PRACTICAL NO-BOX ADVERSARIAL ATTACKS WITH TRAINING-FREE HYBRID IMAGE TRANSFORMATION

Anonymous authors
Paper under double-blind review

ABSTRACT

In recent years, the adversarial vulnerability of deep neural networks (DNNs) has raised increasing attention. Among all the threat models, no-box attacks are the most practical but extremely challenging since they neither rely on any knowledge of the target model or similar substitute model, nor access the dataset for training a new substitute model. Although a recent method has attempted such an attack in a loose sense, its performance is not good enough and the computational overhead of training is expensive. In this paper, we move a step forward and show the existence of a **training-free** adversarial perturbation under the no-box threat model, which can be successfully used to attack different DNNs in real-time. Motivated by our observation that high-frequency component (HFC) domains in low-level features and plays a crucial role in classification, we attack an image mainly by manipulating its frequency components. Specifically, the perturbation is combined by the suppression of the original HFC and the adding of noisy HFC. We empirically and experimentally analyze the requirements of effective noisy HFC and show that it should be regionally homogeneous, repeating and dense. Extensive experiments on the ImageNet dataset demonstrate the effectiveness of our proposed no-box method. It attacks ten well-known models with a success rate of **98.13%** on average, which outperforms state-of-the-art no-box attacks by **29.39%**. Furthermore, our method is even competitive to mainstream transfer-based black-box attacks. Our code is available in our appendix.

1 INTRODUCTION

Deep neural networks (DNNs) are widely known to be vulnerable to adversarial examples (Szegedy et al., 2013; Goodfellow et al., 2015), *i.e.*, a human-imperceptible perturbation can lead to misclassification. In adversarial machine learning, the term *threat model* defines the rules of the attack, such as the resources the attacker can access. Based on the threat model, the attacks are often divided into white-box attacks and black-box attacks. In the white-box threat model (Szegedy et al., 2013; Goodfellow et al., 2015; Madry et al., 2018a), the attacker has full knowledge of a target model, such as the model weights and the whole training dataset. Recognizing the threat of these adversarial attacks, a model owner is unlikely to leak a model's information to the public. Thus, the white-box attack is often used to evaluate the model robustness for revealing its weakest point (Madry et al., 2018a), but often not considered as a practical attack method (Chen et al., 2017). To this end, numerous works have investigated a more realistic threat model, where the attacker does not require full knowledge of the target model, *i.e.*, the backpropagation on the target model is prohibited. This threat model is called black-box attack (Papernot et al., 2016; Tramèr et al., 2016; Papernot et al., 2017; Narodytska & Kasiviswanathan, 2017; Chen et al., 2017; Brendel et al., 2017; Dong et al., 2019b; Yan et al., 2019; Chen et al., 2020; Zhou et al., 2020). However, such a black-box threat model usually involves a major concern of being resource-intensive in terms of query cost and time. In real-world attack scenarios, even if we ignore such concerns, query-based black-box attack can still be infeasible, *e.g.*, the model API is inaccessible to the attacker. Moreover, it might cause suspicion due to repeated queries to the model with almost the same adversarial image. To alleviate this issue, another line of black-box threat model (Dong et al., 2018; Xie et al., 2019b; Dong et al., 2019a; Wu et al., 2020; Lin et al., 2020; Gao et al., 2020a; 2021) called transfer-based attack is proposed. In this threat model, adversarial examples are crafted via the local available pre-trained substitute model, which usually trains on the same training dataset as the target model. The resultant adversarial examples

are expected to attack the target model. However, without the feedback from the target model, the transferability heavily depends on how large the gap between the substitute model and target model. In practice, this gap is large because the structure and the training technique of the target model are usually not publicly available due to security and privacy concerns.

From the analysis above, we argue that both white-box and black-box attacks can hardly be considered as practical attacks. A practical attack should satisfy two criteria: (a) **model-free**, *i.e.*, no dependence on the pre-trained substitute model or the target model for either backward propagation or only forward query; (b) **data-free**, *i.e.*, no dependence on the dataset for training a substitute model. We term it no-box attack. A recent work (Li et al., 2020a) is the first (to our knowledge) as well as the only work to have attempted such an attack in a loose sense. Their threat model still requires a small number of auxiliary samples, such as 20 images. Admittedly, collecting a small number of samples might not be difficult in most cases, but might be still infeasible in some security-sensitive applications. Specifically, their approach (Li et al., 2020a) attempts to train a substitute model by adopting the classical auto-encoder model instead of the supervised classification model due to the constraint of a small-scale dataset. Overall, to attack a certain sample, their approach consists of three steps: (1) collecting a small number of images; (2) training a substitute model; (3) white-box attack on the substitute model. If a new sample, especially from a different class, needs to be attacked, the above process needs to be repeated. Thus, their approach is very resource-intensive. Besides, their attack success rate is still significantly lower than existing black-box attacks.

By contrast, our approach does not require any of the above three steps and is even training-free. With the help of visualization technique proposed by (Zeiler & Fergus, 2014), we observe that the high-frequency component (HFC), *e.g.*, the edge and texture features, is dominant in shallow layers and the low-frequency component (LFC), *e.g.*, the plain areas in the image, is paid less attention to be extracted. Combined with the insight into the classification logic of DNNs in Sec. 3.1, we observe that HFC plays a crucial role in recognition. As shown in Fig. 1, without LFC, the confidence of HFC is even higher than the raw image. Although it does not hold true for all samples, it does demonstrate the importance of HFC.

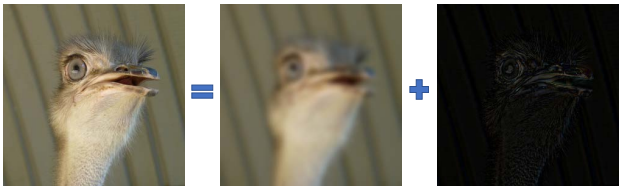


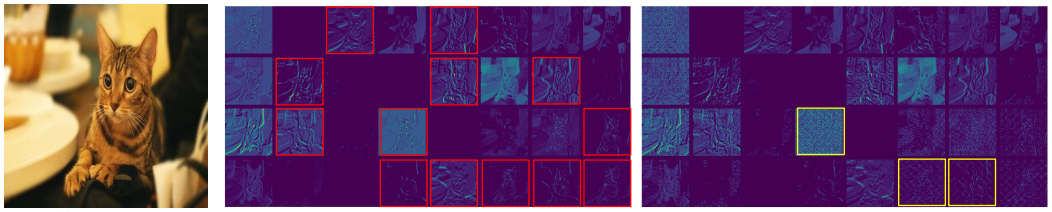
Figure 1: The confidence of a raw image (**left**), its low-frequency component (**middle**) and high-frequency component (**right**) on Inc-v3 (Szegedy et al., 2016).

Motivated by this, we take the idea of hybrid image (Oliva, 2013) and propose a novel **Hybrid Image Transformation (HIT)** attack method to craft adversarial examples. Formally, it only needs three steps but can effectively fool various DNNs without any training: First, due to the training-free setting and inspired by the analysis from Sec. 3.2, we simply utilize matplotlib[1] tool to draw several geometric patterns which serve as the proto-patterns, and the resultant synthesized adversarial patches are thus richer in **regionally homogeneous**, **repeating** and **dense** HFC. Second, we extract the LFC of the raw image and HFC of the adversarial patch. Finally, we combine these two pieces of components and clip them to the ε-ball of the raw image to get the resultant adversarial hybrid example. Extensive experiments on ImageNet demonstrate the effectiveness of our method. By attacking ten state-of-the-art models in the no-box manner, our HIT significantly increases the average success rate from 68.74% to **98.13%**. Notably, our HIT is even competitive to mainstream transfer-based black-box attacks.

2 Related Work

Adversarial Attack. Let $\boldsymbol{x}$ denote raw image without any perturbation, $\boldsymbol{x}^{\boldsymbol{adv}}$ and y denote the corresponding adversarial example and true label respectively. In generally, we use l_∞-norm to measure the perceptibility of adversarial perturbations, *i.e.*, $||\boldsymbol{x}^{\boldsymbol{adv}} - \boldsymbol{x}||_\infty \leq \varepsilon$. In this paper, we focus on non-targeted attacks (Dong et al., 2018; Xie et al., 2019b; Wu et al., 2020; Lin et al., 2020; Gao et al., 2020a) which aim to cause misclassification of DNNs $f(\cdot)$, *i.e.*, $f(\boldsymbol{x}^{\boldsymbol{adv}}) \neq y$.

[1] https://matplotlib.org/



(a) Raw image (b) Shallow layer representations of (a) (c) Shallow layer representations of adversarial example crafted by our HIT with circle patch

Figure 2: The visualization for the shallow layer ("activation 2") feature maps of Inception-v3 (Szegedy et al., 2016) w.r.t the input (a) and its corresponding adversarial example crafted by our HIT.

Competitors. Transferability is an important property for adversarial examples. With it, the resultant adversarial example crafted via one model may fool others. For the black-box threat model, Goodfellow et al. (2015) argue that the vulnerability of DNNs is their linear nature, and generate adversarial examples efficiently by performing FGSM which is a single-step attack. Papernot et al. (2017) train a local model with many queries to substitute for the target model. Dong et al. (2018) integrate a momentum term into I-FGSM Kurakin et al. (2017) to stabilize the update direction during the attack iterations. Xie et al. (2019b) apply diverse input patterns to improve the transferability of adversarial examples. Dong et al. (2019a) propose a translation-invariant attack to mitigate the effect of different discriminative regions between models. Gao et al. (2020a) introduce patch-wise perturbation by amplifying the step size and reuse the cut noise to perturb more information in discriminative regions. For the no-box threat model, Li et al. (2020a) attempt to attack the target model without any model query or the accessible pre-trained substitute model. In their work, with a limited amount of data, they try different mechanisms (with or without supervised technique) to train the substitute model, and then utilize this substitute model to craft transferable adversarial examples. Different from these approaches, our method does not depend on transferability since we do not need any substitute model. In this paper, we craft the adversarial examples from the perspective of the classification logic of DNNs.

