StyLess: Boosting the Transferability of Adversarial Examples

Kaisheng Liang Bin Xiao*
The Hong Kong Polytechnic University
{cskliang, csbxiao}@comp.polyu.edu.hk

Abstract

Adversarial attacks can mislead deep neural networks (DNNs) by adding imperceptible perturbations to benign examples. The attack transferability enables adversarial examples to attack black-box DNNs with unknown architectures or parameters, which poses threats to many real-world applications. We find that existing transferable attacks do not distinguish between style and content features during optimization, limiting their attack transferability. To improve attack transferability, we propose a novel attack method called style-less perturbation (StyLess). Specifically, instead of using a vanilla network as the surrogate model, we advocate using stylized networks, which encode different style features by perturbing an adaptive instance normalization. Our method can prevent adversarial examples from using non-robust style features and help generate transferable perturbations. Comprehensive experiments show that our method can significantly improve the transferability of adversarial examples. Furthermore, our approach is generic and can outperform state-of-the-art transferable attacks when combined with other attack techniques. [1]

1. Introduction

Deep neural networks (DNNs) [14, 24] are currently effective methods for solving various challenging tasks such as computer vision, and natural language processing. Although DNNs have amazing accuracy, especially for computer vision tasks such as image classification, they are also known to be vulnerable to adversarial examples [12, 43]. Adversarial examples are malicious images obtained by adding imperceptible perturbations to benign images. Notably, the transferability of adversarial examples is an intriguing phenomenon, which refers to the property that the same adversarial example can successfully attack different black-box DNNs [5, 30, 34, 51].

It has been observed that image style can be decoupled from image content, and style transfer techniques allow us to generate stylized images based on arbitrary style images [18]. Image style refers to the unique visual characteristics of an image, including its colors, textures, and lighting. For instance, two photos of the same object taken by different photographers can have very different styles. Robust DNNs should rely more on content features of data than style features. This inspired us to improve attack transferability from the perspective of avoiding non-robust features. We believe that style features of DNNs are non-robust for building transferable attacks. However, existing attacks do not distinguish between the surrogate models' style and content features, which may reduce attack transferability.

We propose using stylized surrogate models to control style features, which can significantly improve transferability. We refer to the original surrogate model as the "vanilla model." The proposed stylized model is created by adding an adaptive instance normalization (IN) layer to the vanilla

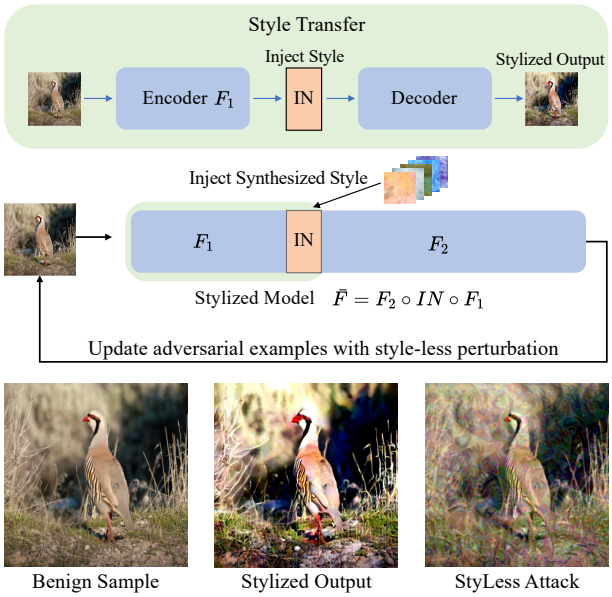


Figure 1. An overview of our StyLess attack. We create stylized model $\bar{F}$ by injecting synthesized style features into the surrogate model ($F = F_2 \circ F_1$) using an adaptive IN layer. StyLess reduces the use of non-robust style features in the vanilla surrogate model F, ultimately improving attack transferability.

*Corresponding author.
[1] Our code is available at https://github.com/uhiu/StyLess

model. By adjusting the parameters of the inserted IN layer, we can easily transform the style features of the surrogate models. To compare the stylized and vanilla surrogate models, we analyzed their network losses during optimization. Surprisingly, we found that the adversarial loss of the vanilla model increased much faster than those of the stylized models, resulting in a widening loss gap. This phenomenon reveals that as the attack iteration progresses, current attack methods only focus on maximizing the loss of the vanilla model, leading to increased use of the style features of the vanilla surrogate model. However, we believe that style features are non-robust for transferable attacks, and relying too much on them may reduce attack transferability. To enhance transferability, we aim to limit the use of non-robust style features and close the loss gap.

Based on the above findings, we propose a novel method called StyLess to improve the transferability of adversarial examples. Our method uses multiple synthesized style features to compete with the original style features during the iterative optimization of attack. The process is illustrated in Figure 1. We encode various synthesized style features into a surrogate model via an IN layer to achieve stylized surrogate models. Instead of using only the vanilla surrogate model, we use the gradients of both the stylized surrogate models and the vanilla one to update adversarial examples. The front part of the surrogate model works as a style encoder, and the IN layer simulates synthesized style features. Although we can use a decoder to explicitly generate the final stylized samples, it is unnecessary for the proposed attack method. Experimental results demonstrate that StyLess can enhance the transferability of state-of-the-art adversarial attacks on both unsecured and secured black-box DNNs. Our main contributions are summarized as follows:

- We introduce a novel perspective for interpreting attack transferability: the original style features may hinder transferability. We verify that current iterative attacks increasingly use the style features of the surrogate model during the optimization process.
- We propose a novel attack called StyLess to enhance transferability by minimizing the use of original style features. To achieve this, we insert an IN layer to create stylized surrogate models and use gradients from both stylized and vanilla models.
- We conducted comprehensive experiments on various black-box DNNs to demonstrate that StyLess can significantly improve attack transferability. Furthermore, we show that StyLess is a generic approach that can be combined with existing attack techniques.

2. Related Work

Adversarial Attacks. Adversarial attacks reveal the vulnerability of current DNNs [43]. The classic adversarial attack methods are gradient-based, such as FGSM [12] and I-FGSM [25]. C&W [4] considers optimizing the distance between adversarial examples and benign samples, and proposed optimization-based attacks. Adversarial attacks can also be performed in the physical world [10, 39]. As for defending against adversarial examples, adversarial training is a popular defense method that uses adversarial examples as extra training data to improve robustness [32].

