DifAttack: Query-Efficient Black-Box Attack via Disentangled Feature Space

Jun Liu[1], **Jiantao Zhou**[1*] **Jiandian Zeng**[2], **Jinyu Tian**[3]

[1] State Key Laboratory of Internet of Things for Smart City,
Department of Computer and Information Science, University of Macau
[2] Institute of Artificial Intelligence and Future Networks, Beijing Normal University
[3] School of Computer Science and Engineering, Macau University of Science and Technology
{yc07453, jtzhou}@um.edu.mo, jiandian@bnu.edu.cn, jytian@must.edu.mo

Abstract

This work investigates efficient score-based black-box adversarial attacks with a high Attack Success Rate (ASR) and good generalizability. We design a novel attack method based on a **Di**sentangled **F**eature space, called **DifAttack**, which differs significantly from the existing ones operating over the entire feature space. Specifically, DifAttack firstly disentangles an image's latent feature into an *adversarial feature* and a *visual feature*, where the former dominates the adversarial capability of an image, while the latter largely determines its visual appearance. We train an autoencoder for the disentanglement by using pairs of clean images and their Adversarial Examples (AEs) generated from available surrogate models via white-box attack methods. Eventually, DifAttack iteratively optimizes the adversarial feature according to the query feedback from the victim model until a successful AE is generated, while keeping the visual feature unaltered. In addition, due to the avoidance of using surrogate models' gradient information when optimizing AEs for black-box models, our proposed DifAttack inherently possesses better attack capability in the open-set scenario, where the training dataset of the victim model is unknown. Extensive experimental results demonstrate that our method achieves significant improvements in ASR and query efficiency simultaneously, especially in the targeted attack and open-set scenarios. The code will be available at https://github.com/csjunjun/DifAttack.git soon.

Introduction

It has been observed that meticulously crafted Adversarial Examples (AEs) are beneficial for evaluating and improving the robustness of Deep Neural Networks (DNNs) (Dong et al. 2020; Croce and Hein 2020b). AEs refer to images that deceive DNNs by incorporating imperceptible perturbations onto clean images. The methods for generating AEs can be roughly classified into three categories based on the information of victim DNNs available to the attacker. The first category is the white-box attack (Dong et al. 2018; Croce and Hein 2020a; Zhang et al. 2021), in which the attack method has access to the architecture and weights of the victim model. The second category is called the black-box attack, where the attacker can only obtain the model's output. The case where the model's output contains both categories and scores is known as a score-based black-box attack, while the situation where only categories are included is named a decision-based black-box attack (Brendel, Rauber, and Bethge 2018). The third category refers to the gray-box attack (Guo et al. 2018), where attackers can obtain models' knowledge that falls between white-box attacks and black-box attacks. Given that contemporary potent classification APIs (Google 2023; Imagga 2023) generally provide image prediction categories alongside scores, this work primarily focuses on practical score-based black-box attacks.

We observe that current state-of-the-art (SOTA) score-based black-box attack methods predominantly exhibit two aspects that can be improved. On the one hand, prior techniques that seek AEs by optimizing image features usually operate within the entire feature space, such as AdvFlow (Mohaghegh Dolatabadi, Erfani, and Leckie 2020), TREMBA (Huang and Zhang 2020) and CGA (Feng et al. 2022), without embracing the disentanglement of adversarial capability and visual characteristics in the image feature. Optimizing the entire feature of images perhaps results in substantial pixel perturbations that will encounter truncation by the predefined disturbance constraint, thus diminishing the optimization efficacy of the features and eventually increasing query numbers. Another possible chain reaction is the decrease in the Attack Success Rate (ASR) caused by the limited query numbers.

