Natural Color Fool: Towards Boosting Black-box Unrestricted Attacks

Shengming Yuan
shengming.yuan@outlook.com

Qilong Zhang
qilong.zhangdl@gmail.com

Lianli Gao*
lianli.gao@uestc.edu.cn

Yaya Cheng
yaya.cheng@hotmail.com

Jingkuan Song
jingkuan.song@gmail.com

Center for Future Media, University of Electronic Science and Technology of China

Abstract

Unrestricted color attacks, which manipulate semantically meaningful color of an image, have shown their stealthiness and success in fooling both human eyes and deep neural networks. However, current works usually sacrifice the flexibility of the uncontrolled setting to ensure the naturalness of adversarial examples. As a result, the black-box attack performance of these methods is limited. To boost transferability of adversarial examples without damaging image quality, we propose a novel Natural Color Fool (NCF) which is guided by realistic color distributions sampled from a publicly available dataset and optimized by our neighborhood search and initialization reset. By conducting extensive experiments and visualizations, we convincingly demonstrate the effectiveness of our proposed method. Notably, on average, results show that our NCF can outperform state-of-the-art approaches by **15.0%∼32.9%** for fooling normally trained models and **10.0%∼25.3%** for evading defense methods. Our code is available at https://github.com/ylhz/Natural-Color-Fool.

1 Introduction

Deep neural networks (DNNs) have achieved excellent performance on a large number of tasks, such as image recognition [1, 2], medical diagnostics [3, 4], and autonomous driving [5]. However, with the rise of the field of adversarial attack [6, 7, 8, 9, 10, 11, 12], the robustness of state-of-the-art DNNs [2, 13, 14, 15, 16] is put in question. Adversarial examples thus reveal the vulnerability of DNNs and inevitably raise concerns about widely deployed models. To avoid potential risks, it is crucial to expose as many of the model's "blind spots" as possible at this stage.

Generally, scenarios of adversarial attacks can divide into white-box and black-box. For the white-box scenario [17, 18, 19, 20], all information about the target model is publicly available, such as the architecture and parameters. Although the attacker can easily fool them with human-imperceptible perturbations in this case, it is not practical in the real world since deployed models are usually opaque to unauthorized users. To overcome this limitation, numerous works begin to study the black-box attack [7, 21, 22, 23, 24, 25], which relies on the cross-model transferability of adversarial examples. Typically, most existing black-box attacks maintain image quality by constraining the L_p-norm of perturbation [26, 23, 27, 8]. However, the L_p-norm constraint in RGB color space is not an ideal indicator for measuring the perceptual distance between two images since adversarial

*Corresponding author

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

examples with a small perturbation size may still look unnatural to human eyes [19]. Thus the unrestricted attack [28, 29, 30, 31, 32, 33, 34] is proposed — replacing the traditional small L_p-norm perturbations with uncontrolled yet human-imperceptible ones, which is more practically meaningful. Among existing unrestricted attacks, the unrestricted color branch is a very promising direction. As demonstrated in previous works [30, 32, 29], only manipulating the color of an image is not noticeable to human eyes but can easily fool DNNs. This is mainly because global uniform changes in images are usually imperceptible, and DNNs are vulnerable to certain large-scale patterns.

However, current unrestricted color attacks usually sacrifice the flexibility of the uncontrolled setting to ensure the naturalness of adversarial examples: either rely on intuition [28, 29] and objective metric [32] or use relatively small changes [30, 31]. Consequently, the transferability of adversarial examples generated by these methods is limited.

To remedy this, we propose a **Natural Color Fool (NCF)** based on a more flexible perturbation space. Specifically, we build a color distribution library that contains diverse yet realistic color distributions sampled from the ADE20K dataset [35] for each semantic class. By borrowing the color mapping technique of [36], we utilize the library to generate a set of image variants with realistic color distribution for each clean image. Then we use the substitute model to select the most adversarial variant. To boost its transferability, we further introduce **neighborhood search** and **initialization reset** to optimize, thus obtaining the resulting adversarial example. Figure 1 illustrates the simplified pipeline for our method.

In summary, our main contributions are:

- We observe that current unrestricted color attacks lack flexibility, which results in limited transferability of the adversarial examples.
- To alleviate this issue, we propose a Natural Color Fool (NCF), which fully exploits color distributions of semantic classes in an image to craft human-imperceptible, flexible, and highly transferable adversarial examples.
- Extensive experiments and visualizations demonstrate the effectiveness of our proposed method. Significantly, with high NIMA scores [37], NCF averagely outperforms state-of-the-art methods by **15.0%∼32.9%** for fooling normally trained models, and **10.0%∼25.3%** for evading defense models.

2 Related Works

2.1 Unrestricted Attacks

Unrestricted attacks [30, 38, 39, 32], which usually significantly modify the image but can yield human-imperceptible adversarial examples, have been widely studied in recent years. For example, some works modify domain-specific structural attributes of the image. SemanticAdv [38] uses an attribute-conditioned image editing model to modify the attributes of the face (e.g., modifying the degree of mouth opening). advCam [39] hides adversarial perturbations by using the style transfer technique (e.g., changing a traffic sign to a rust style). Of course, there are also many works modify generic attributes of images, such as texture and color. Specifically, tAdv [32] infuses the texture of another image to generate adversarial examples. EdgeFool [33] craft adversarial perturbations by enhancing edge details of an image.

Compared with texture-based attacks, color-based ones usually yield more natural results since manipulating groups of pixels along dimensions is less perceptible to human eyes. To our knowledge, Hosseini&Poovendran [28] first proposed the Semantic Adversarial Examples (SAE) in this branch. Specifically, they convert the image from the RGB color space to the HSV one, then uniformly and randomly perturbed the H (Hue) and S (Saturation) channels of the entire image. Laidlaw&Feizi [30] also notice that RGB color space is not a perceptually uniform space. Thus, they operate in the CIELUV color space and propose a ReColorAdv that uses a specific function to uniformly transform all pixels belonging to the same color. Similar to ReColorAdv, ACE [31] proposes a simple piecewise-linear color adjustment filter to manipulate the image. The main difference is that ACE treats three channels of an image independent of each other when transformed. AGV [40] increases the attack effectiveness by filter composition. Unlike them, cAdv [32] performs color transformation using a

pretrained colorization network and jointly varies input hints and masks to manipulate adversarial examples.

Among existing unrestricted color attacks, ColorFool [29], which exploits image semantics to modify colors selectively, is the most related to ours. Nevertheless, there are several significant differences between our work and ColorFool. First, ColorFool requires manually selecting several human-sensitive semantic classes (i.e., sky, water, plants, and humans), while our approach does not need. Second, ColorFool adds uncontrolled perturbations on human-insensitive semantic classes, while our approach utilizes a color distribution library as a guide. Third, ColorFool optimizes adversarial examples only via multiple random trials, while our approach also utilizes the gradient of the substitute model to fine-tune results.

2.2 Adversarial Defenses

To mitigate the threat of adversarial examples, various defense methods have been proposed for the past few years. Generally, existing defense techniques can fall into two categories: adversarial training [41, 42, 43, 44] and input pre-process [45, 46, 47]. The former gains immunity to attacks by leveraging adversarial examples during the training phase. For example, Tramèr *et al.* [42] improve the black-box robustness by considering adversarial examples generated by other irrelevant models. Xie *et al.* [43] introduce feature denoising modules and train them on adversarial examples to build white-box robust models. Although adversarial training is usually the most robust strategy, it suffers from expensive training costs. To overcome this limitation, the input pre-process branch is designed to cure the infection of adversarial examples before feeding to DNNs. For example, Guo *et al.* [45] introduce several input transformation techniques (e.g., JPEG compression [45]) to recover from the adversarial perturbations. Xie *et al.* [48] propose R&P which mitigates the adversarial influence through random resizing and padding. Liao *et al.* [49] come up with a high-level representation guided denoiser (HGD) to reduce the effect of adversarial perturbations. Jia *et al.* [46] propose an end-to-end image compression model called ComDefend against adversarial examples. Naseer *et al.* [47] train a neural representation purifier model (NRP) that cleans adversarial perturbation based on the automatically derived supervision.

3 Methodology

3.1 Problem Formulation

Let $\boldsymbol{x}$ be a clean image with true label y and $\mathcal{F}_\theta(\cdot)$ denote the deep learning classifier with parameters θ. Formally, unrestricted attacks aim to craft human-imperceptible adversarial perturbations (may be very large in RGB color space) for $\boldsymbol{x}$ so that the resulting adversarial examples $\boldsymbol{x'}$ can induce $\mathcal{F}_\theta(\cdot)$ to misclassify:

$$\mathcal{F}_\theta(\boldsymbol{x'}) \neq y, \qquad s.t.\ \boldsymbol{x'}\,\text{is natural}. \tag{1}$$

In this paper, we focus on a more challenging black-box scenario (compared to the white-box scenario), where the target model's information (e.g., parameters and structure) is inaccessible. Therefore, adversarial examples can only be crafted via the accessible substitute model $\mathcal{F}_\phi(\cdot)$ and rely upon their transferability to fool target models.

3.2 Color Distribution Library

Existing unrestricted color attacks tend to sacrifice their flexibility so that they can generate human-imperceptible adversarial examples. For example, ReColorAdv [30] requires constraining the perturbation to a relatively small range, which cannot take full advantage of the "unrestricted" setting; cAdv [32] enforces the color belonging to the low-entropy cluster to remain unchanged, which inevitably reduces the attack space; ColorFool [29] manually splits an image into two parts and adds controlled noises on the human-sensitive part, which largely depends on the authors' intuition (but it varies from person to person).