Increasing Attack Transferability. An intriguing property of adversarial attacks is the transferability. Ensemble-based attack [30] uses multiple surrogate networks instead of one network. Ghost networks [27] generates different surrogate networks by perturbing skip connection and dropout layers. Optimization methods, such as MI [5], uses a momentum-based optimization, while VT [47] introduces gradient variance to control the stability of the localized gradients. RAP [35] generates adversarial examples located in a flat loss region. Data augmentation methods, such as DI [51], uses image transformation like resizing and padding, while TI [6] considers translating image pixels. SI [29] calculates gradients with the help of several scaled benign samples. Admix [48] calculates iterative gradients by mixing the benign images with randomly sampled images.

Various network architectures and features exhibit different relationships with adversarial attacks. DNNs' linearity is believed to cause adversarial vulnerability [12], and LinBP [13] skips the nonlinear activation during the back-propagation. SGM [49] uses more gradients through skip connections in residual networks. To better leverage the intermediate layers, one can train auxiliary classifiers based on feature spaces [20, 21], maximize the distance between natural images and their adversarial examples in feature spaces [54], or fine-tune the existing adversarial examples in intermediate layer level by ILA [17, 26].

Style Transfer and Instance Normalization. Style transfer can change the style of an image to match the style of another one [9, 11, 23]. Fast feedforward networks can perform stylization with arbitrary styles in a single forward pass [18, 28]. Interestingly, style transfer has a wide range of applications. AdvCam [8] uses natural styles to hide non-L_p restricted perturbations. FSA [52] generates natural-looking adversarial examples by using optimized style changes. Style transfer has also been used to improve network robustness by exploring additional feature information [33]. Latent style transformations can detect adversarial attacks [46]. AMT-GAN [15] proposes an adversarial makeup transfer to protect facial privacy by preserving stronger black-box transferability.

The family of instance normalization (IN) including batch normalization [22], layer normalization [1], instance normalization [45], and group normalization [50]. Normalizations are mainly used to reduce the covariate shift, and speed model training. Recently, normalizations have been found to be related to robustness. It has been shown that

batch normalization makes DNNs use more non-robust but useful features [3, 19]. AdvBN proposed adding an extra batch normalization into network training to increase training loss adversarially, which enables the network to resist various domain shifts [40]. Adjusting batch normalization statistics such as the running mean and variance in the inference phase, which are estimated during training, improves robustness and defense common corruption [2, 38].

Among existing style-based attacks, FSA [52] differentiates style features and content features, which is similar to our method. However, there are three significant differences between FSA and our approach: 1) FSA proposes to hide adversarial perturbations in the optimized style, while we avoid relying on any style. 2) FSA aims at enhancing the natural looking of non-ℓ_p restricted attacks, while we focus on the transferability of ℓ_p restricted adversarial examples. 3) Both FSA and our work are inspired by AdaIN [18], but we use IN layer differently. FSA perturbs the IN layer to search malicious styles and requires a decoder. But we use randomized IN layers to augment attacks and don't need to train a decoder.

3. Methodology

3.1. Threat Model

Attack objective. Given a benign image x with label y, transfer-based attacks aim to generate an adversarial perturbation based on a white-box surrogate network F. The general attack objective can be formulated as follows:

$$\max_{\delta} \mathcal{L}(F(x+\delta), y) \quad \text{s.t.} \quad \|\delta\| \leq \epsilon, \tag{1}$$

where $\mathcal{L}$ denotes the adversarial loss, δ is the adversarial perturbation, and ϵ is the maximum perturbation size.

A popular framework to solve the above problem is iterative fast gradient sign method (I-FGSM) [12, 25]:

$$x_{adv}^{t+1} = x_{adv}^{t} + \alpha \cdot \text{sign}\left(\nabla_x \mathcal{L}\left(F(x_{adv}^{t}), y\right)\right), \tag{2}$$

where α is the learning rate, and a clip function will be used on x_{adv}^{t+1} to ensure $\|x_{adv}^{t+1} - x\| \leq \epsilon$.

Attacker capability. We follow the same setting in previous work that attackers have a surrogate model and some test samples, but cannot access target models, and don't know network architectures, training data, or defense strategies. It should be noted that our method doesn't require any additional datasets. Our approach involves style features, which can be extracted from an arbitrary image or synthesized without any style image.

Transferable attacks as black-box attacks. Transferable attacks use the surrogate model F to create adversarial examples that can fool unseen target models. In this way, these attacks can be viewed as black-box attacks.

3.2. Motivation

Existing transferable attacks often rely on the gradient of the adversarial loss function $\mathcal{L}$ (Equation 2) without considering the impact of different components of $\mathcal{L}$. However, these approaches have limitations because transferable attacks should minimize the use of non-robust features of the surrogate model. Interestingly, for image classification task, style features of images are typically less robust than content features. Based on this observation, we propose to enhance attack transferability by explicitly reducing the use of style features of the surrogate model within the loss $\mathcal{L}$.

Our key idea is simulating various surrogate models without the style features of the given vanilla surrogate model. We discovered that inserting an IN layer into the vanilla surrogate model enables us to create new surrogate models that we refer to as *stylized surrogate models*. Sometimes we omit the word "surrogate." Stylized models can explicitly manipulate style features without compromising model accuracy. Recall that our goal is to construct adversarial examples that can mislead unseen target models, which should include these stylized models. However, existing methods, such as MI and I, only focus on maximizing the loss of the vanilla model.

Figure 2 indicates that MI and I(-FGSM) have limited attack transferability on stylized models since the vanilla model's adversarial loss increases much faster than stylized models', resulting in a widening loss gap. In the following sections, we will demonstrate how our method addresses this issue by maximizing the loss of both the stylized and vanilla models, which significantly improves transferability.

3.3. Stylized Surrogate Models

This section will give the definition of our stylized surrogate models, which encode various style features by inserting an IN layer. Then we will analyze the stylized loss gap ($\Delta\mathcal{L}$) between the vanilla and stylized surrogate models. Specifically, we will illustrate the increasing $\Delta\mathcal{L}$ limits transferability and our idea to decrease $\Delta\mathcal{L}$.

