, Tao Bai, Jun Zhao, and Bo Li. Generating adversarial yet inconspicuous patches with a single image. *arXiv preprint arXiv:2009.09774*, 2020. 2

[32] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 2

[33] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proceedings of the IEEE international conference on computer vision*, pages 2794–2802, 2017. 3

[34] Lars Mescheder, Andreas Geiger, and Sebastian Nowozin. Which training methods for gans do actually converge? In *International conference on machine learning*, pages 3481–3490. PMLR, 2018. 3

[35] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. In *International Conference on Learning Representations*, 2018. 3

[36] Nir Morgulis, Alexander Kreines, Shachar Mendelowitz, and Yuval Weisglass. Fooling a real car with adversarial traffic signs. *arXiv preprint arXiv:1907.00374*, 2019. 2

[37] Muzammal Naseer, Salman Khan, and Fatih Porikli. Local gradients smoothing: Defense against localized adversarial attacks. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 1300–1307. IEEE, 2019. 2

[38] Mikhail Pautov, Grigorii Melnikov, Edgar Kaziakhmedov, Klim Kireev, and Aleksandr Petiushko. On adversarial patches: real-world attack on arcface-100 face recognition system. In *2019 International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON)*, pages 0391–0396. IEEE, 2019. 2

[39] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition. In *Proceedings of the 2016 acm sigsac conference on computer and communications security*, pages 1528–1540, 2016. 2

[40] Mahmood Sharif, Sruti Bhagavatula, Lujo Bauer, and Michael K Reiter. A general framework for adversarial examples with objectives. *ACM Transactions on Privacy and Security (TOPS)*, 22(3):1–30, 2019. 2

[41] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations*, 2013. 2

[42] Simen Thys, Wiebe Van Ranst, and Toon Goedemé. Fooling automated surveillance cameras: adversarial patches to attack person detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 1, 2, 4, 5, 6, 7

[43] Tzungyu Tsai, Kaichen Yang, Tsung-Yi Ho, and Yier Jin. Robust adversarial objects against deep learning models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 954–962, 2020. 2

[44] Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey, and Xingjun Ma. Skip connections matter: On the transferability of adversarial examples generated with resnets. In *International Conference on Learning Representations*, 2020. 2

[45] Zuxuan Wu, Ser-Nam Lim, Larry S Davis, and Tom Goldstein. Making an invisibility cloak: Real world adversarial attacks on object detectors. In *European Conference on Computer Vision*, pages 1–17. Springer, 2020. 1, 2, 6, 7

[46] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2730–2739, 2019. 2

[47] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial t-shirt! evading person detectors in a physical world. In *European Conference on Computer Vision*, pages 665–681. Springer, 2020. 1, 2

[48] Kaichen Yang, Tzungyu Tsai, Honggang Yu, Tsung-Yi Ho, and Yier Jin. Beyond digital domain: Fooling deep learning based recognition system in physical world. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 1088–1095, 2020. 2

[49] Ping yeh Chiang, Renkun Ni, Ahmed Abdelkader, Chen Zhu, Christoph Studor, and Tom Goldstein. Certified defenses for adversarial patches. In *International Conference on Learning Representations*, 2020. 2

[50] Dong Yin, Raphael Gontijo Lopes, Jonathon Shlens, Ekin D Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. *arXiv preprint arXiv:1906.08988*, 2019. 6