To remedy the above limitation, we borrow the idea of [50] to construct a "distribution of color distributions" (DoD) for different semantic classes based on the ADE20K [35] dataset. Specifically, for each class, DoD provides 20 different dominant distribution sets (represented by the color distri-

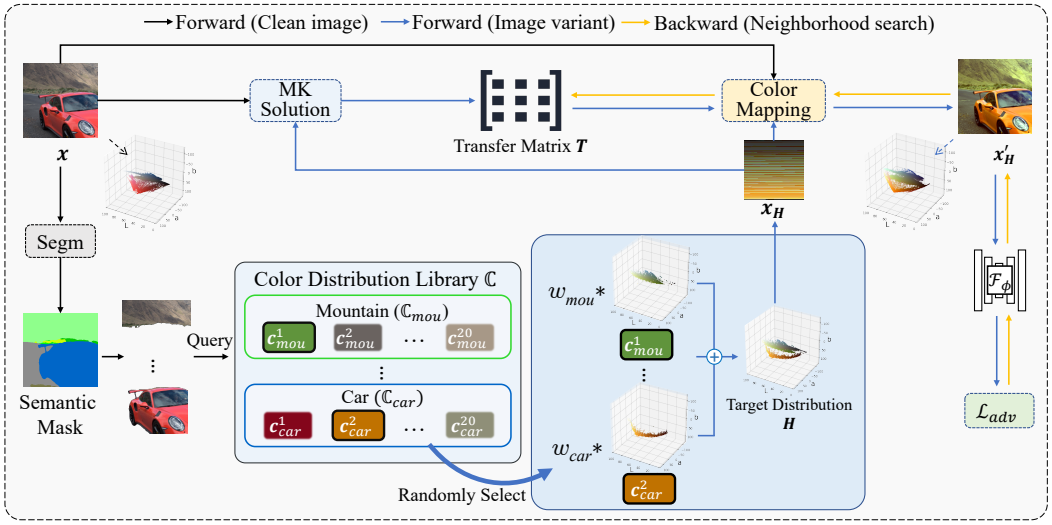


Figure 1: **The simplified pipeline of NCF** (optimizing one image variant without initialization reset). For the input image $\boldsymbol{x}$, we first obtain its mask by a segmentation model (Segm). Then we randomly assign a realistic color distribution $\boldsymbol{c}$ for each class via our color distribution library $\mathbb{C}$ and fuse them into a target $\boldsymbol{H}$ based on the proportion of corresponding area w in the image. Here $\boldsymbol{x}_H$ is an image (without spatial information) directly converted by $\boldsymbol{H}$ so that we can adopt Monge-Kantorovitch (MK) solution to obtain the transfer matrix $\boldsymbol{T}$. With it and Eq. 4, we can get the color mapping image variant $\boldsymbol{x'_H}$. Finally, we fine-tune $\boldsymbol{T}$ (i.e., neighborhood search) according to the loss $\mathcal{L}_{adv}$ so that more threatening adversarial examples can be mapped.

butions), which are obtained by a hierarchical clustering based on color palettes[2] of corresponding segmentation class in the dataset.

Since the color distributions in each distribution set are similar and averaging all color distributions of a specific distribution set may not yield a natural representation, we randomly select a color distribution to represent the overall color characteristics for simplicity. Therefore, for a semantic class $\tilde{y}$, its realistic color distribution space can be expressed as:

$$\mathbb{C}_{\tilde{y}} = \{\boldsymbol{c}^m_{\tilde{y}}\}^{20}_{m=1}, \tag{2}$$

where $\boldsymbol{c}^{\boldsymbol{m}}_{\tilde{\boldsymbol{y}}}$ is a randomly sampled distribution from the m-th set, and our color distribution library $\mathbb{C}$ can be denoted as:

$$\mathbb{C} = \mathbb{C}_1 \cup \mathbb{C}_2 \cup ... \cup \mathbb{C}_M, \tag{3}$$

where M denotes the total number of semantic classes contained in our library (see Appendix A for the pipeline of building color distribution library). In the following, we adopt this color distribution library to ensure the colors of adversarial examples are natural and coordinated, thus making our attack more flexible.

3.3 Natural Color Fool

To generate natural adversarial examples, precisely controlling the color perception is necessary. In this paper, we craft adversarial perturbations in CIELab color space where the perception is more uniform than RGB color space. The framework of our proposed **Natural Color Fool (NCF)** is illustrated in Figure 1 (refer to Appendix B for Algorithm of NCF). Formally, for a given image $\boldsymbol{x}$, we first obtain a semantic mask through a semantic segmentation model. Then, for each semantic class $\tilde{y}$ in $\boldsymbol{x}$, we randomly select a color distribution $\boldsymbol{c}^{\boldsymbol{i}}_{\tilde{\boldsymbol{y}}}$ from $\mathbb{C}_{\tilde{y}}$ in the color distribution library. After that, we fuse the color distributions of all semantic classes based on the proportion of images occupied by each class (e.g., w_{car}, w_{mou} in Figure 1), thus generating a target color distribution $\boldsymbol{H}$. With it, we can use the following Eq. 4 to generate a target image variant $\boldsymbol{x'_H}$ (refer to Figure 1)

[2]Color palette is a simplified color distribution composed of object's primary colors. It is used to reduce the computation cost of clustering.

which approximately replaces the color distribution of each semantic class of $\boldsymbol{x}$ with corresponding one in target $\boldsymbol{H}$ [36]:

$$\boldsymbol{x}'_{\boldsymbol{H}} = \boldsymbol{T}(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}}) + \boldsymbol{\mu}_{\boldsymbol{x}_{\boldsymbol{H}}}, \tag{4}$$

$$\boldsymbol{T}\boldsymbol{\Sigma}_{\boldsymbol{x}}\boldsymbol{T}^{\top} = \boldsymbol{\Sigma}_{\boldsymbol{x}_{\boldsymbol{H}}}, \tag{5}$$

where $\boldsymbol{T} \in \mathbb{R}^{3\times3}$ is the transfer matrix, $\boldsymbol{x}_{\boldsymbol{H}}$ is an image reconstructed by $\boldsymbol{H}$ but without spatial information[3] (see Figure 1), and $\boldsymbol{\mu}_{\boldsymbol{x}}, \boldsymbol{\mu}_{\boldsymbol{x}_{\boldsymbol{H}}}(\in \mathbb{R}^3), \boldsymbol{\Sigma}_{\boldsymbol{x}}, \boldsymbol{\Sigma}_{\boldsymbol{x}_{\boldsymbol{H}}}(\in \mathbb{R}^{3\times3})$ are channel means of $\boldsymbol{x}$, channel means of $\boldsymbol{x}_{\boldsymbol{H}}$, the covariance matrix of $\boldsymbol{x}$, and the covariance matrix of $\boldsymbol{x}_{\boldsymbol{H}}$, respectively.

Formally, there are numerous solutions [51, 52, 53, 54] for $\boldsymbol{T}$. However, most of them may not obtain an intended color mapping, i.e., only color proportions are expected. To tackle this issue, we follow Pitié *et al.* [36] and convert color mapping into a Monge–Kantorovich transportation problem, thus getting the solution for $\boldsymbol{T}$:

$$\boldsymbol{T} = \boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1/2}\left(\boldsymbol{\Sigma}_{\boldsymbol{x}}^{1/2}\boldsymbol{\Sigma}_{\boldsymbol{x}_{\boldsymbol{H}}}\boldsymbol{\Sigma}_{\boldsymbol{x}}^{1/2}\right)^{1/2}\boldsymbol{\Sigma}_{\boldsymbol{x}}^{-1/2}. \tag{6}$$

Intuitively, relying solely on randomly picking color distribution may limit the attack success rate. Since our color distribution library contains a wide variety of color combinations and we have accessible substitute models, there is no reason not to take advantage of them. Thus, we utilize the substitute model $\mathcal{F}_{\phi}(\cdot)$ to select a most adversarial distribution from η different color distributions sampled from our color distribution library:

$$\boldsymbol{H}^* = \underset{\boldsymbol{H}\in\mathbb{H}}{\arg\max}\ \mathcal{L}_{adv}(\mathcal{F}_{\phi}(\boldsymbol{x}'_{\boldsymbol{H}}), y) \tag{7}$$

where $\mathbb{H}$ is the set containing η different color distributions and $\mathcal{L}_{adv}$ is the C&W loss [18]:

$$\mathcal{L}_{adv}(\boldsymbol{z}, y) = \max\{\boldsymbol{z}_j : j \neq y\} - \boldsymbol{z}_y, \tag{8}$$

where $\boldsymbol{z}_j$ denotes the logit concerning the j-th class. To further boost our attack, we propose two techniques: **neighborhood search** and **initialization reset**. Specifically, the former aims to slightly adjust $\boldsymbol{T}$ (derived from Eq. 6) to improve the attack success rate while ensuring a natural mapping result (see the comparison of adversarial examples with or without adopting neighborhood search in Appendix E). Thus, the neighborhood search can be formulated as the following optimization problem:

$$\underset{\boldsymbol{T}'}{\arg\max}\ \mathcal{L}_{adv}(\mathcal{F}_{\phi}(\boldsymbol{T}'(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}}) + \boldsymbol{\mu}_{\boldsymbol{x}_{\boldsymbol{H}^*}}), y) \quad s.t.\ ||\boldsymbol{T} - \boldsymbol{T}'||_{\infty} \leq \epsilon, \tag{9}$$

where ϵ denotes the maximal perturbation size for $\boldsymbol{T}$. In our paper, we adopt MI-FGSM [21] with N iterations to optimize Eq. 9. Nonetheless, an adversarial transfer matrix $\boldsymbol{T}'$ found in the neighborhood of a specific $\boldsymbol{T}$ may not be effective, since the search space is indeed limited. To alleviate this issue, the initialization reset repeats the operation of selecting one from η randomly sampled color distributions (i.e., Eq. 7) K times, thus obtaining K ideal initial color distributions for the neighborhood search. Then we choose the best one among them:

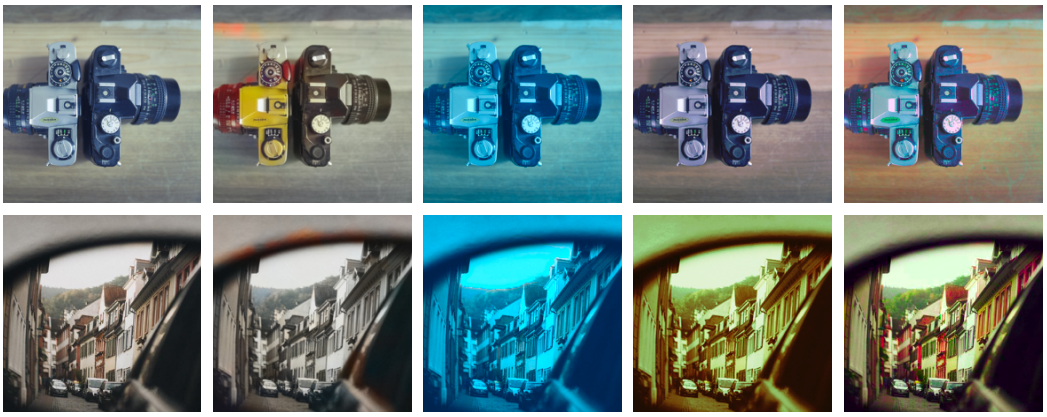
$$i' = \underset{i\in1,2,..,K}{\arg\max}\ \mathcal{L}_{adv}(\mathcal{F}_{\phi}(\boldsymbol{T}'_{\boldsymbol{i}}(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}}) + \boldsymbol{\mu}_{\boldsymbol{x}_{\boldsymbol{H}^*_{\boldsymbol{i}}}}), y), \tag{10}$$

where $\boldsymbol{H}^*_{\boldsymbol{i}}$ and $\boldsymbol{T}'_{\boldsymbol{i}}$ denote the resulting color distribution and corresponding transfer matrix of the i-th reset. Consequently, resulting adversarial examples can be obtained by:

$$\boldsymbol{x}' = \boldsymbol{T}'_{\boldsymbol{i}'}(\boldsymbol{x} - \boldsymbol{\mu}_{\boldsymbol{x}}) + \boldsymbol{\mu}_{\boldsymbol{x}_{\boldsymbol{H}^*_{\boldsymbol{i}'}}}. \tag{11}$$

In Figure 2, we show resulting adversarial examples generated by our proposed method and recent works, i.e., cAdv [32], ColorFool [29], and ACE [31] (see Appendix G for more comparisons). Notably, despite our NCF implements significant modifications to the color distribution of the original images, the adversarial examples still look very natural.

[3] $\boldsymbol{x}_{\boldsymbol{H}}$ is used to calculate the covariance matrix. Please note that calculation result is the same no matter how we reconstruct it.



Clean cAdv ColorFool ACE NCF (Ours)

Figure 2: Adversarial examples generated by different unrestricted color attacks and ours.

4 Experiments

In this section, we first introduce the setup for our experiments in Section 4.1. Then we report attack success rates on normally trained and defense models in Section 4.2. After that, we compare the image quality of adversarial examples in Section 4.3. Finally, we conduct a series of ablation studies in Section 4.4. To better analyze the adversarial example, we also provide an illustration of attention shift [55] in Appendix F.

4.1 Setup

Dataset. We conduct our experiments on the ImageNet-compatible Dataset[4]. This dataset is comprised of 1,000 images and is widely used in recent transfer-based attacks [26, 22, 23, 56, 57, 58, 59].

Models. In this paper, we evaluate the vulnerability of various convolutional neural networks (CNNs) and vision transformers (ViTs). For CNNs, we consider both normally trained models and defense models. Normally trained models include VGG-19 [60], ResNet-18 (Res-18), ResNet-50 (Res-50) [2], Inception-v3 (Inc-v3) [61], DenseNet-121 (Dense-121) [13], and MobileNet-v2 (Mobile-v2) [62]. Defense methods contain JPEG [45], Gray (gray-scale conversion), HGD [49], R&P [48], ComDefend [46], NRP [47],[5] Inc-v3$_{\text{ens3}}$, IncRes-v2$_{\text{ens}}$ [42], Res152$_{\text{B}}$, Res152$_{\text{D}}$ and ResNeXt$_{\text{DA}}$ [43]. For ViT, we consider normally trained ViT-S/16 (ViT-S) [63], XCiT-N12/16 (XCiT-N12) [64], and DeiT-S [63].

Implementation Details. In all experiments, the semantic segmentation model is Swin-T [65] (the influence of different segmentation models on the results can be found in Appendix H) and the color distribution library size $M = 150$, the number of random searches $\eta = 50$, the iteration of neighborhood search $N = 15$, the maximum perturbation of transfer matrix $\epsilon = 0.2$, the step size $\alpha = \epsilon/N \approx 0.013$, the momentum $u = 0.6$ (for MI-FGSM), and the reset number $K = 10$ (see Appendix C for ablation of K). We compare our proposed method with Semantic Adversarial Examples (SAE) [28], ReColorAdv [30], cAdv [32], ColorFool [29], and ACE [31]. The parameters of each competitor follow the corresponding default setting. Our experiments are run on an NVIDIA TITAN Xp GPU with 12GB of memory.

[4] https://github.com/cleverhans-lab/cleverhans/tree/master/cleverhans_v3.1.0/examples/nips17_adversarial_competition/dataset.

[5] For HGD and R&P, we adopt the official models used in corresponding papers. For JPEG, Gray, ComDefend, and NRP, we adopt VGG-19 as the target model.

4.2 Transferability Comparison

4.2.1 Results on Normally Trained Models

In this section, we compare our proposed method with SA [28], ReColorAdv [30], cAdv [32], ColorFool [29] and ACE [31] on a variety of normally trained CNNs and Vision Transformers (ViTs). Adversarial examples are crafted via Res-18, VGG-19, Mobile-V2, and Inc-V3, respectively. For the results of ensemble attack [21] and using ViTs as the substitute model, please refer to Appendix I and D.

The attack success rates are presented in Table 1. We can observe that our resulting adversarial examples usually achieve the highest transferability compared to those generated via state-of-the-art competitors. For example, when the substitute model is Res-18, only 46.9%, 39.9%, and 40.8% of adversarial examples crafted by SA, ReColorAdv, and cAdv can successfully fool VGG-19. In comparison, our NCF can achieve a much higher transferability of **72.1%**. Besides, we also consider the transferability from CNNs to ViTs, as their structures are quite different. As shown in Table 1, when transferring adversarial examples from Mobile-V2 to XCiT-N12, recent ColorFool and ACE only obtain success rates of 22.8% and 22.6%, respectively, while our method can get **56.4%** success rate. On average, adversarial examples crafted by our method are capable of achieving a **54.5%** success rate, which significantly outperforms state-of-the-art approaches by **15.0%**$\sim$**32.9%**. This result convincingly demonstrates the effectiveness of our method in fooling normally trained models.

Table 1: **Transferability comparison on normally trained CNNs and ViTs**. We report attack success rates (%) of each method and the leftmost model column denotes the substitute model ("*" means white-box attack results).