3.3.1 Encoding Styles by Stylized Models

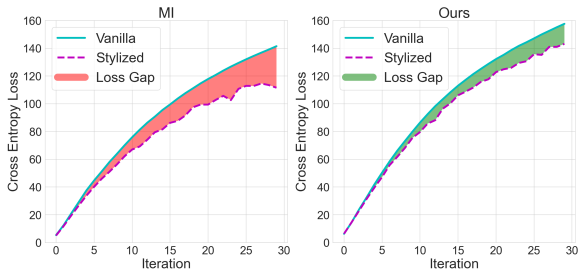
Given a classifier $F = F_2 \circ F_1$ as the surrogate model, we define a stylized surrogate model as

$$\bar{F}_{x_s} = F_2 \circ \text{IN}_{x_s} \circ F_1, \tag{3}$$

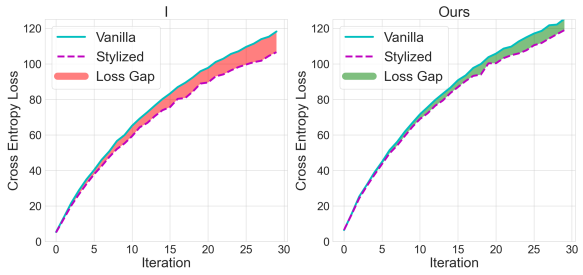
where x_s is a style input, IN_{x_s} is an IN layer instantiated by x_s. In general, an IN layer is defined as

$$\text{IN}(x; \mu, \sigma) = \sigma \cdot \left(\frac{x - \mu(x)}{\sigma(x)}\right) + \mu, \tag{4}$$

where μ and σ are the network parameters of layer IN, and $\mu(x)$ and $\sigma(x)$ are the mean and variance of input x. According to adaptive instance normalization (AdaIN) [18], to



(a) Baseline attack MI vs ours MI+StyLess



(b) Baseline attack I(-FGSM) vs ours I+StyLess

Figure 2. Illustration of the loss gap between vanilla network and stylized network, with RN50 as surrogate model. The greater the loss, the better the attack performance. The widening loss gap in the baseline means that attack performance on stylized models is lagging behind. The proposed StyLess can close the gap.

stylize an input x with a given style input x_s, we only need to instantiate the IN as

$$\mathrm{IN}_{x_s}(x) = \mathrm{IN}(x)\,\big|_{\mu=\mu(x_s),\sigma=\sigma(x_s)} \tag{5}$$

Our stylized model $\bar{F}_{x_s}$ has encoded style features. Based on AdaIN, the F_1 from the classifier $F = F_2 \circ F_1$ works as an encoder for style transfer, and given a style input x_s, we can get a stylized image as $\bar{x} = D \circ \mathrm{IN}_{x_s} \circ F_1(x)$, where D denotes a decoder. Thus, $\mathrm{IN}_{x_s} \circ F_1(x)$ has encoded the style features of x_s to $\bar{F}_{x_s}$.

3.3.2 Stylized Loss Gap Limits transferability

To validate the performance of adversarial attacks on these stylized surrogate models, we define a *stylized loss gap* as

$$\Delta\mathcal{L} = \mathbb{E}_{x_s\in\mathcal{D}}[\mathcal{L}(F(x), y) - \mathcal{L}(\bar{F}_{x_s}(x), y)], \tag{6}$$

where F and $\bar{F}_{x_s}$ are vanilla and stylized model, respectively; x_s is style input.

An increasing loss gap $\Delta\mathcal{L}$ can be observed in Figure 2 which limits attack transferability, as explained below. Hypothetically, let's say we can decouple style-dependent loss from content-dependent loss in $\mathcal{L}$ as follows:

$$\begin{aligned} \mathcal{L} &= \mathcal{L}^c + \mathcal{L}^s_x, \\ \bar{\mathcal{L}} &= \mathcal{L}^c + \mathcal{L}^s_{x_s}, \end{aligned} \tag{7}$$

where $\mathcal{L}$ is the vanilla loss, and $\bar{\mathcal{L}}$ is the stylized one; $\mathcal{L}^c$ is the common content-dependent loss; $\mathcal{L}^s_x$ is input sample x-specific style-dependent loss, while $\mathcal{L}^s_{x_s}$ is style sample x_s-specific style-dependent loss. In this case, $\Delta\mathcal{L} = \mathcal{L}^s_x - \mathcal{L}^s_{x_s}$.

In other words, the non-robust features are related to $\mathcal{L}^s_x$ and $\mathcal{L}^s_{x_s}$, while $\mathcal{L}_c$ is supposed to be shared by other unseen DNNs. MI and I-(FGSM) have increasing gaps in Figure 2 means that attackers gradually focus on optimizing the $\mathcal{L}^s_x$ part, which only belonged to the vanilla surrogate model F. Therefore, the loss gap limits the transferability of adversarial examples in unseen stylized models.

To decrease $\Delta\mathcal{L}$ and boost transferability, we consider involving $\mathcal{L}^s_{x_s}$ as a competitor of $\mathcal{L}^s_x$ in optimization process to suppress the growth of $\mathcal{L}^s_x$. In general, we can assume that all these losses in Equation 7 are non-negative, and they satisfy: $\mathcal{L}^c \gg \mathcal{L}^s_x$, $\mathcal{L}^c \gg \mathcal{L}^s_{x_s}$. Also, there is a upper bound B for these losses: $\mathcal{L} < B$, $\bar{\mathcal{L}} < B$, as the adversarial perturbations are required to smaller than a given ϵ. If we only maximize the vanilla loss $\mathcal{L}$, both $\mathcal{L}^c$ and $\mathcal{L}^s_x$ are likely to be increased. To this end, we propose to maximize $\mathbb{E}_{x_s\in\mathcal{D}}\mathcal{L}^s_{x_s} + \mathcal{L}^s_x + \mathcal{L}^c$, which involves multiple style-dependent losses $\mathcal{L}^s_{x_s}(x_s \in \mathcal{D})$ to compete with $\mathcal{L}^s_x$ and leads to a decrease of $\Delta\mathcal{L}$.