To improve the unsatisfactory ASR and query efficiency, our DifAttack disentangles an image's latent feature into an adversarial feature and a visual feature, where the former dominates the adversarial capability of an image, while the latter largely determines its visual appearance. Such a disentanglement is feasible because the adversarial capability is mainly entwined with the decision boundary of classifiers and is relatively independent of the intrinsic image signal, whereas a contrast holds true for the visual appearance. Afterward, in DifAttack, the adversarial feature is iteratively optimized according to the query feedback from the victim model, while keeping the visual feature unaltered, until a successful AE can be reconstructed from the fusion of these two features.

On the other hand, some score-based attack techniques including P-RGF (Cheng et al. 2019) and GFCS (Lord, Mueller, and Bertinetto 2022), exploit the gradients from local white-box surrogate models to update AEs during each query. These methods are effective when image categories

[*] Corresponding author.

of the training dataset for surrogate models and the victim model are the same, namely a **close-set** black-box attack scenario analogy to the close-set recognition setting (Vaze et al. 2021). When it comes to an **open-set** black-box attack, wherein the training dataset of the victim model is unknown, the effectiveness of these methods declines, as the possible incongruence between the output categories of surrogate models and the victim model hampers the calculation of gradients from surrogate models. Although there are other approaches, such as SimBA (Guo et al. 2019) and CGA (Feng et al. 2022), which are workable in both the open-set and close-set scenarios. These methods still face challenges in terms of query efficiency and ASR, especially in the challenging targeted attack. Our DifAttack circumvents the need to compute gradients of surrogate models when updating AEs to be queried. Instead, before embarking upon querying the victim model, we leverage AEs generated by performing white-box attack methods on surrogate models to learn how to extract a disentangled adversarial feature and visual feature from the image's latent feature, which enables DifAttack to more effectively conduct black-box attacks in the open-set scenarios. The efficiency of DifAttack in the open-set scenarios can be attributed to the fact that the disentangled feature space for optimizing AEs is also responsible for the image reconstruction task, which means that an image can still be reconstructed into its original form with minimal loss after being mapped into our disentangled feature space. In contrast, the feature space in other methods, such as TREMBA and CGA, is directly used to generate adversarial perturbations that are tightly bound to the data distribution of the training dataset. The latent feature for image reconstruction in DifAttack has better generalizability on unknown datasets than the feature space learned from adversarial perturbation distributions of a specific dataset.

Specifically, in our DifAttack, we firstly train an autoencoder equipped with our proposed **D**ecouple-**F**usion (**DF**) module to disentangle adversarial features as well as visual features and to realize image reconstructions from these features. Then, in the attack stage, we iteratively update the adversarial feature using the natural evolution strategy (Wierstra et al. 2014), eventually reconstructing an AE from the fusion of the perturbed adversarial feature and the clean image's visual feature. To the best of our knowledge, this may be the first work that employs the disentangled adversarial and visual features for black-box adversarial attacks. The contributions of our work can be summarized as follows:

- We design a new disentanglement method to distill an adversarial feature, which has a significant impact on the image's adversarial capability while minimizing its influence on visual perception, and a visual feature that has the opposite attributes, from an image's latent feature.
- Based on this disentanglement, we propose a new black-box attack method, DifAttack, which generates AEs by optimizing the adversarial feature according to the query feedback while maintaining the visual feature invariant.
- Experimental results demonstrate that our approach outperforms SOTA score-based black-box attack methods in ASR and query numbers simultaneously in almost all cases, especially in targeted attacks and open-set scenarios. DifAttack also achieves perfect (100%) ASR in most datasets and reduces query numbers by over 40%.