Model	Attacks	CNNs						Transformers		
		Res-18	VGG-19	Mobile-v2	Inc-v3	Dense-121	Res-50	ViT-S	XCiT-N12	DeiT-S
	Clean	16.1	11.4	12.8	19.2	7.9	7.5	13.3	13.7	5.8
Res-18	SAE	93.4*	46.9	45.5	31.3	36.5	37.0	44.5	37.4	22.2
	ReColorAdv	98.6*	39.9	47.3	38.2	37.2	38.1	21.4	36.7	17.3
	cAdv	**100.0***	40.8	48.2	41.6	43.0	41.2	34.4	44.9	30.4
	ColorFool	93.0*	27.8	30.5	28.1	19.8	22.9	35.5	22.3	9.2
	ACE	99.4*	26.0	27.2	27.6	19.9	18.3	21.6	22.4	9.1
	NCF (Ours)	92.9*	**72.1**	**72.7**	**48.3**	**55.3**	**66.7**	**53.0**	**55.3**	**32.8**
VGG-19	SAE	52.2	91.4*	48.8	32.3	39.3	39.0	48.3	37.6	24.3
	ReColorAdv	42.5	96.0*	41.9	33.2	33.8	31.7	20.4	33.4	16.6
	cAdv	54.0	**100.0***	48.0	43.7	43.4	40.7	38.8	43.9	**32.9**
	ColorFool	44.0	90.9*	36.5	29.2	23.5	26.6	42.2	25.6	9.6
	ACE	33.4	99.7*	27.8	28.3	21.6	18.0	20.7	21.6	9.5
	NCF (Ours)	**73.7**	93.3*	**70.3**	**49.4**	**53.6**	**64.3**	**56.5**	**53.5**	30.7
Mobile-V2	SAE	53.5	49.6	92.2*	34.5	38.1	39.3	46.6	37.7	23.3
	ReColorAdv	46.3	36.5	97.8*	36.4	32.4	34.4	20.7	36.7	20.0
	cAdv	54.7	39.9	**100.0***	42.8	44.3	39.1	36.0	44.1	30.8
	ColorFool	41.5	30.6	93.2*	28.1	23.3	24.5	39.7	22.8	9.4
	ACE	31.8	25.8	99.1*	26.7	20.0	19.0	20.3	22.6	9.3
	NCF (Ours)	**72.8**	**72.2**	92.7*	**50.0**	**54.4**	**66.2**	**55.4**	**56.4**	**32.6**
Inc-V3	SAE	49.5	45.8	45.4	78.2*	36.1	36.5	**46.6**	34.7	23.4
	ReColorAdv	26.5	19.9	21.9	96.2*	17.2	15.9	16.3	22.5	10.5
	cAdv	32.7	23.4	27.6	**99.8***	23.9	20.8	26.0	28.2	18.4
	ColorFool	40.4	31.8	35.4	84.1*	23.9	25.3	42.6	26.5	12.6
	ACE	28.6	24.1	23.9	96.9*	18.6	15.5	19.4	21.8	9.2
	NCF (Ours)	**57.7**	**57.7**	**56.8**	83.8*	**40.1**	**47.7**	45.3	**45.2**	**23.8**

4.2.2 Results on Defense Models

Threats posed by L_p-norm adversarial attacks [7, 21, 23, 57] promote the emergence of adversarial defenses [41, 43, 47]. However, whether these defense mechanisms are robust against unrestricted attacks has not been extensively explored. Therefore, in this section, we consider both input pre-process defenses (i.e., JPEG [45], Gray, R&P [48], HGD [49], ComDefend [46], and NRP [47]) and adversarially trained models (i.e., Inc-v3$_{\text{ens3}}$, IncRes-v2$_{\text{ens}}$, Res$_{\text{B}}$ [42], Res$_{\text{D}}$, and ResNext$_{\text{DA}}$ [43]) to fully investigate the performance of our method.

Table 2: **Transferability comparison on defense CNNs**. We report attack success rates (%) of each method and the substitute model is Inc-v3.

Methods	JPEG	Gray	R&P	HGD	ComDefend	NRP	Res_{B}	Res_{D}	$\text{ResNeXt}_{\text{DA}}$	$\text{Inc-v3}_{\text{ens3}}$	$\text{IncRes-v2}_{\text{ens}}$
Clean	17.4	26.7	2.4	2.0	28.2	16.8	29.2	24.4	19.8	7.7	4.2
SAE	52.7	33.6	13.0	11.5	62.2	53.7	55.6	53.5	50.1	19.8	13.7
ReColorAdv	26.9	34.4	7.8	5.6	38.0	27.6	32.9	28.8	24.3	14.7	9.3
cAdv	30.6	27.8	12.4	10.5	40.9	36.2	42.8	37.2	33.1	20.3	15.1
ColorFool	42.1	31.1	8.9	7.5	65.8	43.6	57.5	52.1	46.3	16.9	10.6
ACE	31.2	30.8	6.1	3.8	47.4	32.6	36.3	31.6	29.3	13.4	7.6
NCF (Ours)	**62.6**	**37.9**	**26.0**	**23.5**	**73.5**	**66.2**	**61.2**	**58.2**	**60.1**	**32.6**	**26.9**

The black-box success rates of unrestricted attacks are reported in Table 2. Similar to Table 1, our approach still consistently outperforms existing advanced methods by a large margin. On average, our proposed NCF is capable of making **48.1%** adversarial examples fool these defenses, while SA, ReColorAdv, cAdv, ColorFool, and ACE only obtain success rates of 38.1%, 22.8%, 27.9%, 34.8%, and 24.6%, respectively. Another interesting observation from Table 2 is that advanced defense algorithms are not necessarily effective when faced with unrestricted adversarial examples. For example, even white-box robust feature denoising models [43] are vulnerable, e.g., **60.1%** adversarial examples can fool $\text{ResNeXt}_{\text{DA}}$. We speculate the main reason is that the defense mechanism is intermediate feature denoising, while unrestricted color attacks mainly manipulate the color and do not generate significant high-frequency noise.

In addition, we notice that many input pre-process defenses with target model VGG-19 cannot defend against our proposed NCF, but instead boost the transferability of adversarial examples (compare to the VGG-19 results of Table 1). To better analyze this abnormal phenomenon, we visualize the purified images generated by JPEG, ComDefend, and NRP in Figure 3. Interestingly, we observe that all defenses have little effect on our unrestricted color perturbations. It is mainly because our adversarial examples are crafted using realistic colors as a guide and, therefore, can circumvent these defenses designed for common perturbation characteristics. Besides, we observe that JPEG and NRP suppress key features of "Great egret", and ComDefend generates many irrelevant repeating patterns. As discussed in [66], these side effects of defense methods can reduce the confidence of the model concerning the true class, resulting in our adversarial examples instead of boosting the transferability after being processed by these approaches. This result also reminds us that the robustness evaluation of defenses needs to be more comprehensive.

Clean | Adversarial (NCF) | Purified (JPEG) | Purified (ComDefend) | Purified (NRP)

Figure 3: **Visualization for the adversarial example and purified images**. This result shows that JPEG, ComDefend, and NRP cannot remove our unrestricted perturbation, but instead may distort key features (e.g., feather) of the object (the label is "Great egret"). Zoom in for better comparison.

4.3 Image Quality

Following ColorFool [29], we quantitatively evaluate image quality by a non-reference perceptual image quality measure called neural image assessment (NIMA) [37] which is highly correlated with human perception. In our paper, NIMA is based on the architecture of MobileNet [67] and is a composite of two models: one trains on AVA [68] for image technical assessment and another trains on TID2013 [69] for image aesthetics assessment. As reported in Table 3, image quality assessment results of our NCF are on par with those of state-of-the-art competitors. For NIMA (technical), both ACE and our NCF obtain a high score of over **4.939**, which is just **0.094** lower than clean images. The case of NIMA (aesthetic) is similar, where aesthetic scores of our adversarial examples and

clean images differ by only **0.097**, indicating that our NCF does not destroy the perceptual quality of images.

4.4 Ablation Studies

About neighborhood search and initialization reset. In this section, we investigate the effect of neighborhood search and initialization reset. Specifically, we use Inc-v3 to craft adversarial examples, and then evaluate the transferability towards Res-18, VGG-19, Mobile-v2, Dense-121, Res-50, ViT-S, XCiT-N12, and DeiT-S. The results are shown in Table 4. In terms of the white-box attack, NCF outperforms NCF-NS (NCF without NS), NCF-IR (NCF without IR), and NCF-IR-NS (NCF without IR and NS) by **17.7%**, **13.3%**, and **32.2%**, respectively. On the black-box attacks, neighborhood search and initialization reset bring an average performance improvement of **8.3%**, and **6.5%**. When both neighborhood search and initialization reset techniques are used, the transferability towards black-box models can be improved by an average of **12.5%**. The significant performance gain confirms the effectiveness of neighborhood search and initialization reset. In addition, we notice that even if neither proposed technique is used, our method still surpasses state-of-the-art competitors, except for SAE. We attribute this phenomenon to the flexibility of our approach, which can naturally modify the whole image without restriction. Note that NCF-IR-NS does not mean selecting random colors to attack. The difference of them is discussed in Appendix J.

Table 3: **Image quality comparison.**

Methods	NIMA ↑ (technical)	NIMA ↑ (aesthetics)
Clean	5.033	4.457
SAE	4.924	4.326
ReColorAdv	4.886	4.117
cAdv	4.718	4.220
ColorFool	4.918	**4.439**
ACE	**5.008**	4.328
NCF (Ours)	4.939	4.360

Table 4: **The effect of neighborhood search (NS) and initialization reset (IR).** Adversarial examples are crafted via Inc-v3 ("*" denotes white-box attack).

Methods	Inc-v3*	Res-18	VGG-19	Mobile-v2	Dense-121	Res-50	ViT-S	XCiT-N12	DeiT-S
NCF	**83.8***	**57.7**	**57.7**	**56.8**	**40.1**	**47.7**	**45.3**	**45.2**	**23.8**
NCF−NS	66.1*	49.3	47.2	46.9	31.2	39.1	41.7	34.4	18.1
NCF−IR	70.5*	50.2	48.8	48.5	34.4	39.7	43.7	38.0	19.0
NCF−IR−NS	51.6*	43.8	42.2	42.4	28.0	33.0	38.3	32.0	14.8

About color distribution library size. In this section, we investigate the influence of color distribution library size M on attack success rates. Specifically, we tune M from 0 to 150 with an interval of 15 and freeze other hyper-parameters, and adversarial examples are crafted via Inc-v3. Note that $M = 0$ means that our proposed library is not used, thus T is an identity matrix. The attack success rate is reported in Figure 4, where the dashed line indicates the white-box attack success rate and solid lines refer to the black-box ones. It can be observed that the library size M plays a key role in attack success rates. Concretely, the attack success rates increase rapidly at first, then gradually stabilize when the library size exceeds 30, and reach the maximum at $M = 150$ in most cases. For instance, when $M = 150$, NCF has success rates of 57.7%, 57.6%, 56.8%, 40.1%, and 47.6% on Res-18, VGG-19, Mobile-v2, Dense-121, and Res-50, respectively. When $M = 0$, it significantly degrades to 24.6%, 18.6%, 21.1%, 14.8%, and 14.2%, respectively. Intuitively, this is because the target color distribution trivially turns to be the corresponding distribution of the original image, and thus the space we perturb on the original image will be reduced. This convincingly demonstrates the effectiveness of building a class-rich color distribution library.