Frequency Perspective on DNNs. Our approach is highly inspired by existing works which explain the generalization and adversarial vulnerability of DNNs from the frequency perspective. The fact that DNNs have good generalization while being vulnerable to small adversarial perturbations has motivated (Jo & Bengio, 2017; Wang et al., 2020) to investigate the underlying mechanism, suggesting that surface-statistical content with high-frequency property is essential for the classification task. From the perspective of texture vs. shape, Geirhos et al. (2019); Wang et al. (2020) reveal that DNNs are biased towards texture instead of shape. Since the texture content is considered to have high-frequency property, their finding can be interpreted as the DNN being biased towards HFC. On the other hand, adversarial perturbations are also known to have the high-frequency property and various defense methods have also been motivated from this insight (Aydemir et al., 2018; Das et al., 2018; Liu & JaJa, 2019; Xie et al., 2019a). Nonetheless, it remains unknown whether manually designed high-frequency patterns are sufficient for attacking the network.

3 METHODOLOGY

Although many adversarial attack methods (Papernot et al., 2016; Dong et al., 2018; Gao et al., 2020a; Li et al., 2020a) have achieved pretty high success rates in both black-box and no-box cases, they all need training, especially for query-based (Papernot et al., 2016; Zhou et al., 2020) and no-box adversarial perturbations (Li et al., 2020a) whose training is usually time-consuming. Then a natural question arises: *Is it possible to generate robust adversarial perturbations without any training?* In the following subsections, we will give our answer and introduce our design.

3.1 MOTIVATION

To better understand the role of HFC and LFC for the classification results of DNNs, we split the information of raw images into these two pieces via Gaussian low-pass filter (defined in Eq. 1).

As illustrated in Fig. 3, when the kernel size is small, *i.e.*, the cutoff frequency is high, the average accuracy of LFC on ten state-of-the-art models is close to 100%. However, if we continue to increase the kernel size, the average accuracy of HFC begins to exceed LFC one. To our surprise, for several specific raw images, *e.g.*, left image of Fig. 1, the true label's confidence of HFC which is mostly black is even higher than the raw image.

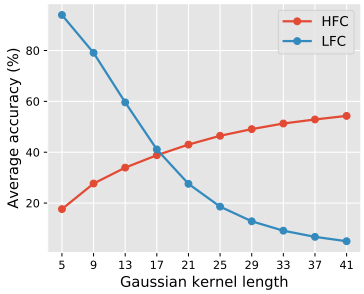


Figure 3: The average accuracy of HFC and LFC obtained by different Gaussian kernel. We show the visualization and confidence for kernel length 17 in Fig. 1.

To explain the above phenomenon, we turn to the perspective of feature space. Inspired by recent intermediate feature-based attacks (Zhou et al., 2018; Ganeshan & Babu, 2019; Inkawhich et al., 2019), we argue low-level features are critical to the classification. Interestingly, as shown in Fig. 2, most[2] feature maps in the shallow layers generally extract the edge and texture features (typical ones are highlighted by red boxes), *i.e.*, HFC, and pay less attention to plain areas in images, *i.e.*, LFC. *Therefore, if a perturbation can effectively manipulate the HFC of an image, totally different low-level features will be extracted and may lead to misclassification.*

3.2 Effective Adversarial HFC

However, what kind of training-free noisy HFC can effectively fool DNNs is still unknown because the performance of any other raw image's HFC is unsatisfactory (see Appendix Sec. A.8). Zhang et al. (2020) have demonstrated that the effectiveness of adversarial perturbation lies in the fact that it contains irrelevant features. The features of perturbation dominate over the features in the raw image, thus leading to misclassification. Inspired by their finding, we intend to design adversarial HFC with strong irrelevant features, and we conjecture that the following properties are essential.

Regionally Homogeneous. Several recent works (Li et al., 2020b; Gao et al., 2020a; Dong et al., 2019a; Gao et al., 2020b) have demonstrated that adversarial perturbations with regionally homogeneous (or patch-wise (Gao et al., 2020a)) property can enhance the transferability of adversarial examples. Inspired by that the raw image is a composite of homogeneous patterns, the reason might be attributed to that this perturbation tend to form irrelevant features recognizable by the DNNs.

Repeating. Nguyen et al. (2015) observe that extra copies of the repeating element do improve the confidence of DNNs. From the perspective of strengthening the irrelevant features, it is expected that repeating the content is beneficial.

Dense. Analogous to the above *repeating* property that performs global repeating, *i.e.*, increases the amount of irrelevant features globally, we can also perform local repeating to strengthen its adversarial effect further. For term distinction, we term this property *dense*.

To verify the effect of the above properties, we conduct the ablation study in Sec. 4.1, and results support our conjecture. Besides, the analysis in Appendix Sec. A.9 also show that our HIT has potential to become a targeted attack.

3.3 Hybrid Image Transformation

Motivated by the above discussion, we take the idea of hybrid image (Oliva, 2013) to apply our no-box attacks. Specifically, Oliva (2013) *replaces* the HFC of one image with the HFC of another carefully picked image and craft hybrid images with two different interpretations: one that appears when the image is viewed up-close, and the other that appears from afar (see Fig.5 of Oliva (2013)). However, confusing human's vision system (without ε constrain) cannot guarantee the misclassification of DNNs since adversarial examples are constrained by the maximum perturbation. Therefore, we propose a novel **Hybrid Image Transformation (HIT)** attack method which *reduces*[3] original

[2] see quantitative analysis in Appendix Sec. A.2.

[3] Due to the ε constraint, we can not completely replace HFC with others

HFC, and meanwhile, adds well-designed noisy ones to attack DNNs. Our method only needs three steps but can generate robust training-free adversarial perturbations in real time:

Figure 4: Three simple geometric patterns serve as proto-patterns.

First, we provide an adversarial patch $\boldsymbol{x^p}$ to generate noisy HFC. Unlike the traditional way that needs training, here we use the matplotlib tool to draw it. Inspired by the observation in Sec. 3.2, we consider three simple **regionally homogeneous** proto-patterns (to avoid cherry-picking) as our basic adversarial patches: concentric circles, concentric squares, and concentric rhombus in Fig. 4. The effect of concentric pattern is to make the resultant HFC **dense**. Then we **repeat** these adversarial patches.

Second, we extract the LFC of the raw image and the HFC of the adversarial patch. Note that several methods can be utilized to extract the HFC and LFC of an image, *e.g.*, Fourier transformation. In this paper, we use an approximated yet simple Gaussian low-pass filter $\boldsymbol{G}$ whose size is $(4k+1)\times(4k+1)$ to get LFC, which can be written as:

$$\boldsymbol{G_{i,j}} = \frac{1}{2\pi\sigma^2} e^{(-\frac{i^2+j^2}{2\sigma^2})}, \tag{1}$$

where $\sigma = k$ determines the *width* of our $\boldsymbol{G}$. In general, the larger σ is, the more HFC is filtered out. We are not going to introduce a new high-pass filter here for simplicity and just get HFC by $\boldsymbol{G}$. More specifically, we obtain HFC by subtracting the LFC of the adversarial patch.

Finally, we can synthesize these two part components to generate our adversarial hybrid image $\boldsymbol{x^{adv}}$:

$$\boldsymbol{x^{adv}} = clip_{x,\varepsilon}(\boldsymbol{x} * \boldsymbol{G} + \lambda \cdot (\boldsymbol{x^p} - \boldsymbol{x^p} * \boldsymbol{G})), \tag{2}$$

where "*" denotes convolution operation, λ is a weight factor to balance the LFC and HFC, and $clip_{x,\varepsilon}(\cdot)$ restricts the resultant adversarial examples within the ε-ball of the raw image in l_∞ space. Therefore, our method is different from adversarial patch attacks (Brown et al., 2017; Liu et al., 2020) which replace a subregion of the image with a well-design patch.

As illustrated in Fig. 2(c), our HIT can effectively reduce relevant HFC and add many other irrelevant noisy ones, *e.g.*, highlighted yellow boxes in (c) cannot find any obvious HFC associated with "cat" at all. As a result, the target model can not extract correct features to make a reasonable prediction, thus leading to misclassification. Besides, our adversarial examples are less perceptible than those of our competitors (See Appendix Sec. A.4).

4 Experiments

Networks. Here we consider ten well-known classification models: VGG19 (Simonyan & Zisserman, 2015), Inception-v3 (Inc-v3) (Szegedy et al., 2016), ResNet-152 (ResNet) (He et al., 2016), DenseNet-121 (Dense) (Huang et al., 2017), WideResNet (WRN) (Zagoruyko & Komodakis, 2016), SENet (Hu et al., 2018), PNASNet (PNA) (Liu et al., 2018), ShuffleNet-v2 (Shuffle) (Ma et al., 2018), SqueezeNet (Squeeze) (Iandola et al., 2017) and MobileNet-v2 (Mobile) (Sandler et al., 2018) as our target models. All the models are available in the Torchvision[4], except for PNA and SENet which are obtained from Github[5]. We also perform our attack on a real-world recognition system in Appendix Sec. A.6.

[4] https://github.com/pytorch/vision/tree/master/torchvision/models
[5] https://github.com/Cadene/pretrained-models.pytorch

Dataset. To make our method more convincing and avoid cherry-picking, we choose 10,000 images (each category contains about 10 images which are resized to $299 \times 299 \times 3$ beforehand) from the ImageNet validation set (Russakovsky et al., 2015) which are classified correctly by all ten networks we consider. We also discuss our methods on other classification tasks in Appendix Sec. A.5.