Related Works

Score-based Black-box Attack Methods

Score-based black-box attack methods can be broadly categorized into three main classes: **transfer-based**, **query-based**, and **query-and-transfer-based** attacks (Feng et al. 2022). 1) **Transfer-based** approaches query the black-box model once using AEs generated on local surrogate models, as explored in (Qin et al. 2023) and etc. Such approaches take advantage of the transferability of AEs across classifiers with distinct architectures, but their ASRs generally fall behind query-based attack methods. 2) The **query-based** techniques tackle the black-box optimization through iteratively querying the victim model, including NES (Ilyas et al. 2018), Signhunter (Al-Dujaili and O'Reilly 2020), $\mathcal{N}$Attack (Li et al. 2019), SimBA (Guo et al. 2019), etc., which exhibit an augmented ASR compared to transfer-based approaches, albeit demanding a greater number of queries. 3) The **query-and-transfer-based** methods amalgamate the merits of transfer-based and query-based techniques, resulting in an elevated ASR and diminished queries. Certain techniques, such as Subspace (Guo, Yan, and Zhang 2019), P-RGF (Cheng et al. 2019), GFCS (Lord, Mueller, and Bertinetto 2022) and BASES (Cai et al. 2022), update queried AEs by performing white-box attacks against surrogate models with the guidance of the query feedback. Other methods learn adversarial distributions by leveraging pairs of clean images and their AEs obtained by attacking surrogate models, including TREMBA (Huang and Zhang 2020) and CGA (Feng et al. 2022), or approximate the adversarial distribution with the clean data distribution, such as AdvFlow (Mohaghegh Dolatabadi, Erfani, and Leckie 2020). Then, the learned feature space for mapping the adversarial distribution is adopted as the search space of AEs. Among them, CGA exhibits a remarkable improvement in query efficiency. However, its query number for targeted attacks on ImageNet still reaches over 3,000, leaving room for further improvement. Some other approaches incessantly refine an auxiliary model using the query feedback to directly generate AEs (Yin et al. 2023a), approximate decision boundaries (Yin et al. 2023b) or gradients (Du et al. 2020) of victim models, which are time-consuming and resource-intensive.

Feature Disentanglement

A lot of recent endeavors (Lin et al. 2019; Zou et al. 2020; Wang et al. 2020) in computer vision have embraced the utilization of autoencoders (Hinton and Zemel 1993; Li, Pei, and Li 2023) and generative adversarial networks (Goodfellow et al. 2014; Arora et al. 2023) to acquire disentangled representations of image features. In the domain of adversarial attacks and defenses, the disentanglement of features is commonly harnessed for detecting AEs and improving models' robustness, such as (Mustafa et al. 2020; Wang et al. 2021; Yang et al. 2021). To the best of our knowledge, only a limited number of *white-box* attack methods have involved

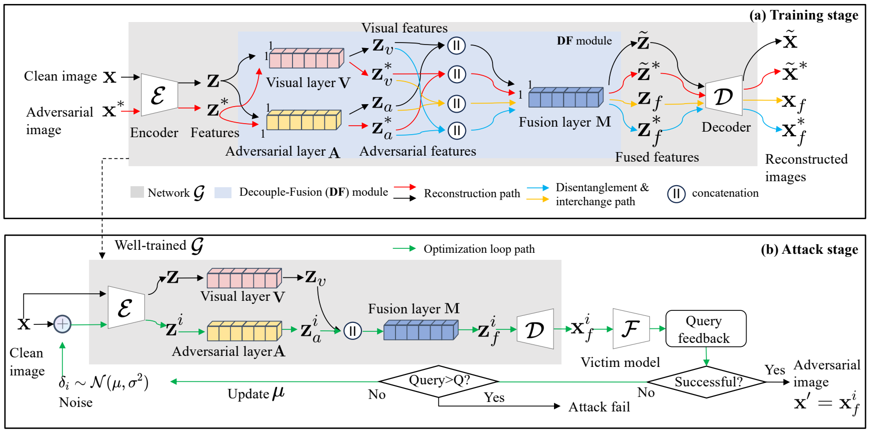

Figure 1: (a) The training procedure of the autoencoder $\mathcal{G}$ equipped with our proposed DF module for disentangling adversarial and visual features. (b) The proposed black-box adversarial attack method, i.e. DifAttack, incorporated with the pre-trained $\mathcal{G}$.