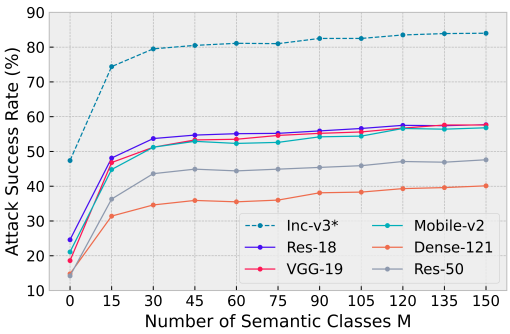


Figure 4: **Attack success rates (%) w.r.t. the number of semantic classes** M. The substitute model is Inc-v3.

5 Conclusion

In this paper, we propose a black-box unrestricted color attack called Natural Color Fool (NCF). Different from existing works which either rely on intuition and objective metrics or use relatively small changes to yield natural results, our method instead exploits color distribution to craft human-imperceptible, flexible, and highly transferable adversarial examples. Extensive experiments and visualizations demonstrate the effectiveness of our NCF. Notably, our adversarial examples can easily evade current L_p robust defense techniques. Furthermore, the transferability of our adversarial examples is even boosted after being processed by several input pre-process defenses. We hope our work can draw researchers' attention to unrestricted color attacks.

Limitation. In our paper, for each semantic class, we provide 20 different color distributions for choosing. Therefore, this library is discrete, which may be insufficient to model the whole color space. In the future, we will try to build a continuous color space to expand the spatial range and more accurately simulate the color range of objects.

Negative Societal Impacts. Adversarial examples crafted by our proposed unrestricted color attack look very natural but are with strong black-box transferability even towards defenses. Therefore, unscrupulous people may adopt our method to undermine real-world applications, which inevitably raises new concerns about AI safety.

6 Acknowledgments

This study is supported by grants from National Key R&D Program of China (2022YFC2009903/2022YFC2009900), the National Natural Science Foundation of China (Grant No. 62122018, No. 62020106008, No. 61772116, No. 61872064).

References

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

[2] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1, 6, 18

[3] Brian MacNamee, Padraig Cunningham, Stephen Byrne, and O. I. Corrigan. The problem of bias in training data in regression problems in medical decision support. *Artif. Intell. Medicine*, 2002. 1

[4] Jizong Peng, Ping Wang, Christian Desrosiers, and Marco Pedersoli. Self-paced contrastive learning for semi-supervised medical image segmentation with meta-labels. In *NeurIPS*, 2021. 1

[5] Yuchi Tian, Kexin Pei, Suman Jana, and Baishakhi Ray. Deeptest: automated testing of deep-neural-network-driven autonomous cars. In *ICSE*, 2018. 1

[6] 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

[7] Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015. 1, 7

[8] Muzammal Naseer, Kanchana Ranasinghe, Salman H. Khan, Fahad Shahbaz Khan, and Fatih Porikli. On improving adversarial transferability of vision transformers. In *ICLR*, 2021. 1

[9] Qilong Zhang, Xiaodan Li, Yuefeng Chen, Jingkuan Song, Lianli Gao, Yuan He, and Hui Xue. Beyond imagenet attack: Towards crafting adversarial examples for black-box domains. In *ICLR*, 2022. 1

[10] Yongqiang Chen, Han Yang, Yonggang Zhang, Kaili Ma, Tongliang Liu, Bo Han, and James Cheng. Understanding and improving graph injection attack by promoting unnoticeability. In *ICLR*, 2022. 1

[11] Yonggang Zhang, Ya Li, Tongliang Liu, and Xinmei Tian. Dual-path distillation: A unified framework to improve black-box attacks. In *ICML*, 2020. 1

[12] Yonggang Zhang, Xinmei Tian, Ya Li, Xinchao Wang, and Dacheng Tao. Principal component adversarial example. *IEEE Trans. Image Process.*, 2020. 1

[13] Gao Huang, Zhuang Liu, Laurens van der Maaten, and Kilian Q. Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 1, 6, 18

[14] 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. In *ICLR*, 2021. 1, 18

[15] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. In *ICCV*, 2021. 1

[16] Ji Zhang, Jingkuan Song, Lianli Gao, Ye Liu, and Heng Tao Shen. Progressive meta-learning with curriculum. *IEEE Trans. Circuits Syst. Video Technol.*, 2022. 1

[17] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: A simple and accurate method to fool deep neural networks. In *CVPR*, 2016. 1

[18] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE S&P*, 2017. 1, 5

[19] Zhengyu Zhao, Zhuoran Liu, and Martha A. Larson. Towards large yet imperceptible adversarial image perturbations with perceptual color distance. In *CVPR*, 2020. 1, 2

[20] Ye Liu, Yaya Cheng, Lianli Gao, Xianglong Liu, Qilong Zhang, and Jingkuan Song. Practical evaluation of adversarial robustness via adaptive auto attack. In *CVPR*, 2022. 1

[21] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *CVPR*, 2018. 1, 5, 7, 21

[22] 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, 6

[23] Lianli Gao, Qilong Zhang, Jingkuan Song, Xianglong Liu, and Heng Tao Shen. Patch-wise attack for fooling deep neural network. In *ECCV*, 2020. 1, 6, 7

[24] Yaya Cheng, Xiaosu Zhu, Qilong Zhang, Lianli Gao, and Jingkuan Song. Fast gradient non-sign methods. *arXiv preprint arXiv:2110.12734*, 2021. 1

[25] Muzammal Naseer, Kanchana Ranasinghe, Salman Khan, Fahad Khan, and Fatih Porikli. On improving adversarial transferability of vision transformers. In *ICLR*, 2022. 1, 19

[26] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *CVPR*, 2019. 1, 6

[27] Lianli Gao, Qilong Zhang, Jingkuan Song, and Heng Tao Shen. Patch-wise++ perturbation for adversarial targeted attacks. *CoRR*, abs/2012.15503, 2020. 1

[28] Hossein Hosseini and Radha Poovendran. Semantic adversarial examples. In *CVPR Workshops*, 2018. 2, 6, 7

[29] Ali Shahin Shamsabadi, Ricardo Sánchez-Matilla, and Andrea Cavallaro. Colorfool: Semantic adversarial colorization. In *CVPR*, 2020. 2, 3, 5, 6, 7, 8

[30] Cassidy Laidlaw and Soheil Feizi. Functional adversarial attacks. In *NeurIPS*, 2019. 2, 3, 6, 7

[31] Zhengyu Zhao, Zhuoran Liu, and Martha A. Larson. Adversarial color enhancement: Generating unrestricted adversarial images by optimizing a color filter. In *BMVC*, 2020. 2, 5, 6, 7

[32] Anand Bhattad, Min Jin Chong, Kaizhao Liang, Bo Li, and David A. Forsyth. Unrestricted adversarial examples via semantic manipulation. In *ICLR*, 2020. 2, 3, 5, 6, 7

[33] Ali Shahin Shamsabadi, Changjae Oh, and Andrea Cavallaro. Edgefool: an adversarial image enhancement filter. In *ICASSP*, 2020. 2

[34] Yang Song, Rui Shu, Nate Kushman, and Stefano Ermon. Constructing unrestricted adversarial examples with generative models. In *NeurIPS*, 2018. 2

[35] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ADE20K dataset. In *CVPR*, 2017. 2, 3, 17

[36] F. Pitie and A. Kokaram. The linear monge-kantorovitch linear colour mapping for example-based colour transfer. In *4th European Conference on Visual Media Production*, 2007. 2, 5

[37] Hossein Talebi Esfandarani and Peyman Milanfar. NIMA: neural image assessment. *IEEE Trans. Image Process.*, 2018. 2, 8

[38] Kaidi Xu, Gaoyuan Zhang, Sijia Liu, Quanfu Fan, Mengshu Sun, Hongge Chen, Pin-Yu Chen, Yanzhi Wang, and Xue Lin. Adversarial t-shirt! evading person detectors in a physical world. In *ECCV*, 2020. 2

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

[40] Alina Elena Baia, Alfredo Milani, and Valentina Poggioni. One for many: an instagram inspired black-box adversarial attack. 2021. 2

[41] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *ICLR*, 2018. 3, 7

[42] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian J. Goodfellow, Dan Boneh, and Patrick D. McDaniel. Ensemble adversarial training: Attacks and defenses. In *ICLR*, 2018. 3, 6, 7

[43] Cihang Xie, Yuxin Wu, Laurens van der Maaten, Alan L. Yuille, and Kaiming He. Feature denoising for improving adversarial robustness. In *CVPR*, 2019. 3, 6, 7, 8

[44] Yonggang Zhang, Mingming Gong, Tongliang Liu, Gang Niu, Xinmei Tian, Bo Han, Bernhard Schölkopf, and Kun Zhang. Adversarial robustness through the lens of causality. In *ICLR*, 2022. 3