Parameters. In our experiments, we use l_∞-norm to measure the perceptibility of adversarial noises, unless specified, the maximum perturbation ε is set to 16 (results with a smaller ε can be found in Appendix Sec. A.7). For our HIT, the size of Gaussian kernel $\boldsymbol{G}$ is 17×17 (*i.e.* $k = 4$), weight factor λ is set to 1.0 (the discussion about λ is shown in Appendix Sec. A.3), and density of proto-pattern is set to 12. For tile-size, unless specified, we set to 50×50, *i.e.*, tile-scheme is 6×6. For no-box methods, we follow the same setting as (Li et al., 2020a). For black-box methods, the iteration T is set to 10 and the step size α is 1.6. For MI-FGSM (Dong et al., 2018), we adopt the default decay factor $\mu = 1.0$. For DI^2-FGSM (Xie et al., 2019b), we set the transformation probability to 0.7. For TI-FGSM (Dong et al., 2019a), the length of Gaussian kernel is 15. For PI-FGSM (Gao et al., 2020a), the length of project kernel is 3, the amplification factor β and project factor γ are 10.0 and 16.0, respectively. Different from PI-FGSM, β and γ for PI-MI-DI^2-FGSM and PI-TI-DI^2-FGSM (Gao et al., 2020a) is 2.5 and 2.0.

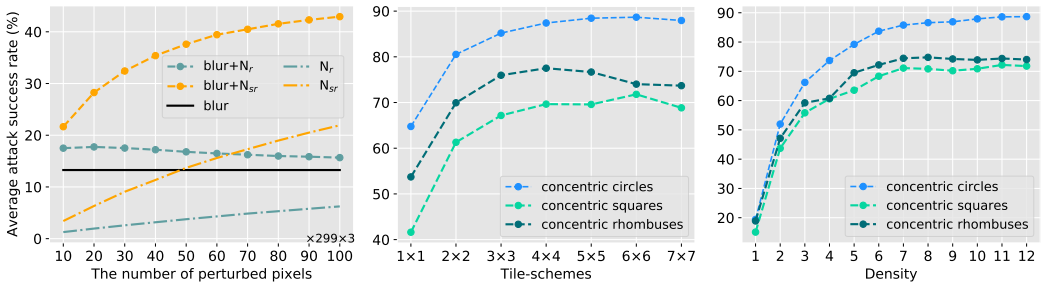


Figure 5: The average attack success rates (%) of ten models w.r.t the strength of semi-random noise N_{sr} and random noise N_r (**left**), tile-schemes (**middle**) and densities (**right**). "blur" denotes using Gaussian kernel to smooth the image (constrained by maximum perturbation ε).

4.1 Ablation Study

In this section, we conduct a series of ablation study for our HIT. Specifically, we investigate the effectiveness of regionally homogeneous pattern, repeating pattern and dense pattern in Sec. 4.1.1, Sec. 4.1.2 and Sec. 4.1.3, respectively. Besides, we also analyze the effect of perturbation size on the performance in Sec. 4.1.4. For the result of HIT without reducing HFC beforehand is shown in Appendix Tab. A.12.

4.1.1 The Effect of Regionally Homogeneous Pattern

To the best of our knowledge, regionally homogeneous perturbations (Dong et al., 2019a; Gao et al., 2020a;b; Li et al., 2020b) are mostly based on the gradient to craft, thereby training is necessary. However, whether arbitrary noise can benefit from the homogeneous property remains unclear. Therefore, we compare random noises with semi-random ones to check it:

Random noise: For a given random location pair set L, we call $\boldsymbol{N_r} \in \mathbb{R}^{H \times W \times C}$ random noise if it meets the following formula:

$$\boldsymbol{N_r}[i,j,c] = \begin{cases} \varepsilon \cdot random(-1,1), & (i,j,c) \in L \\ 0, & else \end{cases} \tag{3}$$

Semi-random noise: Different from the random noise, semi-random noise has some regularity. Let S denotes a semi-random location pair set, and here we take H-dimension random noise as an example. $\boldsymbol{N_{sr}}$ can be written as:

$$\boldsymbol{N_{sr}}[i,:,:] = \begin{cases} \varepsilon \cdot random(-1,1), & i \in S \\ 0, & else \end{cases} \tag{4}$$

where *random*(-1, 1) returns 1 or -1 randomly. As depicted in Fig. 5, the success rates of N_{sr} are consistently higher than those of N_r. As the number of perturbed pixels increases, the margin between them also increases. This demonstrates that training-free noise can also benefit from **regionally homogeneous** property. To exploit this conclusion further, in Fig. 4, we extend semi-random noise to other more complex "continuous" patterns, *e.g.*, circle.

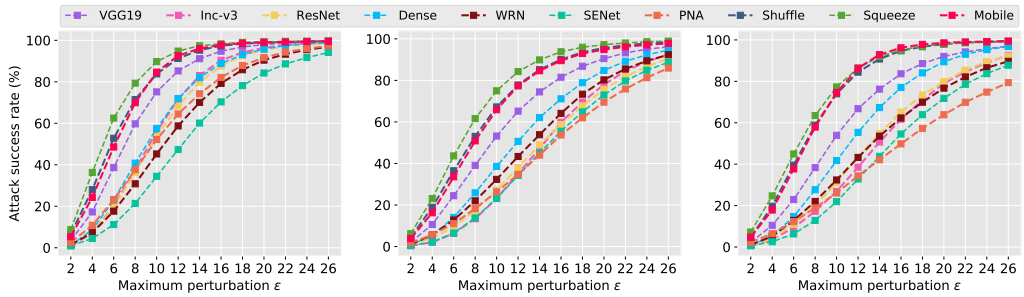


Figure 6: The attack success rates (%) of circle patches (**left**), square patches (**middle**) and rhombuses patches (**right**) w.r.t maximum perturbation ε.

4.1.2 The Effect of Repeating Pattern

In this section, we show the experimental results of our proposed HIT w.r.t different tile-sizes. Here we consider seven different tile-schemes including 1×1, 2×2, 3×3, 4×4, 5×5, 6×6 and 7×7, and the tile-sizes thereby are 300×300, 150×150, 100×100, 75×75, 60×60, 50×50, 42×42, respectively. We will resize back to $299 \times 299 \times 3$ to match the size of raw images. The visualizations of these patches can be found in Appendix Sec. A.11.

In Fig. 5, we report the average attack success rates of ten models. The success rates increase very quickly at first and then keep stable after 4×4 tile-scheme. If we continue to increase the tile-size, the attack success rates may go down. The main reason might be that the distortion caused by the resizing operation. It indirectly blurs resultant tiled adversarial patches, thus reducing the available HFC. Compared to the other two geometric patterns, we find that circle patches always perform the best. For example, the success rate is up to **88.67%** when tile-size is 6×6. This result demonstrates that the attack ability of training-free perturbations can benefit from **repeating** property.

4.1.3 The Effect of Dense Pattern

To validate the effect of dense pattern, we analyze the average attack success rates w.r.t densities. Since the trends of different patterns are similar, we only discuss the results of circle patch whose tile-scheme is 6×6. Here we control the density from 1 to 12. For example, "2" denotes only two circles in the proto-pattern, and more visualizations can be found in Appendix Sec. A.11.

As shown in Fig. 5, the success rates increase rapidly at the beginning, then remain stable after the density exceeds 8, and reach the peak at 12. This experiment demonstrates the effectiveness of **dense** pattern. Therefore, we set the default density of each proto-pattern to 12 in our paper.

4.1.4 The Size of Perturbation

In this section, we study the influence of the maximum perturbation ε on the performance of our HIT. The result of Fig. 6 depicts the growth trends of each model under different adversarial patches. No matter what the adversarial patch is, the performance proliferates at first, then remains stable after ε exceeds 16 for most models. Besides, the circle patch (curve-like) always performs best while the performance of the other two adversarial patches (straight-like) is similar. For example, when $\varepsilon = 16$ and the target model is VGG19, the attack success rate of circles patch is **94.75%** while the square patch and rhombuses patch ones are 81.52% and 83.63%, respectively. This demonstrates that DNNs are more vulnerable to curve-like perturbations than straight-like ones (we also analyze the reasons for this in Appendix Sec. A.10).

Another observation from this result is that our HIT can serve as a universal attack, although not in a strict sense. As demonstrated in Fig. 6, when $\varepsilon = 10$ which is the common constraint for universal adversarial perturbations (Mopuri et al., 2017; Moosavi-Dezfooli et al., 2017; Mopuri et al., 2018; Reddy Mopuri et al., 2018; Liu et al., 2019; Hashemi et al., 2020), our HIT with circle patch can achieve a success rate of **63.23%** on average. Notably, it can be up to **89.74%** on Squeeze.

Table 1: The comparison of attack success rates (%) between state-of-the-art no-box attacks and ours with the maximum perturbation $\varepsilon = 25.5$ (Sup. means supervised mechanism).

Attack	VGG19	Inc-v3	ResNet	DenseNet	WRN	SENet	PNA	Shuffle	Squeeze	Mobile	Avg.
Naïve‡ w/o Sup. (Li et al., 2020a)	54.08	36.06	39.36	43.52	41.20	34.46	26.86	-	-	62.24	42.22
Jigsaw w/o Sup. (Li et al., 2020a)	68.46	49.72	53.76	57.62	48.76	40.94	37.68	-	-	74.76	53.96
Rotation w/o Sup. (Li et al., 2020a)	68.86	51.86	52.60	58.74	49.28	41.80	40.06	-	-	74.00	54.65
Naïve† w/ Sup. (Li et al., 2020a)	23.80	19.14	16.24	21.06	15.84	13.00	13.04	-	-	27.56	18.71
Prototypical w/ Sup. (Li et al., 2020a)	80.22	63.54	62.08	70.84	62.72	55.44	51.42	-	-	82.22	66.06
Prototypical* w/ Sup. (Li et al., 2020a)	81.26	66.32	65.28	73.94	66.86	57.64	54.98	-	-	83.66	68.74
Beyonder w/ Sup. (Li et al., 2020a)	75.04	48.88	69.40	72.88	66.06	56.22	48.20	-	-	72.98	63.71
HIT w/ Square (Ours)	95.94	91.67	89.58	94.09	91.72	87.67	84.85	98.40	98.86	97.71	93.05
HIT w/ Rhombus (Ours)	96.77	91.92	91.51	96.54	88.83	86.86	78.24	99.27	99.18	99.40	92.85
HIT w/ Circle (Ours)	**99.30**	**98.92**	**98.55**	**98.69**	**96.53**	**93.61**	**97.02**	**99.56**	**99.63**	**99.52**	**98.13**

Table 2: The comparison of attack success rates (%) on normally trained models between black-box attacks ("*" denotes white-box attack) and our no-box attacks with the maximum perturbation $\varepsilon = 16.0$.