3.4. Proposed Style-Less Perturbations (StyLess)

Based on the above analysis, we propose style-less perturbations (StyLess) method to increase attack transferability by optimizing stylized loss and vanilla loss together:

$$\max_{\delta} \mathbb{E}_{x_s\in\mathcal{D}}\mathcal{L}(\bar{F}_{x_s}(x+\delta), y) + \mathcal{L}(F(x+\delta), y). \tag{8}$$

The key to generate multiple stylized models $\bar{F}_{x_s}$ is synthesizing style statistics μ, σ for Equation 5 to obtain parameterized IN layers. We propose using scaling and interpolation to simulate multiple style features, formulated as

$$\begin{aligned} \mu &= \beta(\lambda\mu_x + (1-\lambda)\mu_s), \\ \sigma &= \gamma(\lambda\sigma_x + (1-\lambda)\sigma_s), \end{aligned} \tag{9}$$

where μ_x, σ_x is the mean and variance of $F_1(x)$, relating to the benign content input x, while μ_s, σ_s are for style input x_s similarly. x_s is an arbitrary image. λ is a scalar that controls the interpolation of two styles. β and γ are c-dimensional vectors that scale the synthesized style, where c refers to the number of channels in the style feature.

We summarize the proposed StyLess attack in Algorithm 1. To obtain the parameters of the IN layer without training, we select a random image as x_s, and generate a pair of μ and σ using Equation 9. With μ and σ, we can create a stylized model $\bar{F}_{x_s}(x)$ using Equation 3 and 5. $\bar{F}_{x_s}(x)$ need to be equal to $F(x)$ or label y, otherwise we will regenerate μ and σ by altering β, γ and λ. In each iteration, we use N stylized models to update adversarial examples based on the proposed objective function in Equation 8.

Algorithm 1 Style-Less Perturbations (StyLess) Algorithm

Input: Surrogate model F, benign example x, iteration number T, maximum perturbation ϵ, data augmentation $\phi(\cdot)$, decay factor η. Scale factors β, γ and interpolation factor λ. The number of stylized models N.

Output: An adversarial example x_{adv}

$x_{adv} = x, g_0 = 0, \alpha = \epsilon/2$.
for $t = 0 \rightarrow T - 1$ **do**:
  Augment input $x_{adv} = \phi(x_{adv})$
  Obtain gradient $\tilde{g}_{t+1}$ with respect to x_{adv} using F
  repeat
    Synthesize a style statistic by Equation 9
    Obtain a stylized model $\bar{F}_{x_s}$ by Equation 3
    Get gradient $\tilde{g}$ with respect to x_{adv} using $\bar{F}_{x_s}$
    Update $\tilde{g}_{t+1} = \tilde{g}_{t+1} + \tilde{g}$
  until obtain N stylized models
  Calculate momentum $g_{t+1} = \eta \cdot g_t + \tilde{g}_{t+1}/\|\tilde{g}_{t+1}\|_1$
  Update example $x_{adv} = x_{adv} + \alpha \cdot \text{sign}(g_{t+1})$
end for
return x_{adv}.

4. Experiments

4.1. Experimental Setup

Dataset. We use ImageNet [36] for experiments. Specifically, we use 1000 images which are randomly selected from ImageNet validation set by Wang et al. [47]. Similar dataset settings have been widely used in previous work [5, 13, 29, 47, 49, 51]. These images include all categories; almost all are correctly classified by the target DNNs.

Models. We evaluate the generated adversarial examples on different black-box DNNs, including both unsecured and secured models. Unsecured models are trained on ImageNet using traditional methods, while secured models are based on adversarial training. The unsecured models include VGG19 [41], AlexNet [24], ResNet50 (RN50) [14], WideResNet101 (WRN101) [53], DenseNet121 (DN121) [16], InceptionV3 (IncV3) [42], MnasNet [44], MobileNetV2 (MNv2) [37], ShuffleNetV2 (SNv2) [31] and ViT [7]. Their pre-training parameters are obtained from PyTorch official. The secured DNNs are $\text{IncV3}_{\text{ens3}}$ (ensemble of 3 InceptionV3 networks), $\text{IncV3}_{\text{ens4}}$ (ensemble of 4 InceptionV3 networks) and $\text{IncResV2}_{\text{ens3}}$ (ensemble of 3 IncResV2 networks). These models were adversarially trained and widely used in previous work [5, 30, 47, 49]. As for the surrogate models, we use VGG19 [41], RN50 [14], WRN101 [53] and DN121 [16].

Implementation Details. We use I-FGSM [12] as the initial baseline. Unless otherwise specified, the attacks are untargeted and l_∞-restricted. The maximum perturbation size is set to $\epsilon = 16/255$. We compare StyLess primarily with six transferable attacks: MI [5], DI [51], TI [6], SI [29], Admix (AI) [48], and an ensemble-based approach [30]. We set the optimization step size to $\alpha = \epsilon/2$, and the number of iterations to $T = 50$. The momentum decay in MI is $\mu = 1$. For DI, SI and AI, we follow the official settings described in the corresponding papers. For StyLess, we simulate 10 stylized models in each iteration, denoted by $N = 10$. To generate a stylized model for a given x, we randomly sample λ from $[0, 0.2]$, and β, γ from $[0, 2]$, and ensure that $\bar{F}_{x_s}(x)$ is equal to $F(x)$ or the real label. The IN layer is inserted after the first bottleneck block for RN50 and WRN101, and after the first dense block for DN121.

4.2. Attacking Unsecured Models

We compare StyLess with other attacks on various unsecured DNNs using three surrogate models. Table 1 shows that StyLess is a powerful and generic method that can be combined with existing attack methods to further improve attack transferability. Specifically, we compare StyLess with I, MI, DI, TI, SI and Admix (AI). For the most challenging case in the table, attacking the black-box IncV3 (let's take RN50 $\Rightarrow$ IncV3 attack as an example), StyLess significantly improves the attack success rate of baseline attacks (I and MI): 46.2% $\rightarrow$ 68.3% (I), 59.2% $\rightarrow$ 78.9% (MI). StyLess also demonstrates its capabilities when using other DNNs as the surrogate network. For instance, DN121 $\Rightarrow$ SNv2 attack, StyLess significantly improves the baseline: 69.3% $\rightarrow$ 91.4% (I), 77.4% $\rightarrow$ 95.1% (MI).

StyLess can be combined with other attack techniques. Previous work has shown that combining various attack methods results in powerful and transferable attacks. StyLess can be integrated with existing combination-based attacks to enhance attack transferability. We report the results of four combinations of existing attacks: MDI, MTDI, MTDSI, and MTDAI. StyLess further enhances these four attack methods' attack success rate by +8.0%, +8.2%, +3.0%, and +3.3% (in the case of RN50 $\Rightarrow$ IncV3 attack). We evaluate attack performance using various surrogate models, including RN50, WRN101 and DN121. For instance, when combining MTDAI with StyLess, our method enhances the transferability of MTDAI: WRN101 $\Rightarrow$ IncV3 attack: +4.7%, WRN101$\Rightarrow$ MNv2 attack: +1.5%, WRN101$\Rightarrow$ SNv2 attack: +3.4%, DN121 $\Rightarrow$ IncV3 attack: +4.3%, DN121 $\Rightarrow$ MNv2 attack: +0.6%, DN121 $\Rightarrow$ SNv2 attack: +3.8%. The results show that StyLess is an efficient and generic approach for improving attack transferability.