[45] Chuan Guo, Mayank Rana, Moustapha Cissé, and Laurens van der Maaten. Countering adversarial images using input transformations. In *ICLR*, 2018. 3, 6, 7

[46] Xiaojun Jia, Xingxing Wei, Xiaochun Cao, and Hassan Foroosh. Comdefend: An efficient image compression model to defend adversarial examples. In *CVPR*, 2019. 3, 6, 7

[47] Muzammal Naseer, Salman H. Khan, Munawar Hayat, Fahad Shahbaz Khan, and Fatih Porikli. A self-supervised approach for adversarial robustness. In *CVPR*, 2020. 3, 6, 7

[48] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan L. Yuille. Mitigating adversarial effects through randomization. In *ICLR*, 2018. 3, 6, 7

[49] Fangzhou Liao, Ming Liang, Yinpeng Dong, Tianyu Pang, Xiaolin Hu, and Jun Zhu. Defense against adversarial attacks using high-level representation guided denoiser. In *CVPR*, 2018. 3, 6, 7

[50] Mahmoud Afifi, Brian L. Price, Scott Cohen, and Michael S. Brown. Image recoloring based on object color distributions. In *Eurographics*, 2019. 3, 17

[51] Erik Reinhard, Michael M. Stark, Peter Shirley, and James A. Ferwerda. Photographic tone reproduction for digital images. *ACM Trans. Graph.*, 2002. 5

[52] Arash Abadpour and Shohreh Kasaei. A fast and efficient fuzzy color transfer method. In *International Symposium on Signal Processing and Information Technology*. IEEE, 2004. 5

[53] Hiroaki Kotera. A scene-referred color transfer for pleasant imaging on display. In *ICIP*, 2005. 5

[54] François Pitié, Anil C. Kokaram, and Rozenn Dahyot. Automated colour grading using colour distribution transfer. *Comput. Vis. Image Underst.*, 2007. 5

[55] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 2017. 6, 19

[56] Lianli Gao, Yaya Cheng, Qilong Zhang, Xing Xu, and Jingkuan Song. Feature space targeted attacks by statistic alignment. In *IJCAI*, 2021. 6

[57] Zhibo Wang, Hengchang Guo, Zhifei Zhang, Wenxin Liu, Zhan Qin, and Kui Ren. Feature importance-aware transferable adversarial attacks. In *ICCV*, 2021. 6, 7

[58] Qilong Zhang, Xiaosu Zhu, Jingkuan Song, Lianli Gao, and Heng Tao Shen. Staircase sign method for boosting adversarial attacks. *CoRR*, abs/2104.09722, 2021. 6

[59] Yuyang Long, Qilong Zhang, Boheng Zeng, Lianli Gao, Xianglong Liu, Jian Zhang, and Jingkuan Song. Frequency domain model augmentation for adversarial attack. In *European Conference on Computer Vision*, 2022. 6

[60] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 6, 18

[61] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, 2016. 6, 18

[62] Mark Sandler, Andrew G. Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *CVPR*, 2018. 6, 18

[63] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. In *ICML*, 2021. 6, 18

[64] Alaaeldin Ali, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, and Hervé Jégou. Xcit: Cross-covariance image transformers. In *NeurIPS*, 2021. 6, 18

[65] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, pages 9992–10002. IEEE, 2021. 6, 18, 20

[66] Qilong Zhang, Chaoning Zhang, Chaoqun Li, Jingkuan Song, Lianli Gao, and Heng Tao Shen. Practical no-box adversarial attacks with training-free hybrid image transformation. *CoRR*, abs/2203.04607, 2022. 8

[67] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. 8

[68] Naila Murray, Luca Marchesotti, and Florent Perronnin. AVA: A large-scale database for aesthetic visual analysis. In *CVPR*, 2012. 8

[69] Nikolay N. Ponomarenko, Oleg Ieremeiev, Vladimir V. Lukin, Karen O. Egiazarian, Lina Jin, Jaakko Astola, Benoît Vozel, Kacem Chehdi, Marco Carli, Federica Battisti, and C.-C. Jay Kuo. Color image database TID2013: peculiarities and preliminary results. In *EUVIP*, 2013. 8

[70] Namuk Park and Songkuk Kim. How do vision transformers work? In *ICLR*, 2022. 18

[71] Byeongho Heo, Sangdoo Yun, Dongyoon Han, Sanghyuk Chun, Junsuk Choe, and Seong Joon Oh. Rethinking spatial dimensions of vision transformers. In *ICCV*, 2021. 18

[72] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. 2020. 20

[73] Hang Zhang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, Yue Sun, Tong He, Jonas Muller, R. Manmatha, Mu Li, and Alexander Smola. Resnest: Split-attention networks. *arXiv preprint arXiv:2004.08955*, 2020. 20

Checklist

1. For all authors...
   (a) Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope? [Yes]
   (b) Did you describe the limitations of your work? [Yes] See Section 5.
   (c) Did you discuss any potential negative societal impacts of your work? [Yes] See Section 5.
   (d) Have you read the ethics review guidelines and ensured that your paper conforms to them? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [N/A]
   (b) Did you include complete proofs of all theoretical results? [N/A]
3. If you ran experiments...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes] The code and instructions have been submitted as the supplemental material.
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes] Our experiments did not involve any model training, but all the algorithm details were specified.
   (c) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [No] We fixed the random seed for all the experiments.
   (d) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [No] We did not mention the run time of the attacks due to the space limitation, but our hardware setting is available in Section 4.1.
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes] see Section 4.1.
   (b) Did you mention the license of the assets? [Yes] In a footnote of Section 4.1.
   (c) Did you include any new assets either in the supplemental material or as a URL? [No] We did not use any new assets.
   (d) Did you discuss whether and how consent was obtained from people whose data you're using/curating? [Yes] In a footnote of Section 4.1.
   (e) Did you discuss whether the data you are using/curating contains personally identifiable information or offensive content? [No]
5. If you used crowdsourcing or conducted research with human subjects...
   (a) Did you include the full text of instructions given to participants and screenshots, if applicable? [N/A]
   (b) Did you describe any potential participant risks, with links to Institutional Review Board (IRB) approvals, if applicable? [N/A]
   (c) Did you include the estimated hourly wage paid to participants and the total amount spent on participant compensation? [N/A]

Natural Color Fool: Towards Boosting Black-box Unrestricted Attacks (Supplementary Material)

CONTENTS

A Color Distribution Library 17

B Algorithm 17

C Initialization Reset 18

D Transferability on ViTs 18

E Neighborhood Search Impact on Human Perception 19

F Visualization of Attention 19

G Visualization of Adversarial Examples 20

H The Influence of Segmentation Models 20

I The Effect of Ensemble Attack 21

J Attack by Selecting Random Colors 21

A Color Distribution Library

In this section, we explain how to build our color distribution library. First, all semantic classes are extracted based on the images and annotations of the ADE20K training set [35] (ADE20K contains 150 semantic classes). The semantic classes are clustered into a set of 20 different dominant distribution sets according to their color styles using hierarchical clustering, which is the "distribution of color distribution" (DoD) [50]. Then, since the color distributions of semantic classes in each cluster are similar, we select one object in each cluster to represent that style. Finally, we extract the color distributions to form our color distribution library. The pipeline is shown in Figure 5.

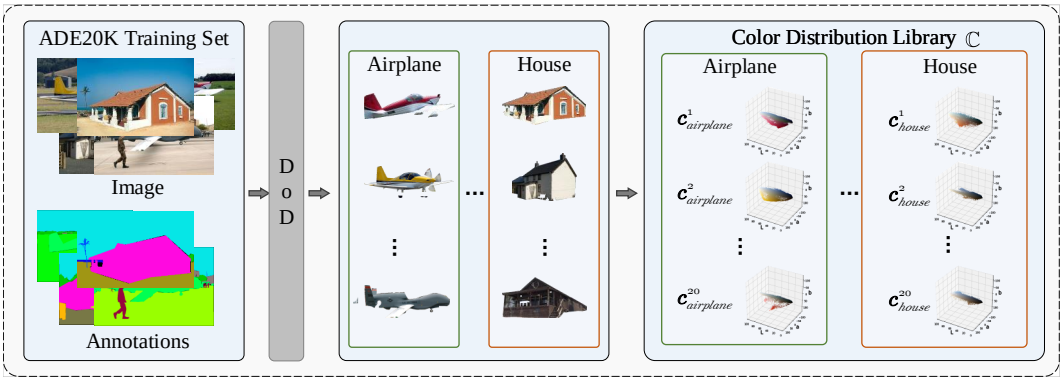


Figure 5: **The pipeline of color distribution library.** We select 20 objects with different color styles for each semantic class and extract their color distributions to build a color distribution library.

B Algorithm

In this section, we provide the algorithm of NCF (see Algorithm 1).