Model	Attack	VGG19	Inc-v3	ResNet	Dense	WRN	SENet	PNA	Squeeze	Shuffle	Mobile	Avg.
VGG19	MI-FGSM	99.96*	23.92	30.82	54.54	28.94	36.81	32.86	69.87	47.15	58.50	42.60
	DI^2-FGSM	99.96*	14.29	27.80	47.95	24.53	32.93	23.19	40.08	27.90	53.01	32.41
	PI-FGSM	99.95*	36.22	36.46	55.39	39.40	35.28	50.84	81.24	60.26	69.89	51.66
	PI-MI-DI^2-FGSM	99.96*	47.23	59.01	80.39	57.05	65.17	55.25	83.20	63.92	83.49	66.08
Inc-v3	MI-FGSM	42.58	99.92*	33.95	42.30	33.43	27.57	41.93	68.34	51.22	53.05	43.82
	DI^2-FGSM	33.91	99.33*	24.34	32.69	21.83	19.18	30.39	35.90	29.35	34.87	29.16
	PI-FGSM	51.77	99.91*	35.56	50.44	38.67	31.78	52.07	78.34	58.96	62.53	51.12
	PI-MI-DI^2-FGSM	68.27	99.76*	56.64	70.09	57.53	51.52	61.86	80.76	67.26	74.01	65.33
ResNet	MI-FGSM	63.75	41.71	99.98*	72.93	85.27	49.99	46.56	75.86	65.63	72.40	63.79
	DI^2-FGSM	76.90	41.22	99.95*	82.16	88.35	60.23	44.73	58.88	60.24	76.22	65.44
	PI-FGSM	64.88	48.16	99.98*	68.92	79.49	45.23	61.37	82.94	71.18	76.32	66.50
	PI-MI-DI^2-FGSM	92.71	77.90	99.99*	**96.04**	**97.80**	**86.09**	78.00	90.75	86.76	93.51	**88.84**
Dense	MI-FGSM	76.89	46.00	69.86	99.98*	67.76	53.05	48.69	78.55	69.63	78.42	65.43
	DI^2-FGSM	81.14	35.96	69.39	99.98*	64.64	48.53	40.03	60.50	55.68	73.03	58.77
	PI-FGSM	74.55	52.09	61.22	99.98*	63.12	49.84	60.09	85.79	74.74	82.37	67.09
	PI-MI-DI^2-FGSM	94.46	74.76	**90.88**	99.99*	89.42	80.99	73.86	91.00	85.34	93.93	86.07
-	HIT w/ Square (Ours)	81.52	59.84	58.75	71.13	64.13	55.51	53.71	93.78	90.04	89.51	71.79
	HIT w/ Rhombus (Ours)	83.63	61.69	65.12	77.03	62.32	54.58	49.80	95.10	94.76	96.20	74.02
	HIT w/ Circle (Ours)	**94.75**	**90.37**	87.62	88.81	79.26	70.31	**82.12**	**98.31**	**97.34**	**97.81**	88.67

4.2 Comparison of HIT with No-box Attacks

In this section, we compare the performance of our no-box HIT with state-of-the-art no-box attacks (Li et al., 2020a). Note that Li et al. (2020a) need to pay 15,000 iterations at most to train a substitute model, and then runs extra 200 iterations baseline attacks and 100 iterations ILA (Huang et al., 2019), which is extremely time-consuming. Significantly different from Li et al. (2020a), our HIT is training-free which does not require any auxiliary images to train a substitute model, thus achieving real-time attack.

The experimental results are reported in Tab. 1. A first glance shows that our HIT outperforms Li et al. (2020a) by a large margin. No matter what the adversarial patches are, our HIT can consistently achieve a success rate of over **92%** on average. By contrast, the best performance of Li et al. (2020a), *i.e.*, Prototypical* w/ Sup, is only 68.74% on average. Notably, our HIT with circle patch remarkably outperforms Li et al. (2020a) by **29.39%** on average and **42.04%** at most when attacking PNA.

4.3 Comparison of HIT with Black-box Attacks

In this section, we compare our no-box HIT with mainstream transfer-based attacks. For MI-FGSM, DI^2-FGSM, PI-FGSM and their extensions PI-MI-DI^2-FGSM, we utilize VGG19, Inc-v3, ResNet

and Dense to iteratively (ten forward & backward propagation) craft adversarial examples and use them to attack the rest of black-box models. As for our proposed HIT, we do not need any substitute model or training process. The results are summarized in Tab. 2, where the models in the leftmost column are substitute models, and the bottom block shows the results of our HIT.

As demonstrated in Tab. 2, our HIT is even on par with state-of-the-art PI-MI-DI^2-FGSM. Specifically, on average, the best performance of PI-MI-DI^2-FGSM is 88.84%, and our HIT based on circle patch can get up to **88.67%**. However, the transferability of adversarial examples largely depends on the substitute model. For example, when adversarial examples are crafted via Inc-v3, the performance of PI-MI-DI^2-FGSM is limited and our HIT can remarkably outperform it by **23.34%** on average. Besides, when the target model is in lightweight models, *e.g.*, Shuffle, our method consistently outperforms these mainstream transfer-based attacks by a large margin.

Table 3: The comparison of attack success rates (%) on defense models between black-box attacks (adversarial examples are crafted via an ensemble of VGG19, Inc-v3, ResNet and Dense) and our no-box attacks with the maximum perturbation $\varepsilon = 16.0$.

Model	Attack	Inc-v3$_{ens3}$	Inc-v3$_{ens4}$	IncRes$_{ens}$	Res152$_B$	Res152$_D$	ResNeXt$_{DA}$	Avg.
-	Raw	2.68	3.11	0.84	14.52	11.50	8.72	6.90
VGG19, Inc-v3, ResNet, Dense	TI-FGSM	22.69	22.62	16.35	16.77	13.09	11.16	17.11
	DI^2-FGSM	18.38	15.90	9.30	15.42	12.13	9.45	13.43
	PI-FGSM	34.21	33.66	22.29	17.25	13.62	11.19	22.04
	PI-TI-DI^2-FGSM	**70.04**	**69.43**	**56.37**	18.47	14.58	12.33	**40.20**
-	HIT w/ Square (Ours)	40.54	38.74	34.36	20.07	15.97	13.52	27.20
	HIT w/ Rhombus (Ours)	47.93	42.06	36.02	20.60	16.23	13.41	29.38
	HIT w/ Circle (Ours)	61.13	61.86	47.72	**20.68**	**16.45**	**13.64**	36.91

Since adversarial training technique (Madry et al., 2018b; Tramèr et al., 2018; Awasthi et al., 2021) can effectively defend against adversarial examples, we conduct an extra experiment on several defense models to demonstrate the effectiveness of our method. The additional target models including three ensemble adversarial training models (EAT) (Tramèr et al., 2018): Inc-v3$_{ens3}$, Inc-v3$_{ens4}$ and IncRes-v2$_{ens}$, and three feature denoising models (FD) (Xie et al., 2019a): ResNet152 Baseline (Res152$_B$), ResNet152 Denoise (Res152$_D$) and ResNeXt101 DenoiseAll (ResNeXt$_{DA}$). As demonstrated in previous works(Guo et al., 2019; Sharma et al., 2019), low-frequency perturbations are more effective for attacking defense models. Motivated by it, we change the tile-schemes to smaller ones (*i.e.*, 2×2 for EAT and 1×1 for FD) and other parameters stay the same (see more details in Appendix Sec. A.12). As observed in Tab. 3, our HIT is effective even for defense models. Notably, HIT based on circle patch can successfully attack Inc-v3$_{ens4}$ by **61.86%**. Besides, for more robust FD, even crafting adversarial examples via an ensemble of VGG19, Inc-v3, ResNet and Dense, transfer-based PI-TI-DI^2-FGSM is still inferior to our HIT. This experimental result reveals that current defenses have not achieved real security, which is even vulnerable to training-free adversarial examples.

5 Conclusion

In this paper, we rethink the classification logic of deep neural networks with respect to adversarial examples. We observe that HFC domains in low-level features and plays a crucial role in classification. Besides, we demonstrate that DNNs are vulnerable to training-free perturbations with regionally homogeneous, repeating, dense property through empirically and experimentally analysis. Motivated by these observations, we propose a novel Hybrid Image Transformation (HIT) attack method by combining the LFC of raw images with the HFC of our well-designed adversarial patches to destroy the useful features and add strong irrelevant noisy ones. Extensive experiments on the ImageNet dataset demonstrate the effectiveness of the proposed method. Surprisingly, our simple method outperforms existing no-box attacks by a significant margin and is even on par with transfer-based black-box attacks that require the substitute model to craft adversarial examples.

In another aspect, since most models are vulnerable to our method, it implies that our adversarial examples may capture the common "blind spots" of them. Therefore, a defense can improve the robustness and stability by covering these "blind spots", *i.e.*, applying data augmentation technique using our adversarial examples.

REFERENCES

Pranjal Awasthi, George Yu, Chun-Sung Ferng, Andrew Tomkins, and Da-Cheng Juan. Adversarial robustness across representation spaces. In *CVPR*, 2021.

Ayse Elvan Aydemir, Alptekin Temizel, Tugba Taskaya Temizel, et al. The effects of jpeg and jpeg2000 compression on attacks using adversarial examples. *arXiv preprint arXiv:1803.10418*, 2018.

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

Tom B. Brown, Dandelion Mané, Aurko Roy, Martín Abadi, and Justin Gilmer. Adversarial patch. *CoRR*, abs/1712.09665, 2017.

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

Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *ACM workshop on artificial intelligence and security*, 2017.