4.3. Attacking Secured Models

We evaluate StyLess on three widely-used secured models: $\text{IncV3}_{\text{ens3}}$, $\text{IncV3}_{\text{ens4}}$ and $\text{IncResV2}_{\text{ens}}$, as shown in Table 2. We present the results of I, MI, MDI, MTDI, and MTDSI on the three secured models using two surrogate models: RN50 and WRN101. These secured networks are more robust than the unsecured DNNs we men-

Table 1. Attacking unsecured black-box models with StyLess.

Source	Attack	VGG19	RN50	WRN101	DN121	IncV3	MNv2	SNv2
RN50	I / +Ours	72.8 / **88.8**	100 / 100	80.8 / **97.1**	83.0 / **97.8**	46.2 / **68.3**	77.1 / **91.9**	60.6 / **77.8**
	MI / +Ours	82.9 / **94.1**	100 / 100	83.9 / **97.2**	87.5 / **98.7**	59.2 / **78.9**	83.8 / **93.2**	72.3 / **83.5**
	MDI / +Ours	97.5 / **99.2**	100 / 100	98.2 / **99.8**	99.4 / **100**	89.5 / **97.5**	98.1 / **99.9**	88.5 / **96.9**
	MTDI / +Ours	98.6 / **99.7**	100 / 100	99.2 / **100**	99.8 / **100**	90.2 / **98.4**	98.7 / **100**	90.8 / **98.1**
	MTDSI / +Ours	98.6 / **99.2**	100 / 100	99.5 / **100**	99.8 / **100**	96.2 / **99.2**	98.7 / **99.9**	96.2 / **98.2**
	MTDAI / +Ours	99.3 / **99.6**	100 / 100	99.8 / **100**	99.9 / **100**	95.5 / **98.8**	99.7 / **100**	95.7 / **99.0**
WRN101	I / +Ours	64.8 / **79.6**	88.6 / **98.7**	100 / 100	77.7 / **94.2**	43.2 / **66.6**	68.1 / **84.7**	58.1 / **73.2**
	MI / +Ours	75.6 / **86.4**	89.8 / **98.7**	100 / 100	83.9 / **96.4**	57.6 / **73.7**	76.1 / **87.0**	70.2 / **81.2**
	MDI / +Ours	92.2 / **98.5**	99.0 / **99.9**	100 / 100	97.1 / **99.9**	86.3 / **96.3**	93.6 / **99.0**	86.8 / **95.8**
	MTDI / +Ours	92.5 / **98.6**	99.2 / **100**	100 / 100	97.9 / **99.6**	89.2 / **97.9**	95.4 / **99.3**	88.4 / **97.0**
	MTDSI / +Ours	95.9 / **98.9**	99.6 / **100**	100 / 100	99.7 / **100**	95.8 / **99.5**	97.7 / **99.9**	94.5 / **98.4**
	MTDAI / +Ours	96.5 / **99.2**	99.9 / **99.9**	100 / 100	99.4 / **100**	93.7 / **98.4**	98.1 / **99.6**	94.6 / **98.0**
DN121	I / +Ours	79.7 / **97.7**	87.0 / **99.4**	74.7 / **98.0**	100 / 100	55.7 / **88.7**	81.3 / **97.0**	69.3 / **91.4**
	MI / +Ours	86.9 / **99.1**	89.4 / **99.7**	78.2 / **98.6**	100 / 100	66.0 / **95.7**	86.5 / **98.7**	77.4 / **95.1**
	MDI / +Ours	96.7 / **99.9**	98.9 / **100**	95.4 / **99.8**	100 / 100	90.5 / **99.0**	97.5 / **100**	90.9 / **98.6**
	MTDI / +Ours	97.5 / **99.9**	99.1 / **100**	95.9 / **99.8**	100 / 100	92.7 / **99.4**	96.9 / **100**	92.2 / **98.6**
	MTDSI / +Ours	97.4 / **99.9**	99.3 / **100**	98.0 / **99.8**	100 / 100	96.5 / **99.8**	98.9 / **100**	95.1 / **99.1**
	MTDAI / +Ours	99.2 / **99.9**	99.8 / **100**	98.1 / **99.9**	100 / 100	95.1 / **99.4**	99.4 / **100**	95.2 / **99.0**

Table 2. Attacking three secured black-box models with StyLess. The surrogate model is RN50 or WRN101.

Attack	$\underline{\textbf{RN50}} \Rightarrow$		
	IncV3_{ens3}	IncV3_{ens4}	IncResV2_{ens}
I / +Ours	21.6 / **34.6**	18.9 / **32.0**	14.1 / **21.5**
MI/ +Ours	31.5 / **47.1**	29.3 / **42.4**	20.8 / **31.0**
MDI / +Ours	59.6 / **78.1**	53.5 / **69.8**	38.3 / **57.3**
MTDI / +Ours	69.2 / **89.6**	63.8 / **81.8**	54.6 / **72.2**
MTDSI / +Ours	88.0 / **93.1**	84.7 / **91.3**	77.8 / **84.5**
	$\underline{\textbf{WRN101}} \Rightarrow$		
I / +Ours	23.4 / **40.3**	20.9 / **34.6**	15.5 / **25.6**
MI/ +Ours	35.0 / **51.2**	30.4 / **46.6**	24.7 / **35.9**
MDI / +Ours	64.0 / **81.5**	58.8 / **76.2**	46.9 / **66.4**
MTDI / +Ours	75.5 / **91.4**	70.5 / **87.4**	63.6 / **81.0**
MTDSI / +Ours	91.6 / **97.5**	88.6 / **96.1**	83.0 / **92.5**

tioned above. Taking RN50 $\Rightarrow$ IncV3_{ens3} attack as an example, I, MI, MDI, and MTDI achieve attack success rates of only 21.6%, 31.5%, 59.6%, and 69.2%, respectively. StyLess effectively improves the performance of these attacks. Notably, MTDSI has been improved by our method: 91.6% $\rightarrow$ 97.5% in WRN101$\Rightarrow$ IncV3_{ens3} attack; 88.6% $\rightarrow$ 96.1% in WRN101$\Rightarrow$ IncV3_{ens4} attack; 83.0% $\rightarrow$ 92.5% in WRN101$\Rightarrow$ IncResV2_{ens} attack. StyLess demonstrates its great power to enhance transferability in breaking these challenging secured DNNs.