Algorithm 1 Natural Color Fool

Input:
$\boldsymbol{x}$: clean image, y: label, $\tilde{y}$: the semantic class, $\mathcal{S}$: semantic segmentation model,
$\mathcal{F}_\phi(\cdot)$: the substitute model, $\mathbb{C}$: color distribution library, K: the restart number,
N: the iteration of NS, u: the momentum, α: the step size,
Σ: covariance matrix, μ: mean

Output: $\boldsymbol{x}'$: adversarial image

Obtain all semantic classes $\tilde{Y}$ of image $\boldsymbol{x}$ by $\mathcal{S}$.
Calculate each semantic class's pixel ratio w in the $\boldsymbol{x}$.
$\bar{\boldsymbol{x}} \leftarrow rgb2lab(x)$
for $i \leftarrow 1$ to K **do** ▷ Initialization Reset.
  for $j \leftarrow 1$ to η **do**
    Randomly sample the color distribution $\boldsymbol{c}_{\tilde{y}}$ of each semantic class $\tilde{y}$ from the library $\mathbb{C}_{\tilde{y}}$
    $\boldsymbol{H}_j \leftarrow \sum_{\tilde{y} \in \tilde{Y}} w_{\tilde{y}} * \boldsymbol{c}_{\tilde{y}}$
    $\bar{\boldsymbol{x}}_{\boldsymbol{H}_j} \leftarrow$ Convert $\boldsymbol{H}_j$ to an image without spatial information.
    $\Sigma_{\bar{\boldsymbol{x}}_{\boldsymbol{H}_j}}, \mu_{\bar{x}_{H_j}} \leftarrow$Calculate covariance matrix and channel means for $\bar{\boldsymbol{x}}_{\boldsymbol{H}_j}$.
    $\boldsymbol{T} \leftarrow \mathrm{MKSolution}(\Sigma_{\bar{x}}, \Sigma_{\bar{x}_{H_j}})$ (see Eq.(6))
    $\bar{\boldsymbol{x}}'_{\boldsymbol{H}_j} \leftarrow \boldsymbol{T}(\bar{x} - \mu_{\bar{x}}) + \mu_{\bar{\boldsymbol{x}}_{\boldsymbol{H}_j}}$
  end for
  $\mathbb{H} \leftarrow \{\boldsymbol{H}_j\}_{j=1}^{\eta}$
  $\boldsymbol{H}^* \leftarrow \arg\max_{\boldsymbol{H} \in \mathbb{H}} \mathcal{L}_{adv}(\mathcal{F}_\phi(\boldsymbol{x}'_{\boldsymbol{H}}), y)$
  $\Sigma_{\bar{\boldsymbol{x}}_{\boldsymbol{H}^*}}, \mu_{\bar{x}_{H^*}} \leftarrow$Calculate covariance matrix and channel means for $\bar{\boldsymbol{x}}_{\boldsymbol{H}^*}$.
  $\boldsymbol{T}^* \leftarrow \mathrm{MKSoulution}(\Sigma_{\bar{x}}, \Sigma_{\bar{x}_{H^*}})$ (see Eq.(6))

$\mu^* \leftarrow \mu_{\bar{x}_{H^*}}$
Initialize $\boldsymbol{T}'_0 \leftarrow \boldsymbol{T}^*$
for $n \leftarrow 1$ to N **do** ▷ Neighborhood Search.
$\bar{\boldsymbol{x}}'_i \leftarrow \boldsymbol{T}'_{n-1}(\bar{\boldsymbol{x}} - \mu_{\bar{\boldsymbol{x}}}) + \mu^*$
$\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{T}'} \mathcal{L}_{adv}(\mathcal{F}_\phi(lab2rgb(\bar{\boldsymbol{x}}'_i)), y)$
$\boldsymbol{g}_n \leftarrow u \cdot \boldsymbol{g}_{n-1} + \frac{\boldsymbol{g}}{||\boldsymbol{g}||_1}$
$\boldsymbol{T}'_n \leftarrow \boldsymbol{T}'_{n-1} + \alpha \cdot \text{sign}(\boldsymbol{g}_n)$
end for
$\boldsymbol{x}'_i \leftarrow lab2rgb(\bar{\boldsymbol{x}}'_i)$
end for
$\boldsymbol{i}' \leftarrow \underset{i \in 1,2,\ldots,K}{\arg\max}\, \mathcal{L}_{adv}(\mathcal{F}_\phi(\boldsymbol{x}'_i), y)$
$\boldsymbol{x}' \leftarrow \boldsymbol{x}'_{i'}$
return $\boldsymbol{x}'$

C Initialization Reset

As mentioned in Section 3.3, initialization reset (IR) avoids the optimization of convert matrix to fall into local optimum. Therefore, we study the appropriate value for the number of resets in this section. Our adversarial examples are crafted via Inc-v3 by NCF with different resets, i.e., from 1 to 20. We report the attack success rate on the victim model in Figure 6. As demonstrated in the figure, in the beginning, there is an approximate positive relationship between the attack success rate and the number of resets. Once $K = 10$ or larger, the attack success rate maintains stable. We choose $K = 10$ to reduce the computational overhead, where the attack successes rate is 84.0%, 57.7%, 57.6%, 56.8%, 40.1%, 47.6% for Inc-v3 [61], Res-18 [2], VGG-19 [60], Mobile-v2 [62], Dense-121 [13] and Res-50 [2], respectively.

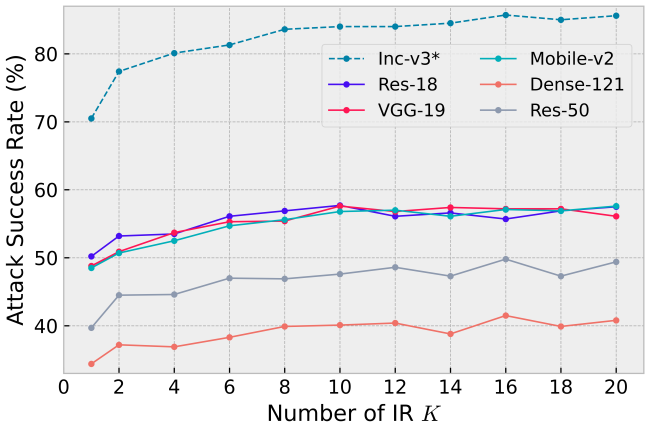


Figure 6: **Attack success rates (%) w.r.t. the number of IR K.** The substitute model is Inc-v3.

D Transferability on ViTs

Recent work [70] points out that the vision transformer models are more concerned with low-frequency information, thus we further evaluate attack success rate of black-box when Transformer models are used as substitute models. Specifically, we generate adversarial examples using ViT-S/16 (ViT-S) [63] and XCiT-N12/16 (XCiT-N12) [64] as substitute models and evaluate the adversarial examples on multiple Transformer models (including ViT-S,XCiT-N12,DeiT-S [63],ViT-B [14],Swin-T [65],PiT-Ti [71]) and multiple CNNs (including Res-18 [2], VGG-19 [60], Mobile-v2 [62]). Table 5 summarizes the results on different black-box models. We can observe that our NCF usually achieves the highest transferability on black-box. In particular, when transferring from ViTs to CNNs, NCF achieves an average success rate of 59.8%, but other attacks only achieve 48.3% (SAE), 29.9% (ReColorAdv), 36.8% (cAdv), 35.1% (ColorFool), 27.3% (ACE).

We observe that the transferability of adversarial examples between ViTs is weaker than different CNNs (see Section 4.2.1). This is consistent with the observation in [25]. It may be since that ViTs have multiple self-attention blocks that can generate class tokens independently. However, our loss only utilizes the last logit (which is equivalent to utilizing only the last class token). The discriminative information in the previous tokens is not directly utilized, which results the poor transferability of the adversarial examples between different ViTs.

Table 5: **Transferability on ViTs.** We report attack success rates (%) of each method and the leftmost model column denotes the substitute model ("*" means white-box attack results).

Model	Attacks	Transformers						CNNs		
		ViT-S	XCiT-N12	DeiT-S	ViT-B	Swin-T	PiT-Ti	Res-18	VGG-19	Mobile-v2
	Clean	13.3	13.7	5.8	10.7	5.0	11.6	16.1	11.4	12.8
ViT-S	SAE	98.2*	38.7	24.2	37.5	19.6	34.0	49.9	47.6	46.5
	ReColorAdv	96.2*	27.4	20.4	30.6	13.8	24.5	27.6	23.7	26.3
	cAdv	**100.0***	36.5	**32.8**	42.4	19.3	35.4	35.5	27.1	33.4
	ColorFool	99.2*	23.6	11.0	24.3	7.6	21.5	35.0	26.7	29.3
	ACE	99.7*	21.1	10.5	20.0	7.1	19.6	29.8	23.8	28.4
	NCF (Ours)	93.9*	**42.4**	25.9	**62.7**	**20.4**	**39.9**	**57.5**	**55.4**	**54.4**
XCiT-N12	SAE	49.3	86.5*	25.9	36.9	18.9	39.9	51.2	47.5	47.0
	ReColorAdv	21.2	95.9*	20.2	16.4	14.7	37.4	36.3	29.5	35.9
	cAdv	39.4	**100.0***	**38.5**	33.0	26.5	53.9	46.9	36.1	41.7
	ColorFool	47.6	85.9	12.6	34.4	9.7	26.5	43.9	36.5	39.4
	ACE	19.5	98.7	9.8	15.0	6.3	21.0	29.5	24.9	27.3
	NCF (Ours)	**54.1**	89.1*	36.1	**38.5**	**27.3**	**55.8**	**65.1**	**63.7**	**62.6**

E Neighborhood Search Impact on Human Perception

In this section, we study the impact of neighborhood search (NS) on human perception. Figure 7 shows the adversarial examples with and without NS, and we observe that our NS strategy has almost no effect on human perception.

Figure 7: **Visualizations of NCF-IR and NCF-IR-NS.** We use Inc-v3 as an alternative model to generate adversarial examples. We observe from the results that there is no perceptual difference after using NS.