Yue Chen, Yalong Bai, Wei Zhang, and Tao Mei. Destruction and construction learning for fine-grained image recognition. In *CVPR*, 2019.

Nilaksh Das, Madhuri Shanbhogue, Shang-Tse Chen, Fred Hohman, Siwei Li, Li Chen, Michael E. Kounavis, and Duen Horng Chau. Shield: Fast, practical defense and vaccination for deep learning using jpeg compression. In *KDD*, 2018.

Salah Ud Din, Naveed Akhtar, Shahzad Younis, Faisal Shafait, Atif Mansoor, and Muhammad Shafique. Steganographic universal adversarial perturbations. *Pattern Recognit. Lett.*

Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *CVPR*, 2018.

Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *CVPR*, 2019a.

Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, and Jun Zhu. Efficient decision-based black-box adversarial attacks on face recognition. In *CVPR*, 2019b.

Aditya Ganeshan and R Venkatesh Babu. Fda: Feature disruptive attack. In *ICCV*, 2019.

Lianli Gao, Qilong Zhang, Jingkuan Song, Xianglong Liu, and Hengtao Shen. Patch-wise attack for fooling deep neural network. In *ECCV*, 2020a.

Lianli Gao, Qilong Zhang, Jingkuan Song, and Heng Tao Shen. Patch-wise++ perturbation for adversarial targeted attacks. *CoRR*, abs/2012.15503, 2020b.

Lianli Gao, Yaya Cheng, Qilong Zhang, Xing Xu, and Jingkuan Song. Feature space targeted attacks by statistic alignment. In *IJCAI*, 2021.

Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained cnns are biased towards texture; increasing shape bias improves accuracy and robustness. In *ICLR*, 2019.

Ian J Goodfellow, Jonathon Shlens, Christian Szegedy, et al. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

Chuan Guo, Jared S. Frank, and Kilian Q. Weinberger. Low frequency adversarial perturbation. In Amir Globerson and Ricardo Silva (eds.), *UAI*, 2019.

Atiye Sadat Hashemi, Andreas Bär, Saeed Mozaffari, and Tim Fingscheidt. Transferable universal adversarial perturbations using generative models. *arXiv preprint arXiv:2010.14919*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

Jie Hu, Li Shen, Gang Sun, et al. Squeeze-and-excitation networks. In *CVPR*, 2018.

Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017.

Qian Huang, Isay Katsman, Horace He, Zeqi Gu, Serge Belongie, and Ser-Nam Lim. Enhancing adversarial example transferability with an intermediate level attack. In *ICCV*, 2019.

Forrest N. Iandola, Matthew W. Moskewicz, Khalid Ashraf, Song Han, William J. Dally, and Kurt Keutzer. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. In *ICLR*, 2017.

Nathan Inkawhich, Wei Wen, Hai Helen Li, and Yiran Chen. Feature space perturbations yield more transferable adversarial examples. In *CVPR*, 2019.

Jason Jo and Yoshua Bengio. Measuring the tendency of cnns to learn surface statistical regularities. *arXiv preprint arXiv:1711.11561*, 2017.

Jonathan Krause, Jia Deng, Michael Stark, and Li Fei-Fei. Collecting a large-scale dataset of fine-grained cars. 2013.

Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial machine learning at scale. In *ICLR*, 2017.

Qizhang Li, Yiwen Guo, Hao Chen, et al. Practical no-box adversarial attacks against dnns. In *NeurIPS*, 2020a.

Yingwei Li, Song Bai, Cihang Xie, Zhenyu Liao, Xiaohui Shen, and Alan L Yuille. Regional homogeneity: Towards learning transferable universal adversarial perturbations against defenses. In *ECCV*, 2020b.

Yingwei Li, Song Bai, Yuyin Zhou, Cihang Xie, Zhishuai Zhang, and Alan L. Yuille. Learning transferable adversarial examples via ghost networks. In *AAAI*, 2020c.

Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *ICLR*, 2020.

Aishan Liu, Jiakai Wang, Xianglong Liu, Bowen Cao, Chongzhi Zhang, and Hang Yu. Bias-based universal adversarial patch attack for automatic check-out. In *ECCV*, 2020.

Chenxi Liu, Barret Zoph, Jonathon Shlens, Wei Hua, Li-Jia Li, Li Fei-Fei, Alan L. Yuille, Jonathan Huang, and Kevin Murphy. Progressive neural architecture search. In *ECCV*, 2018.

Chihuang Liu and Joseph JaJa. Feature prioritization and regularization improve standard accuracy and adversarial robustness. In *IJCAI*, 2019.

Hong Liu, Rongrong Ji, Jie Li, Baochang Zhang, Yue Gao, Yongjian Wu, and Feiyue Huang. Universal adversarial perturbation via prior driven uncertainty approximation. In *ICCV*, 2019.

Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet V2: practical guidelines for efficient CNN architecture design. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss (eds.), *ECCV*, 2018.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018a.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018b.

Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew B. Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. volume abs/1306.5151, 2013.

Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *CVPR*, 2017.

Konda Reddy Mopuri, Utsav Garg, R. Venkatesh Babu, et al. Fast feature fool: A data independent approach to universal adversarial perturbations. In *BMVC*, 2017.

Konda Reddy Mopuri, Aditya Ganeshan, Venkatesh Babu Radhakrishnan, et al. Generalizable data-free objective for crafting universal adversarial perturbations. *TPAMI*, 2018.

Nina Narodytska and Shiva Prasad Kasiviswanathan. Simple black-box adversarial attacks on deep neural networks. In *CVPR Workshops*, volume 2, 2017.

Anh Mai Nguyen, Jason Yosinski, Jeff Clune, et al. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, 2015.

Aude Oliva. The art of hybrid images: Two for the view of one. *Art & Perception*, 1(1-2):65–74, 2013.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, et al. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016.

Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *ACM on Asia conference on computer and communications security*, 2017.

Konda Reddy Mopuri, Phani Krishna Uppala, R Venkatesh Babu, et al. Ask, acquire, and attack: Data-free uap generation using class impressions. In *ECCV*, 2018.

Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael S. Bernstein, Alexander C. Berg, and Fei-Fei Li. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015.

Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018.

Yash Sharma, Gavin Weiguang Ding, and Marcus A. Brubaker. On the effectiveness of low frequency perturbations. In Sarit Kraus (ed.), *IJCAI*, 2019.

Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In Yoshua Bengio and Yann LeCun (eds.), *ICLR*, 2015.

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

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016.

Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *USENIX*, 2016.

Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian J. Goodfellow, Dan Boneh, and Patrick D. McDaniel. Ensemble adversarial training: attacks and defenses. In *ICLR*, 2018.

C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, California Institute of Technology, 2011.

Haohan Wang, Xindi Wu, Zeyi Huang, and Eric P. Xing. High-frequency component helps explain the generalization of convolutional neural networks. In *CVPR*, 2020.

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 *ICLR*, 2020.

Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L. Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *CVPR*, 2019a.

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 *CVPR*, 2019b.

Ziang Yan, Yiwen Guo, Changshui Zhang, et al. Subspace attack: Exploiting promising subspaces for query-efficient black-box attacks. *NeurIPS*, 2019.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016.

Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars (eds.), *ECCV*, 2014.

Chaoning Zhang, Philipp Benz, Tooba Imtiaz, and In-So Kweon. Understanding adversarial examples from the mutual influence of images and perturbations. In *CVPR*, 2020.

Mingyi Zhou, Jing Wu, Yipeng Liu, Shuaicheng Liu, and Ce Zhu. Dast: Data-free substitute training for adversarial attacks. In *CVPR*, 2020.

Wen Zhou, Xin Hou, Yongjun Chen, Mengyun Tang, Xiangqi Huang, Xiang Gan, and Yong Yang. Transferable adversarial perturbations. In *ECCV*, 2018.

A APPENDIX

A.1 SETUP

Networks. Here we consider ten well-known classification models: VGG19 Simonyan & Zisserman (2015), Inception-v3 (Inc-v3) Szegedy et al. (2016), ResNet-152 (ResNet) He et al. (2016), DenseNet-121 (Dense) Huang et al. (2017), WideResNet (WRN) Zagoruyko & Komodakis (2016), SENet Hu et al. (2018), PNASNet (PNA) Liu et al. (2018), ShuffleNet-v2 (Shuffle) Ma et al. (2018), SqueezeNet (Squeeze) Iandola et al. (2017) and MobileNet-v2 (Mobile) Sandler et al. (2018) as our target models.

Dataset. To make our method more convincing and avoid cherry-picking, we choose 10,000 images (each category contains about 10 images) from the ImageNet validation set Russakovsky et al. (2015) which are classified correctly by all ten networks we consider. Besides, all images are resized to $299 \times 299 \times 3$ beforehand.

Parameters. In our experiments, we use l_∞-norm to measure the perceptibility of adversarial noises, the maximum perturbation ε is set to 16. For our HIT, the size of Gaussian kernel $\boldsymbol{G}$ is 17×17 (*i.e.* $k = 4$), weight factor λ is set to 1.0.

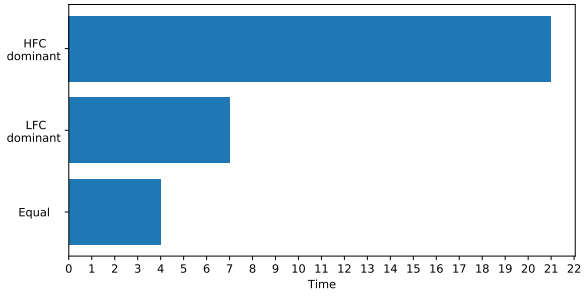


Figure 7: We compare the average responses of HFC with LFC for each feature map. "HFC dominant" means the average responds of HFC is higher than LFC, and "LFC dominant" is vice versa.