Figure 3 shows a comparison of StyLess and LinBP. We consider LinBP and its combination with existing attacks such as ILA, SGM, and MDI on various target DNNs. Our method exhibits the best attack transferability.

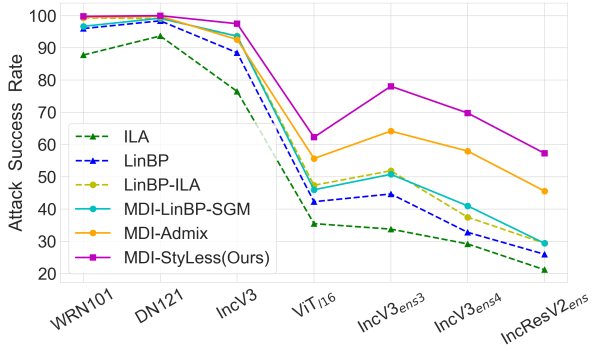


Figure 3. Comparing with LinBP on different black-box models and defenses. The surrogate model is RN50.

4.4. Combining with Ensemble-Based Method

Generating adversarial examples based on multiple surrogate networks simultaneously can improve attack performance in practice [30]. This ensemble-based method has been shown to be a powerful attack, and StyLess can further improve it. We use an ensemble of RN50, WRN101, and DN121 as the integrated surrogate model.

We compared our StyLess with ensemble-based MI and MTDI in Table 3. The considerable strength of the ensemble-based method can be seen when comparing with the results in Table 1 and 2. For example, in RN50 $\Rightarrow$ IncV3 attack, the baseline attack MI only get 59.2%, while it get 81.0% attack success rate in the case of RN50+WRN101+DN121 $\Rightarrow$ IncV3 attack. Noted that we generally use $\epsilon = 16$, which is a standard setting. As we can see, when $\epsilon = 16$, the ensemble-based MTDI achieves an average attack success rate of less than 90% on the three

Table 3. Combining with ensemble-based attack method.

ϵ	Attack	RN50+WRN101+DN121 ⇒ VGG19	RN50	WRN101	DN121	IncV3	MNv2	SNv2	$IncV3_{ens3}$	$IncV3_{ens4}$	$IncResV2_{ens}$
4	MI	56.8	96.4	94.8	97.2	35.7	62.3	48.3	18.4	15.8	9.4
	MTDI	79.2	97.9	96.6	98.8	67.5	84.5	71.6	36.5	32.1	22.3
	+StyLess	**90.4**	**99.5**	**99.4**	**99.9**	**83.2**	**94.3**	**85.4**	**50.0**	**42.9**	**29.1**
8	MI	79.2	99.6	98.7	99.4	58.3	83.3	68.0	31.4	26.8	18.2
	MTDI	95.2	99.9	99.5	99.9	89.5	97.5	90.4	67.6	61.8	46.9
	+StyLess	**99.1**	**100**	**100**	**100**	**97.5**	**99.6**	**97.2**	**85.1**	**77.4**	**66.1**
16	MI	92.3	**100**	**100**	**100**	81.0	93.2	85.0	53.0	46.3	35.0
	MTDI	99.2	**100**	**100**	**100**	97.5	**100**	97.9	92.7	86.4	80.7
	+StyLess	**100**	**100**	**100**	**100**	**99.8**	**100**	**100**	**98.5**	**97.5**	**94.3**

secured networks: $IncV3_{ens3}$, $IncV3_{ens4}$, and $IncResV2_{ens}$. StyLess can further boost the transferability of ensemble-based MTDI: 92.7% → 98.5% for $IncV3_{ens3}$, 86.4% → 97.5% for $IncV3_{ens4}$, 80.7% → 94.3% for $IncResV2_{ens}$. These results show that StyLess is a different type of attack, and can work perfectly with the ensemble-based method.

We also report the experimental results with different ϵ. We use $\epsilon = 4$ or 8 to increase the difficulty to attack. For instance, in the case of ensemble-based MTDI⇒ $IncV3_{ens3}$ attack, attack success rate drops from 92.7% to 67.6% when $\epsilon = 8$ instead of $\epsilon = 16$, and StyLess can help ensemble-based MTDI gains +17.5% (67.6% → 85.1%), which is a huge improvement. When $\epsilon = 4$, the results also demonstrate the advantages of StyLess. This shows that StyLess consistently delivers strong transferability when faced with a more robust network that is difficult to attack.

4.5. Attacking the Google Cloud Vision API

We use the Google Cloud Vision API as an example of real-world applications to evaluate the transferability of attacks. This API allows us to use various vision features, such as image labeling and face detection. As a black-box model, regular users like us cannot access its network architecture, training data, or defense mechanism. In this section, we will focus on attacking its image labeling feature.

To utilize the image labeling feature of the API, we need to upload images and obtain the predicted labels. We use 1000 images from ImageNet as the target images, as described in the experimental setup. Due to the fact that the API does not support all 1000 categories in ImageNet, our objective is to mislead the API's original top-1 prediction.

We use ResNet50 as the surrogate network to compare the baseline method MTDSI with our method. Experimental results show that MTDSI achieves an attack success rate of 75.6% on the Google Cloud Vision API. Our approach, MTDSI-SyLess, achieves an attack success rate of 85.2%, which is a 9.6% improvement over the baseline. Our results demonstrate that transfer-based black-box attacks pose a severe threat to real-world applications, and StyLess can effectively boost attack transferability.

4.6. Ablation Study

In this session, we will present four ablation studies: 1) The position of the inserted IN layer; 2) The number of the generated stylized models; 3) The clean losses of stylized models and attack transferability; 4) The most important statistic of style features.