F Visualization of Attention

In this section, we show the Gradient-weighted Class Activation Mapping (Grad-CAM) [55] of different adversarial examples. Figure 8 illustrates that all of SAE, ReColorAdv, cAdv, ColorFool, and ACE have difficulty shifting the attention of the model, while our NCF can dramatically it, i.e., no longer on the target object.

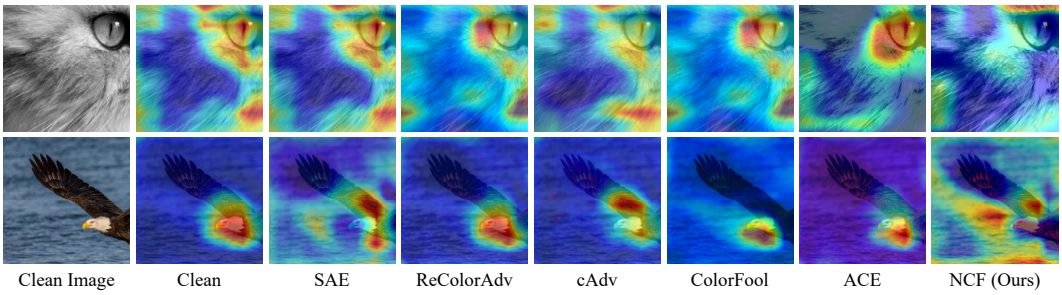


Figure 8: **Visualization of attention.** We use Inc-v3 as the alternative model to generate adversarial examples, and then visualize the attention on VGG-19.

G Visualization of Adversarial Examples

In this section, we show some adversarial examples. In Figure 9, we observe that SAE is prone to global color distortion due to the unbounded range of variation. ReColorAdv limits the perturbation to a small range, thus its adversarial examples are less perceptually different from the clean images. However, some areas of color gradients (e.g., the "clouds" in the second row of Figure 9) may produce some perceptual anomalies. cAdv relies on the colorization model, and some color anomalies are likely to occur locally in the image. ColorFool does not set the perturbation range in non-sensitive areas, resulting in non-sensitive areas prone to color distortion. ACE is similar to ReColorAdv in that sometimes details are lost. (e.g., the "plants" in the third row of Figure 9). NCF transforms the color distribution so that the images' details are preserved.

H The Influence of Segmentation Models

In this section, we compare the performance of our NCF under different semantic segmentation models pre-trained on ADE20K (including Swin-T [65], OCRNet [72] and Deeplabv3+ [73]). As indicated in Table 6, segmentation models have impact on the attack success rates of resulting adversarial examples. Among these models, Swin-T is usually the best choice for our NCF. Therefore, in our paper, we choose it to segment inputs. Note that even if the segmentation model affects our method, the lowest black-box attack success rate of NCF is still much higher than the existing methods.

Table 6: The influence of different segmentation models on attack success rates. ("*" denotes the white-box attack)

Segm	Res-18*	VGG-19	Mobile-v2	Inc-v3	Dense-121	Res-50	ViT-S
Swin-T	**92.9***	**72.1**	**72.7**	**48.3**	**55.3**	**66.7**	53.0
OCRNet	89.9*	69.1	67.1	44.2	50.6	61.1	**56.5**
Deeplabv3+	91.0*	68.0	68.6	45.3	49.2	62.0	54.0

I The Effect of Ensemble Attack

As for ensemble model attack (fusing the logits of multiple models like [21]), here we report the result of NCF. As indicated in Table 7, the attack success rate of NCF can be further improved when crafting via an ensemble of models.

Table 7: Comparison of ensemble attack and single model attack. We report attack success rates (%) of NCF and the leftmost model column denotes the substitute model, where Ensemble means an ensmeble of Res-18, VGG-19 and Mobile-v2.

Model	Dense-121	Res-50	ViT-S	XCiT-N12	DeiT-S
Clean	7.9	7.5	13.3	13.7	5.8
Res-18	55.3	66.7	53.0	55.3	32.8
VGG-19	53.6	64.3	56.5	53.5	30.7
Mobile-V2	54.4	66.2	55.4	56.4	32.6
Ensemble	**63.5**	**71.6**	**59.7**	**61.7**	**37.0**

J Attack by Selecting Random Colors

In this section, we discuss the difference between NCF-IR-NS and random color attack. Formally, NCF-IR-NS does not mean selecting random colors to attack. Specifically, it first generates a set of adversarial examples with different color distributions and then selects the best example from them based on the loss of the white-box model to attack. Therefore, NCF-IR-NS is close to a white-box attack.

To support our claim, we evaluate the performance of random color attack (NCF-IR-NS-), i.e., randomly select colors for each semantic class and use the resulting adversarial examples to attack. As demonstrated in Table 8, the performance of NCF-IR-NS- is much lower than NCF-IR-NS. For example, NCF-IR-NS- only achieves a 27.3% (degraded from 51.6%) success rate on Inc-v3. Thus, directly using random colors to generate adversarial examples is ineffective.

Table 8: The attack success rate of using white-box information and not using it. NCF-IR-NS using Inc-v3 as the substitute model. ("*" denotes the white-box attack)

Methods	Inc-v3*	Res-18	VGG-19	Mobile-v2	Dense-121	Res-50	ViT-S	XCiT-N12	DeiT-S
Clean	19.2*	16.1	11.4	12.8	7.9	7.5	13.3	13.7	5.8
NCF-IR-NS	**51.6***	**43.8**	**42.2**	**42.4**	**28.0**	**33.0**	**38.3**	**32.0**	**14.8**
NCF-IR-NS-*	27.3	34.8	30.9	31.1	20.5	24.2	32.7	25.0	11.6

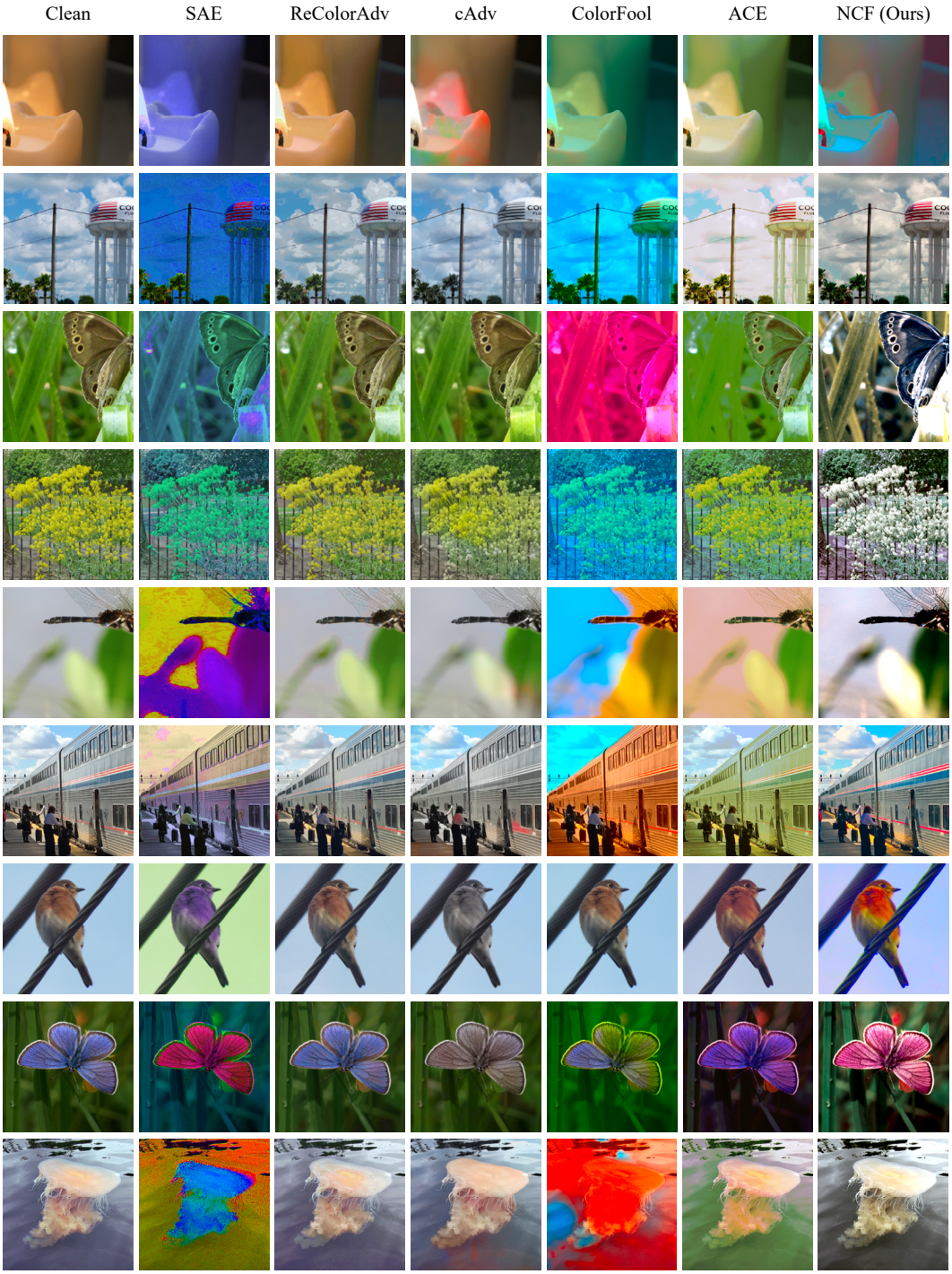


Figure 9: **Visualization of adversarial examples.** All adversarial examples are generated on Inc-v3.