A.2 QUANTITATIVE ANALYSIS ABOUT HFC AND LFC

To quantitatively analyze whether HFC or LFC is dominant in the feature map of shallow layer, we conducted this experiment. Considering that the size of each shallow-layer feature map in Fig. 2(b) is 147, we first resize the Fig. 2(a) (299×299) to 147×147, and denote the resultant image by $\boldsymbol{x_r}$. Then we get the $\boldsymbol{x_r^H}$ (HFC of $\boldsymbol{x_r}$) by:

$$\boldsymbol{x_r^H} = \boldsymbol{x_r} - \boldsymbol{x_r} * \boldsymbol{G}. \tag{5}$$

To quantitatively compare the response of HFC and LFC, we calculate the average response of each feature map $\phi(\boldsymbol{x})$ in low-frequency regions versus that in the other HFC regions. To that end, we generate two masks to distinguish the two regions. More specific, the mask of high-frequency regions $\boldsymbol{M^H}$ can be written as:

$$\boldsymbol{M_{i,j}^H} = \begin{cases} 1, & |\boldsymbol{x_{r(i,j)}^H}| > \tau \\ 0, & else \end{cases}, \tag{6}$$

where $\tau = 20$ is the pre-set threshold which applied to filter out low response. After getting the $\boldsymbol{M^H}$, the mask of LFC $\boldsymbol{M^L}$ can be easy derivated:

$$\boldsymbol{M^L} = 1 - \boldsymbol{M^H}. \tag{7}$$

Therefore, the average response of HFC a^H and the average response of LFC a^L can be expressed as:

$$\boldsymbol{a^H} = \frac{\sum_{i,j} \boldsymbol{M^H} \odot \phi(\boldsymbol{x})}{\sum_{i,j} \boldsymbol{M^H}}, \tag{8}$$

$$\boldsymbol{a^L} = \frac{\sum_{i,j} \boldsymbol{M^L} \odot \phi(\boldsymbol{x})}{\sum_{i,j} \boldsymbol{M^L}}. \tag{9}$$

In this paper, if a feature map meets $\boldsymbol{a^H} > \boldsymbol{a^L}$, we call it "HFC dominant", otherwise we call it "LFC dominant". As demonstrate in Fig. 7, most feature maps are focus on HFC, and the "HFC dominant" to "LFC dominant" ratio is **3:1**.

A.3 The Effect of Weight Factor λ

In this section, we discuss the effect of different weight factors λ on the experimental results. We tune λ from 0.1 to 10, and the results are shown in Fig. 8. When $\lambda \leq 1$, the attack success rate increases rapidly at the beginning and then remains stable. However, further increasing λ from 1 to 10 does not improve the performance. Actually, the success rates keep stable with a slight drop.

Apparently, a larger λ leads to more perturbations (*i.e.*, increase the average perturbation of all pixels), and our reported results are a little inconsistent with linear assumption (Goodfellow et al., 2015). It probably because our HIT is completely independent of any prior information (*e.g.* the gradient of any model or data distribution). So it is not the larger the noise is, the farther the deviation from the true label will be. Besides, we notice that the activation functions of these victim's models are all *Relu*, which may be another reason for this phenomenon. More specifically, *Relu* is defined as

$$Relu(\boldsymbol{z}) = \begin{cases} 0, & \boldsymbol{z} < 0. \\ \boldsymbol{z}, & else. \end{cases} \tag{10}$$

where $\boldsymbol{z}$ is the intermediate output before activation layer. If the intermediate adversarial perturbation is large enough, *i.e.*, $\boldsymbol{\delta'} \leq -\boldsymbol{z}$, then $Relu(\boldsymbol{z} + \boldsymbol{\delta'})$ will return 0. But for a misclassification label $y^{adv} \neq y$, positive activation which is different from the original $\boldsymbol{z}$ may be more helpful than 0.

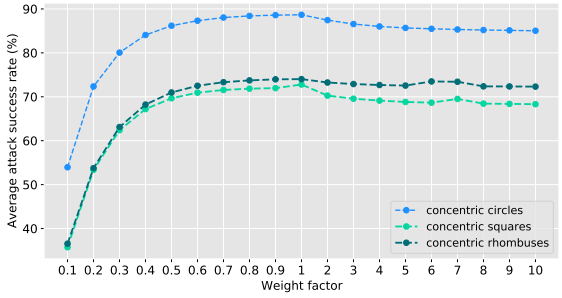


Figure 8: The average attack success rates (%) of different adversarial patches against NTs w.r.t weight factors.

A.4 Qualitative Comparison for Adversarial Examples

To better reflect the advantages of our approach, in this section we compare the visual quality of the generated adversarial examples. Specifically, we consider state-of-the-art black-box PI-FGSM (Gao et al., 2020a) and no-box attack (Li et al., 2020a) as our competitor. As depicted in Fig. 9, both PI-FGSM (Gao et al., 2020a) and no-box attack (Li et al., 2020a) will cause more perceptible distortions. In contrast, the adversarial perturbation crafted by our HIT is much more imperceptible.

A.5 Attack Other Classification Tasks

To highlight the practical property of our HIT, in this section we apply our HIT for other classification tasks. Specifically, we consider three well-known fine-grained classification including

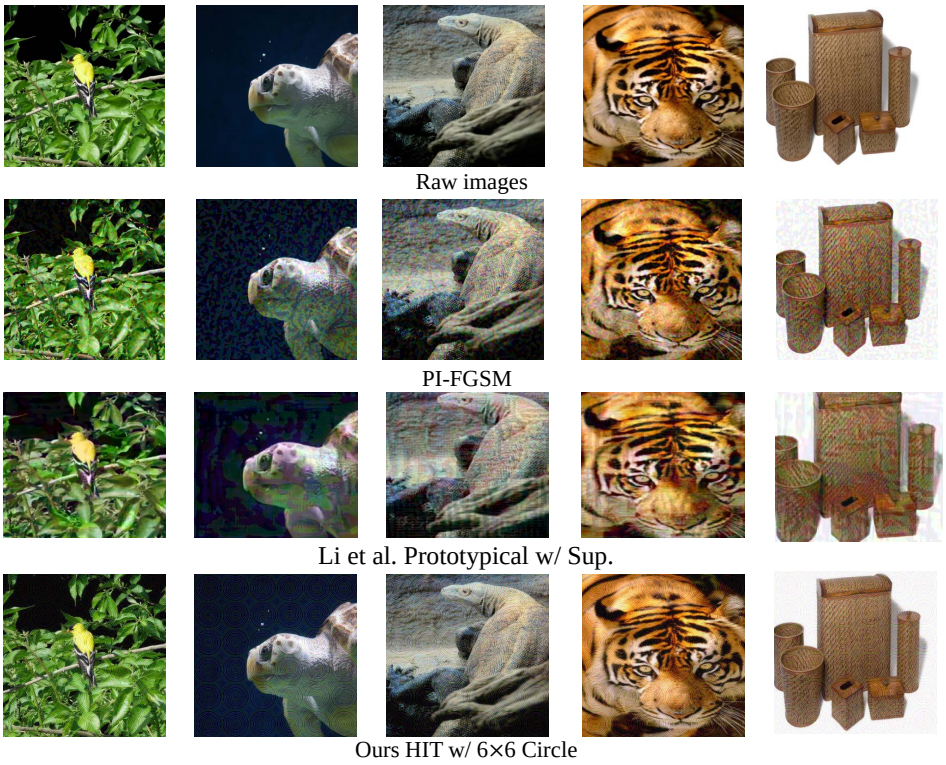


Figure 9: Qualitative comparison for adversarial examples crafted by different methods. The maximum perturbation ε is 16.

CUB-200-2011 (Wah et al., 2011), Stanford Cars (Krause et al., 2013) and FGVC Aircraft (Maji et al., 2013) and the victim model is trained via DCL (backbone: Res-50) (Chen et al., 2019). The resolution of inputs is 448×448. Therefore, we set the "tile size = 448 / tile scheme". For example, if the tile-scheme is 4×4, then the tile-size is 112. To ensure our defaulting setting (*i.e.*, λ and tile-scheme) for HIT is applicable, we conduct two experiments in the following.

Table 4: Average attack success rates of HIT (w/ Circle) w.r.t tile-schemes. The maximum perturbation $\varepsilon = 16$.

Attacks	CUB-200-2011	Standford Cars	FGVC aircraft
1×1	30.07	28.46	43.81
2×2	38.87	54.11	55.47
3×3	51.22	81.08	73.16
4×4	55.19	80.18	74.04
5×5	59.23	84.94	70.95
6×6	67.30	88.25	**76.71**
7×7	**70.19**	**88.46**	74.69

Discussion on tile scheme. We first report the average attack success rates (%) of our HIT w/ Circle w.r.t tile-scheme in Tab. 4. From the result, we can observe that our HIT is also effective for attacking other datasets. Notably, our HIT can fool DCL with about 90% success rate on Stanford Cars dataset. Besides, a relatively smaller tile-size is also helpful in improving the success rate of the attack, which is consistent with the conjecture given in Sec. 3.2.

Discussion on λ. We then report the average attack success rate (%) of our HIT w/ Circle w.r.t λ in Tab. 5. Although set $\lambda = 1.0$ is not optimal, the gap between the best results and the results of is very small. Therefore, our default setting for HIT is still applicable.

A.6 Attack Real-world Recognition System

Table 5: Average attack success rates of HIT (w/ Circle) w.r.t λ. The maximum perturbation $\varepsilon = 16$.

λ	CUB-200-2011	Standford Cars	FGVC aircraft
0.2	51.90	63.72	37.92
0.4	67.18	83.92	64.03
0.6	**68.31**	87.55	71.96
0.8	67.97	**88.36**	75.34
1.0	67.26	88.25	76.73
1.2	66.54	87.99	**76.77**
1.4	66.13	87.66	76.41
1.6	65.88	87.28	76.28
1.8	65.46	87.06	76.08
2.0	65.22	86.93	75.76

To further demonstrate the practical property of our HIT, in this section we apply our HIT (w/ Circle) to attack a real-world recognition system, *i.e.*, Google Cloud Vision API[6]. Different from existing works (Chen et al., 2017; Brendel et al., 2017) which need a large number of queries for optimization, we directly apply our HIT with the default setting (*i.e.*, tile-scheme is 6×6 and $\lambda = 1.0$). As illustrated in Fig. 10, our no-box HIT with $\varepsilon = 16$ can effectively change top-k labels. For example, the top-5 label of "fish" is "Fish", "Fin", "Seafood", "Ray-finned fish" and "Marine biology", while our adversarial example is "Reptile","Turtle", "Terrestrial Animal", "Pattern" and "Art". Notably, there is no overlap on top-k labels between clean image and our adversarial example, which also demonstrate the effectiveness of our no-box HIT.