Which position to insert the IN layer? As shown in Figure 4, we use features from different layers of the vanilla surrogate networks as the encoder F_1 for style transfer. We evaluate the attack success rates using two surrogate networks: WRN101 and DN121. After injecting synthesized styles into different network layers, we report the attack success rates on various DNNs, including AlexNet, VGG19, RN50, WRN101, and DN121. We observe a trend that the best attack success rates are usually achieved in the shallow layers of the surrogate networks. For example, when using WRN101 as the surrogate network, injecting the synthesized styles in the layers before layer ten is a good choice. Using intermediate layers such as layers 40 to 80 is also acceptable, but the attack performance may be unstable or even worse when injecting the styles in the last few layers. The results of using DN121 as the surrogate network also indicate that the last few layers are the worst choices, and the shallow layers are the optimal.

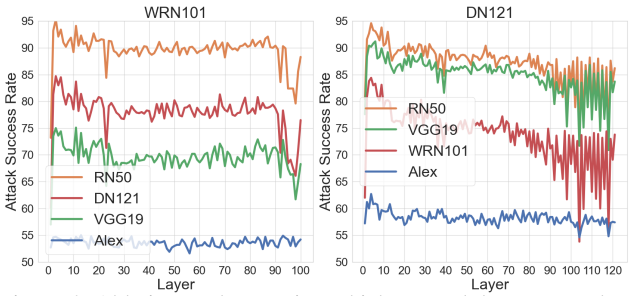


Figure 4. Ablation study on using which network layer to synthesize styles. The surrogate networks are WRN101 and DN121.

How many stylized models should be created in each iteration? In Figure 5, we vary the number of stylized models generated in each attack iteration from zero to

ten. We conduct experiments by combining StyLess with two baseline attacks: MI and DI. The first figure evaluates the attack success rates on three unsecured networks for our MI+StyLess attack. In the second figure, we test DI+StyLess on three secured models. When the number of stylized models is 0, StyLess is not involved, so it is the vanilla baseline attack. As we can see, when the number increases from 0 to 1, the attack success rate starts to grow, which means StyLess starts to work. There is an anomaly when DI begins to combine with StyLess. If the number is less than three, the attack success rate on IncResV2$_{ens}$ is slightly worse than the baseline (around 25%). When the number increases by more than three, the attack success rate becomes higher than the baseline. This may be because IncResV2$_{ens}$ is very strong, and more synthesized style features need to be injected into the surrogate model. According to the figure, StyLess works quite well when six to ten stylized models are used in an attack iteration.

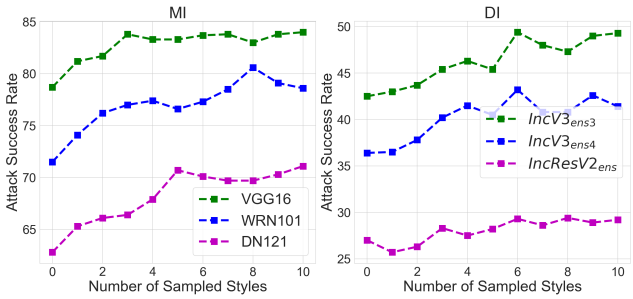


Figure 5. Ablation study on the number of stylized models in an attack iteration. The surrogate network is RN50.

How network loss affects attack success rate? In Figure 6, we demonstrate how varying strengths of style injection can affect network loss, which in turn impacts attack performance on a surrogate network (in this case, VGG19). The strength of synthesized style features is denoted by the number of stars, with more stars indicating greater style strength that alters the style features of the surrogate model. Generally, injecting synthesized style features should not significantly affect the clean loss, as an increase in network loss typically leads to a decrease in clean accuracy. Overly corrupted stylized surrogate models can also result in bad gradients for attack methods. Therefore, there is an upper bound on clean loss when generating stylized models. The red line in the figure represents the situation of overly corrupting, in which the clean loss exceeds the estimated bound (indicated by a yellow line in the right figure), and the attack success rate drops significantly. This demonstrates the importance of maintaining relatively good clean accuracy when creating stylized networks.

Which statistic of style features matters most? In Figure 7, we compare the effects of the mean and variance of style features on StyLess. Here the interpolation factor is $\lambda = 0$. Equation 9 shows that β involves the mean in IN, while γ affects the variance. The attack success rate represents the average success rates of attacks on five DNNs: VGG19, AlexNet, RN50, WRN101, and DN121. The results show that γ plays a more important role than β. Specifically, when RN50 is used as the surrogate network, modifying β alone barely improves the baseline attack, while involving γ alone enhances the attack success rate by around 5%. A similar observation can be made for VGG19. From the perspective of gradient calculations, this also makes sense. When we backpropagate through an IN layer, we have $\frac{\partial}{\partial x}\text{IN} = \sigma$, which also indicates that the variance matters most for adversarial attacks.

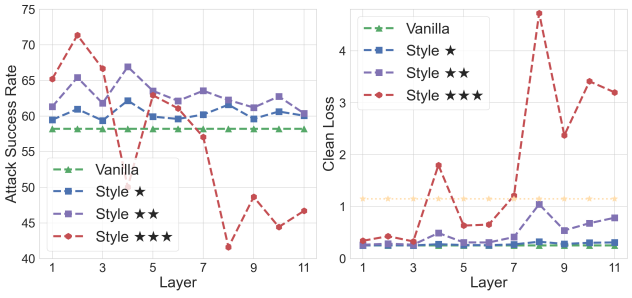


Figure 6. Study how synthesized style features affect the clean loss, which in return impacts the attack success rate. The number of stars indicates the degree of change in original style features.

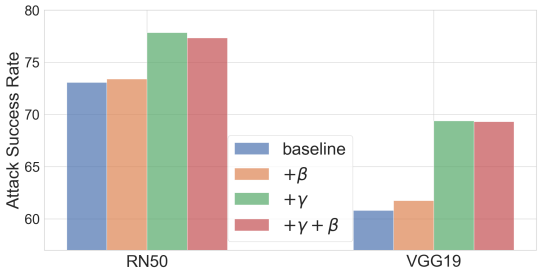


Figure 7. The effect of different statistics of style.

5. Conclusion

In this work, we analyze the mechanism of attack transferability in terms of style features. We demonstrate that existing attack methods increasingly use the style features of surrogate models during the iterative optimization, which hampers attack transferability. To address this issue, we propose a novel attack method called StyLess to enhance transferability by reducing reliance on original style features. StyLess uses stylized surrogate models instead of a vanilla surrogate model. Experimental results show that StyLess outperforms existing attacks by a large margin, and can be combined with other attack methods. Notably, StyLess is a different paradigm from previous transferable attack methods, and we hope it will shed light on the interpretation of adversarial attacks in the future.