feature disentanglement. RNF (Kim, Lee, and Ro 2021) is a white-box attack approach that optimizes image perturbations by maximizing the gradient of non-robust features in perturbed images, wherein non-robust features are distilled from the target classifier's intermediate features based on information bottleneck (Alemi et al. 2016). Another white-box attack method (Lu et al. 2021) trains an autoencoder, called SSAE, to directly generate target models' AEs. Its training loss aims to minimize the norm difference of output logits between clean images and AEs to achieve better visual consistency meanwhile maximizing the angle discrepancy of output logits for mis-classification. We conjecture that the logits of the deep penultimate layer overfit to the specific model architecture, which makes it lack strong adversarial transferability in a black-box attack scenario. Compared to SSAE and RNF, our DifAttack decouples features related to the adversarial capability and visual perception from latent features used for image reconstructions, not from the intermediate features of specific classifiers, which is probably more suitable for black-box adversarial attacks.

Proposed DifAttack

Method Overview

As illustrated in Figure 1, our approach unfolds in two phases. In the first *training stage*, we train an autoencoder $\mathcal{G}$ for feature disentanglement and image reconstruction, by using pairs of clean image $\mathbf{x}$ along with its AE $\mathbf{x}^*$ generated by performing untargeted white-box attacks on surrogate models. Taking the clean image $\mathbf{x}$ as an example, $\mathcal{G}$ can disentangle the latent feature $\mathbf{z}$ of $\mathbf{x}$ into the adversarial feature $\mathbf{z}_a$ and visual feature $\mathbf{z}_v$ via our proposed DF module, and then reconstructs another image $\tilde{\mathbf{x}}$ similar to $\mathbf{x}$ by decoding the fused $\mathbf{z}_a$ and $\mathbf{z}_v$. In the second *attack stage*, we leverage the pre-trained $\mathcal{G}$ to generate an AE $\mathbf{x}'$ of $\mathbf{x}$ to attack the black-box model $\mathcal{F}$. This $\mathbf{x}'$ is obtained through an iterative optimization process, wherein we continuously sample a batch of perturbed images to extract adversarial features $\mathbf{z}_a^i$ by updating the sampling parameters according to the query feedback, while keeping the visual feature $\mathbf{z}_v$ consistent with that of the clean image, resulting in batches of reconstructed images $\mathbf{x}_f^i$ to query $\mathcal{F}$, until either the query number reaches its maximum threshold Q or one of the perturbed images $\mathbf{x}_f^i$ successfully deceives $\mathcal{F}$, thereby yielding the AE $\mathbf{x}' = \mathbf{x}_f^i$.

Training Stage: Train the Autoencoder $\mathcal{G}$

Suppose that there is a clean image $\mathbf{x}$ in the training dataset and we have pre-trained several white-box surrogate models on this dataset, forming a set $\mathbb{C}$. Note that the availability of these white-box surrogate models has also been assumed in the existing approaches (Lord, Mueller, and Bertinetto 2022; Feng et al. 2022). We adopt the white-box attack method PGD (Madry et al. 2018) to generate an AE $\mathbf{x}^*$ of $\mathbf{x}$ against a surrogate model $\mathcal{C}_j$ selected randomly from $\mathbb{C}$. PGD can be replaced by other white-box attack methods or a collection of them, in which case the performance is presented in the supplementary file. We aim to train an autoencoder $\mathcal{G}$ based on pairs of clean image $\mathbf{x}$ and its AE $\mathbf{x}^*$, for accomplishing two tasks: the feature disentanglement and image reconstruction. Next, we delve into the details of each task.

Disentanglement. For discovering an image's adversarial feature to which the image's adversarial capability is sensitive, meanwhile ensuring that alterations in this feature maintain minimal impact on the image's visual appearance, we disentangle adversarial and visual features from the latent feature used for image reconstruction. For realizing such a disentanglement, we exchange the adversarial and visual features between a pair of clean image and its AE, expecting images reconstructed from these exchanged features to exhibit different adversarial capabilities and visual appearance according to the original images their features stem from.