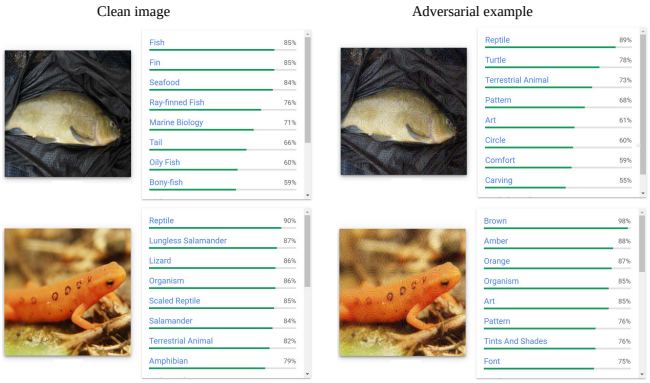


Figure 10: The results for attack Google Cloud Vision API. The maximum perturbation ε is 16.

A.7 Results for Smaller Perturbation

In this experiment, we report the average success rates (%) between state-of-the-art black-box attacks (further add Ghost Network algorithm (Li et al., 2020c) as our competitor) and our proposed no-box attack with a smaller perturbation $\varepsilon = 8$.

As demonstrated in Tab. 6, our proposed methods are still competitive to mainstream transfer-based black-box attacks, even though they combine many effective techniques. Remarkably, our no-box attack can significantly outperform Ghost Networks (+MI-FGSM). Although Ghost Networks (+PI-MI-DI-FGSM) is much more powerful, our no-box attack can surpass it in some cases. For example, when fooling Shuffle, our HIT (w/ Circle) can outperform Ghost Networks (+PI-MI-DI-FGSM) by about 8%.

A.8 Raw Image for Attack

To highlight the effectiveness of our design for adversarial patches, here we conduct the experiment where raw images (shown in Fig. 11) serve as "adversarial patch". More specially, we utilize the HFC of these raw images (like Din et al.) to manipulate adversarial examples. However, even the HFC of texture-rich raw images (*e.g.*, "Grifola frondosa" and "Capitulum") do not achieve a good result. As demonstrated in Tab. 8, the average attack success rates are all less than 40%. By contrast, our well-designed adversarial patches can significantly achieve a success rate of nearly **90%**, which demonstrates the effectiveness of our design.

[6] https://cloud.google.com/vision/docs/drag-and-drop

Robin Norfolk terrier Grifola frondosa Capitulum

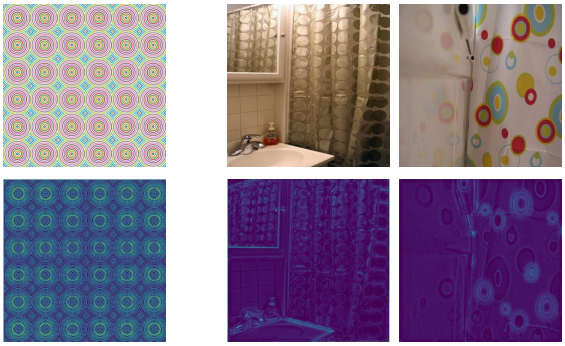
Figure 11: We randomly selected four raw images from ImageNet dataset Russakovsky et al. (2015) to replace our adversarial patches, then test the attack performance by our HIT.

Table 6: The comparison of attack success rates (%) on normally trained models between black-box attacks and our no-box attacks with maximum perturbation $\varepsilon = 8$. For black-box attacks, adversarial examples are crafted via Inc-v3.

Attacks	VGG19	ResNet	DenseNet	WRN	SENet	PNA	Shuffle	Squeeze	Mobile	AVG.
MI-FGSM	21.26	15.63	20.47	14.45	11.58	23.36	24.13	32.61	25.64	21.01
Ghost Networks (+MI-FGSM)	27.12	17.45	22.31	14.92	13.43	28.63	30.08	40.80	33.67	25.38
DI-FGSM	18.41	11.24	16.44	10.19	8.28	17.59	15.03	18.40	18.20	14.86
PI-FGSM	24.12	14.98	22.91	15.38	12.21	27.32	25.93	39.20	27.29	23.26
PI-MI-DI^2-FGSM	40.88	31.02	41.34	29.70	25.56	38.46	36.79	46.70	42.38	36.98
Ghost Networks (+PI-MI-DI^2-FGSM)	**63.56**	**43.59**	**55.21**	**40.91**	**40.33**	**54.73**	63.48	**81.98**	**73.46**	**57.47**
HIT w/ 6×6 Circle	37.64	36.21	40.80	30.82	21.36	37.63	**71.45**	79.34	69.90	47.24
HIT w/ 6×6 Square	13.40	17.50	25.85	21.99	13.79	18.31	53.03	61.61	50.88	30.71
HIT w/ 6×6 Rhombus	17.24	19.95	27.65	22.00	12.85	18.91	58.84	63.49	57.90	33.20

A.9 Discussion on Targeted Attack

Although we do not explicitly force the resultant adversarial examples to be misclassified as a specific targeted label, we observe that our HIT tends to implement a targeted attack due to the frequency domain operation and classification logic of DNNs. In Tab. 7, we report the top-5 prediction labels of our adversarial examples, which are crafted by 6×6 concentric circle pattern. A first glance shows that almost all models tend to misclassify adversarial examples generated by our HIT as several specific labels, *e.g.*, 794 ("shower curtain"). Furthermore, this phenomenon is more obvious for Mobile and ResNet whose ratio is up to **47.69%** and **75.75%** respectively.



(a) Adversarial Patch (b) Label ID: 794 ("shower curtain")

Figure 12: We show (a) our adversarial patch and (b) some images which classified as shower curtain from ImageNet dataset. The bottom row is their HFC extracted by Eq. 1.

To better understand this phenomenon, we show several clean images whose labels are "shower curtain" from the ImageNet dataset and our adversarial patch in Fig. 12. We observe that the HFC of "shower curtain" is somehow aligned with our adversarial patch, *i.e.*, they all show similar certain repetitive circles. We suspect this phenomenon might be because our proposed perturbation dominates the overall features of the image, and instead, the original features of the image become noise. Since existing algorithms are not effective yet and simply replacing our adversarial patch with a clean targeted image does not achieve an effective targeted attack (as demonstrated in Sec. A.8), we will further study the selection and generation of adversarial patches, *e.g.*, fusing the shallow texture information of targeted distribution to guide the resultant adversarial examples towards the targeted category.

Table 7: The top-5 label IDs that appear in classification results (range from 0 to 999) after HIT attack (tile-size is 50×50, proto-pattern is concentric circle). The top row is victim's models, ratio (%) represents the proportion of a specific prediction label to the total number of misclassified adversarial examples.

	VGG19		Inc-v3		ResNet		DenseNet		WRN		SENet		PNA		Shuffle		Squeeze		Mobile	
	Label	Ratio	Label	Ratio	Label	Ratio	Label	Ratio	Label	Ratio	Label	Ratio	Ratio	Label	Ratio	Label	Ratio	Label	Ratio	Label
Top-1	815	27.86	**794**	**34.02**	**794**	**75.75**	84	38.97	815	27.04	**794**	**13.64**	**794**	**33.83**	879	20.55	455	35.11	**794**	**47.69**
Top-2	646	25.21	862	16.07	109	3.31	**794**	**17.34**	549	9.90	109	8.85	862	19.70	893	12.42	**794**	**25.26**	109	19.73
Top-3	506	16.50	750	8.23	854	3.06	884	4.63	721	5.83	721	8.65	549	5.93	721	9.92	109	9.11	885	11.57
Top-4	**794**	**10.75**	911	4.78	646	2.56	862	4.30	862	3.38	750	5.25	815	3.97	**794**	**8.37**	753	7.05	884	4.21
Top-5	868	3.15	109	4.65	750	2.11	506	3.46	921	3.37	549	4.15	700	2.95	109	7.20	854	4.42	854	3.24

Table 8: The comparison of attack success rates (%) w.r.t raw images

Attack	VGG19	Inc-v3	ResNet	Dense	WRN	SENet	PNA	Shuffle	Squeeze	Mobile	Avg.
HIT w/ Robin	27.37	19.78	16.62	18.72	17.07	11.73	25.70	37.34	48.77	33.47	25.66
HIT w/ Norfolk terrier	33.01	28.06	21.87	27.94	24.88	19.39	34.57	42.70	59.34	39.45	33.12
HIT w/ Grifola frondosa	37.93	29.25	25.38	33.02	28.36	19.78	41.77	46.10	61.30	43.17	36.61
HIT w/ Capitulum	46.54	26.82	29.00	37.18	30.49	30.14	41.77	41.00	55.78	51.36	39.01

A.10 Why Circle Pattern is Usually Better?

Here we attempt to provide an insight into the performance gap between Circle and the other two patterns by analyzing the intermediate feature response. Without loss of generality, we set the layer index to "depth of each DNN" / 2 and report the average cosine similarity of the features between 10,000 raw images and their adversarial examples. The result from Tab. 9 shows that Circle consistently leads to lower cosine similarity than other patterns. Consequently, the features that feed to the deep layer are more featureless, thus leading to misclassification.

A.11 Visualization of Our Adversarial Patches

In this section, we first visualize the concentric circle with respect to densities in Fig. 13. Here we control the density from 1 to 12, *e.g.*, "2" denotes only two circles in the proto-pattern. With the increase of density, the distance between any two circles will also be reduced.