References

[1] Lei Jimmy Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization. *CoRR*, abs/1607.06450, 2016. 2

[2] Philipp Benz, Chaoning Zhang, Adil Karjauv, and In So Kweon. Revisiting batch normalization for improving corruption robustness. In *WACV*, 2021. 3

[3] Philipp Benz, Chaoning Zhang, and In So Kweon. Batch normalization increases adversarial vulnerability and decreases adversarial transferability: A non-robust feature perspective. In *ICCV*, 2021. 3

[4] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017. 2

[5] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *CVPR*, 2018. 1, 2, 5

[6] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *CVPR*, 2019. 2, 5

[7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. *ICLR*, 2021. 5

[8] Ranjie Duan, Xingjun Ma, Yisen Wang, James Bailey, A. K. Qin, and Yun Yang. Adversarial camouflage: Hiding physical-world attacks with natural styles. In *CVPR*, 2020. 2

[9] Vincent Dumoulin, Jonathon Shlens, and Manjunath Kudlur. A learned representation for artistic style. In *ICLR*, 2017. 2

[10] 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 *CVPR*, 2018. 2

[11] Leon A Gatys, Alexander S Ecker, and Matthias Bethge. Image style transfer using convolutional neural networks. In *CVPR*, 2016. 2

[12] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 1, 2, 3, 5

[13] Yiwen Guo, Qizhang Li, and Hao Chen. Backpropagating linearly improves transferability of adversarial examples. In *NeurIPS*, 2020. 2, 5

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 5

[15] Shengshan Hu, Xiaogeng Liu, Yechao Zhang, Minghui Li, Leo Yu Zhang, Hai Jin, and Libing Wu. Protecting facial privacy: Generating adversarial identity masks via style-robust makeup transfer. In *CVPR*, 2022. 2

[16] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 5

[17] Qian Huang, Isay Katsman, Zeqi Gu, Horace He, Serge J. Belongie, and Ser-Nam Lim. Enhancing adversarial example transferability with an intermediate level attack. In *ICCV*, 2019. 2

[18] Xun Huang and Serge Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *ICCV*, 2017. 1, 2, 3

[19] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features. In *NeurIPS*, 2019. 3

[20] Nathan Inkawhich, Kevin J. Liang, Lawrence Carin, and Yiran Chen. Transferable perturbations of deep feature distributions. In *ICLR*, 2020. 2

[21] Nathan Inkawhich, Kevin J. Liang, Binghui Wang, Matthew Inkawhich, Lawrence Carin, and Yiran Chen. Perturbing across the feature hierarchy to improve standard and strict blackbox attack transferability. In *NeurIPS*, 2020. 2

[22] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *ICML*, 2015. 2

[23] Justin Johnson, Alexandre Alahi, and Li Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016. 2

[24] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *NeurIPS*, 2012. 1, 5

[25] Alexey Kurakin, Ian J. Goodfellow, and Samy Bengio. Adversarial examples in the physical world. In *ICLR (Workshop)*, 2017. 2, 3

[26] Qizhang Li, Yiwen Guo, and Hao Chen. Yet another intermediate-level attack. In *ECCV*, 2020. 2

[27] Yingwei Li, Song Bai, Yuyin Zhou, Cihang Xie, Zhishuai Zhang, and Alan L. Yuille. Learning transferable adversarial examples via ghost networks. In *AAAI*, 2020. 2

[28] Yijun Li, Chen Fang, Jimei Yang, Zhaowen Wang, Xin Lu, and Ming-Hsuan Yang. Universal style transfer via feature transforms. *NeurIPS*, 2017. 2

[29] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E. Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *ICLR*, 2020. 2, 5

[30] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. In *ICLR*, 2017. 1, 2, 5, 6

[31] Ningning Ma, Xiangyu Zhang, Hai-Tao Zheng, and Jian Sun. Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *ECCV*, 2018. 5

[32] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018. 2

[33] Muzammal Naseer, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. Stylized adversarial defense. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2022. 2

[34] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *arXiv preprint arXiv:1605.07277*, 2016. 1

[35] Zeyu Qin, Yanbo Fan, Yi Liu, Li Shen, Yong Zhang, Jue Wang, and Baoyuan Wu. Boosting the transferability of adversarial attacks with reverse adversarial perturbation. In *NeurIPS*, 2022. 2

[36] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision (IJCV)*, 115:211–252, 2015. 5

[37] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 5

[38] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. In *NeurIPS*, 2020. 3

[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 Conference on Computer and Communications Security (CCS)*, pages 1528–1540, 2016. 2

[40] Manli Shu, Zuxuan Wu, Micah Goldblum, and Tom Goldstein. Encoding robustness to image style via adversarial feature perturbations. *NeurIPS*, 2021. 3

[41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 5

[42] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 5

[43] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian J. Goodfellow, and Rob Fergus. Intriguing properties of neural networks. In *ICLR*, 2014. 1, 2

[44] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V Le. Mnasnet: Platform-aware neural architecture search for mobile. In *CVPR*, 2019. 5

[45] Dmitry Ulyanov, Andrea Vedaldi, and Victor S. Lempitsky. Instance normalization: The missing ingredient for fast stylization. *CoRR*, abs/1607.08022, 2016. 2

[46] Shuo Wang, Surya Nepal, Alsharif Abuadbba, Carsten Rudolph, and Marthie Grobler. Adversarial detection by latent style transformations. *IEEE Transactions on Information Forensics and Security (TIFS)*, 17:1099–1114, 2022. 2

[47] Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. In *CVPR*, 2021. 2, 5

[48] Xiaosen Wang, Xuanran He, Jingdong Wang, and Kun He. Admix: Enhancing the transferability of adversarial attacks. In *ICCV*, 2021. 2, 5

[49] 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. 2, 5

[50] Yuxin Wu and Kaiming He. Group normalization. In *ECCV*, 2018. 2

[51] 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*, 2019. 1, 2, 5

[52] Qiuling Xu, Guanhong Tao, Siyuan Cheng, and Xiangyu Zhang. Towards feature space adversarial attack by style perturbation. In *AAAI*, 2021. 2, 3

[53] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *BMVC*, 2016. 5

[54] Wen Zhou, Xin Hou, Yongjun Chen, Mengyun Tang, Xiangqi Huang, Xiang Gan, and Yong Yang. Transferable adversarial perturbations. In *ECCV*, 2018. 2