Then we list our adversarial patches with respect to tile-schemes in Fig. 14. More specifically, we first crop the $600 \times 600 \times 3$ proto-patterns to $300 \times 300 \times 3$ adversarial patches, then resize them into different tile-sizes (*e.g.*, $150 \times 150 \times 3$) and tile them to $300 \times 300 \times 3$, finally resize back to $299 \times 299 \times 3$ to match the size of raw images. As we can see, if we decrease the tile-size, distortion is inevitable.

A.12 The Effect of Repeating Pattern for Defenses

In this section, we further consider six additional well-known defense models, which including three ensemble adversarial training models (EAT) (Tramèr et al., 2018): Inc-v3$_{ens3}$, Inc-v3$_{ens4}$ and IncRes-v2$_{ens}$,[7] and three feature denoising models (FD) (Xie et al., 2019a): ResNet152 Baseline (Res152$_B$), ResNet152 Denoise (Res152$_D$), ResNeXt101 DenoiseAll (ResNeXt$_{DA}$),[8] to discuss the effect of repeating pattern.

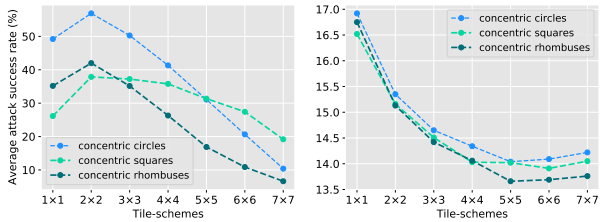


Figure 16: The average attack success rates (%) of different adversarial patches against EAT (left) and FD (right) w.r.t tile-schemes.

[7] https://github.com/tensorflow/models/tree/archive/research/adv_imagenet_models

[8] https://github.com/facebookresearch/ImageNet-Adversarial-Training

Table 9: The cosine similarity comparison for different patterns.

Model	Attack	VGG19	Inc-v3	ResNet	Dense	WRN	SENet	PNA	Squeeze	Shuffle	Mobile	Avg.
-	HIT w/ Square (Ours)	0.6215	0.7419	0.7638	0.8090	0.7599	0.5838	0.7437	0.6940	0.6704	0.4838	0.6872
	HIT w/ Rhombus (Ours)	0.6218	0.7458	0.7448	0.7853	0.7280	0.6258	0.7672	0.6738	0.6461	0.4005	0.6746
	HIT w/ Circle (Ours)	**0.5472**	**0.6685**	**0.7306**	**0.7779**	**0.7223**	**0.5613**	**0.6747**	**0.6643**	**0.6062**	**0.3617**	**0.6314**

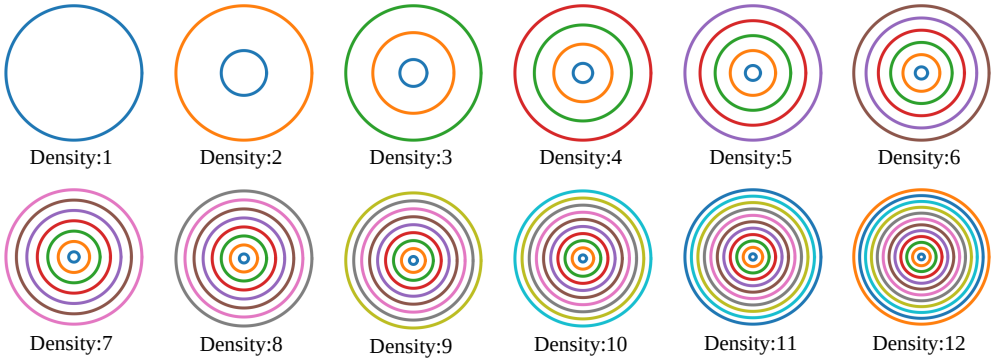


Figure 13: We visualize our proto-patterns w.r.t densities. Here we take concentric circles as an example.

Generally, a smaller tile-scheme can generate a more perceptible perturbation. As shown in Fig. 15, the area of each regionally homogeneous (*i.e.* continues) line in adversarial examples crafted by 1×1 patches is bigger than 6×6 ones. Different from the trends on NTs, smaller tile-schemes are more effective for attacking defense models. As demonstrated in Tab. 16, when attacking EAT, 2×2 adversarial patches perform best, and further increasing the tile-scheme will significantly degrade performance, *e.g.,* 7×7 rhombuses only successfully attack EAT by 6.61% on average. The trend of FD is similar to that of EAT, except that 1×1 adversarial patches work best. The reason might be that thin regionally homogeneous lines are more easily to filter out by the denoising block of (Xie et al., 2019a). Therefore, in our paper, we use 2×2 and 1×1 adversarial patches to attack EAT and FD, respectively.

Table 10: Average attack success rates of HIT (w/ Circle) w.r.t tile-schemes. "w/ LF" means adding our perturbations on LFC (i.e., reducing the HFC beforehand) and "w/o LF" means adding perturbations on benign samples. The maximum perturbation $\varepsilon = 16.0$.

	Vgg19	Inc-v3	ResNet	DenseNet	WRN	SENet	PNA	Shuffle	Squeeze	Mobile	Avg.
1x1 w/o LF	68.48	52.17	49.82	58.79	49.70	41.67	73.00	63.34	72.80	59.33	58.91
1x1 w/ LF	**73.95**	**59.94**	**54.39**	**64.38**	**56.33**	**50.16**	**73.53**	**68.28**	**80.15**	**66.52**	**64.76**
2x2 w/o LF	91.32	72.74	67.21	76.79	68.48	54.61	83.54	78.27	88.46	88.31	76.97
2x2 w/ LF	**92.89**	**77.33**	**71.29**	**80.42**	**72.80**	**60.17**	**85.21**	**82.45**	**91.54**	**90.32**	**80.44**
3x3 w/o LF	91.40	81.11	71.19	78.58	71.08	66.64	83.24	88.78	92.12	94.14	81.83
3x3 w/ LF	**92.55**	**85.42**	**76.44**	**82.84**	**76.03**	**70.80**	**86.64**	91.95	**94.10**	**95.17**	**85.19**
4x4 w/o LF	92.91	83.89	76.02	83.78	69.55	63.68	82.18	91.62	94.82	95.89	83.43
4x4 w/ LF	**93.97**	**88.28**	**82.69**	**88.34**	**76.85**	**69.27**	**86.18**	**94.93**	**96.65**	**96.81**	**87.40**
5x5 w/o LF	92.36	85.64	79.48	84.78	70.70	63.58	80.26	93.69	96.45	97.19	84.41
5x5 w/ LF	**94.42**	**89.81**	**85.92**	**89.44**	**78.40**	**69.15**	**84.95**	**96.67**	**97.75**	**97.97**	**88.45**
6x6 w/o LF	92.99	85.86	81.49	83.41	71.82	65.92	76.43	95.00	97.36	96.96	84.72
6x6 w/ LF	**94.75**	**90.37**	**87.62**	**88.80**	**79.26**	**70.31**	**82.12**	**97.34**	**98.31**	**97.81**	**88.67**

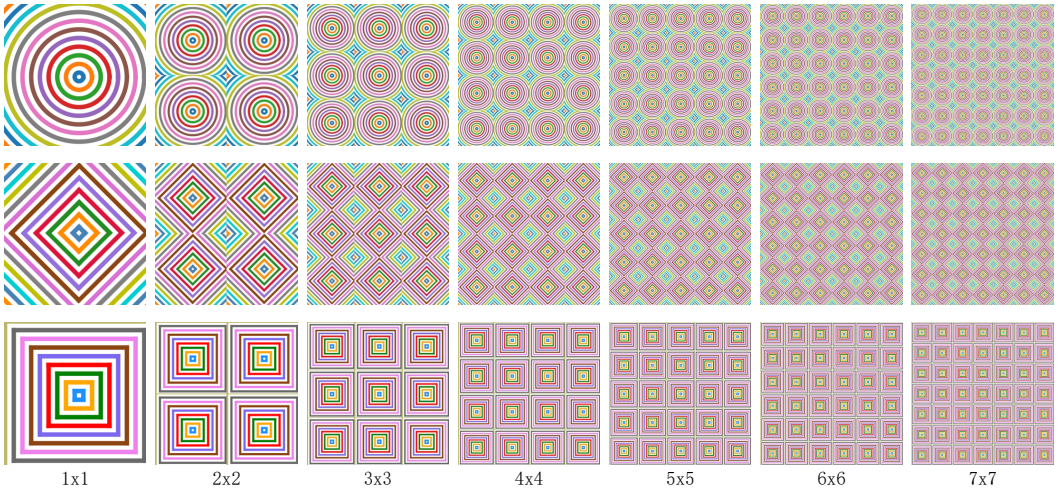


Figure 14: We visualize our adversarial patches w.r.t tile-schemes.

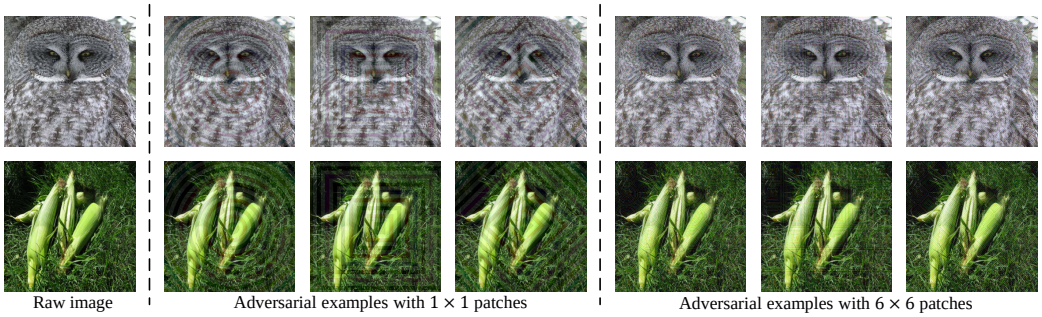


Figure 15: The visualization for resultant adversarial examples w.r.t tile-schemes.