Specifically, for implementing the feature disentanglement, as illustrated in Figure 1(a), we design a DF module to not only extract but also fuse the adversarial and visual features. The DF module consists of the adversarial layer $\mathbf{A}$, visual layer $\mathbf{V}$ and fusion layer $\mathbf{M}$, which are all realized with a series of stacked 1×1 convolution layers. Taking the clean image $\mathbf{x}$ as an example, DF feeds its latent feature

$\mathbf{z} = \mathcal{E}(\mathbf{x})$, where $\mathcal{E}$ denotes a Convolutional Neural Network (CNN)-based encoder, into $\mathbf{A}$ and $\mathbf{V}$, then outputs a new feature $\tilde{\mathbf{z}}$ from $\mathbf{M}$. The layers $\mathbf{A}$ and $\mathbf{V}$ employ distinct weights to activate $\mathbf{z}$ into the adversarial feature $\mathbf{z}_a$ and the visual feature $\mathbf{z}_v$, respectively. The $\tilde{\mathbf{z}}$ is capable of reconstructing $\tilde{\mathbf{x}} \approx \mathbf{x}$ through the CNN-based decoder $\mathcal{D}$. We formulate the functionality realized by the DF module as:

$$\mathrm{DF}(\mathbf{z}, \mathbf{z}) = \mathbf{M}(\mathbf{A}(\mathbf{z}) || \mathbf{V}(\mathbf{z})). \tag{1}$$

Here, $||$ denotes the channel-wise concatenation. Notably, DF does not satisfy the commutative property, i.e. $\mathrm{DF}(\mathbf{z}^*, \mathbf{z}) \neq \mathrm{DF}(\mathbf{z}, \mathbf{z}^*)$, where $\mathbf{z}^* = \mathcal{E}(\mathbf{x}^*)$, unless $\mathbf{z} = \mathbf{z}^*$.

Then we utilize the DF module to conduct feature interchange for accomplishing disentanglement. Specifically, as depicted by the yellow path in Figure 1(a), the adversarial feature of the clean sample, i.e. $\mathbf{z}_a$, is firstly concatenated with the visual feature of the adversarial image, i.e. $\mathbf{z}_v^*$. Afterward, they are inputted into the fusion layer to obtain the fused feature $\mathbf{z}_f$, thereafter decoded into a new image $\mathbf{x}_f$. Symmetrically, as shown by the blue path, we can reconstruct another image $\mathbf{x}_f^*$ from $\mathbf{z}_a^* = \mathbf{A}(\mathbf{z}^*)$ and $\mathbf{z}_v = \mathbf{V}(\mathbf{z})$. The generation of $\mathbf{x}_f$ and $\mathbf{x}_f^*$ can be formulated as:

$$\mathbf{x}_f = \mathcal{D}(\mathrm{DF}(\mathbf{z}, \mathbf{z}^*)),\ \mathbf{x}_f^* = \mathcal{D}(\mathrm{DF}(\mathbf{z}^*, \mathbf{z})). \tag{2}$$

Subsequently, since we desire the adversarial feature to control the adversarial capability of the image, while the visual feature to govern its visual perception, $\mathbf{x}_f$ should be closer to the visual perception of the AE $\mathbf{x}^*$ and be non-adversarial. Conversely, $\mathbf{x}_f^*$ is expected to be closer to the visual perception of the clean image $\mathbf{x}$ and be adversarial against all surrogate models so as to improve the generalization ability of the adversarial feature on different classifiers. Hence, we aim to minimize the disentanglement loss L_{dis} defined as:

$$\begin{aligned} L_{dis} &= ||\mathbf{x}^* - \mathbf{x}_f||_2 + \frac{1}{N}\sum\nolimits_{j=1}^{N} L_{adv}(\mathbf{x}_f, y, 1, \mathcal{C}_j, k) \\ &+ ||\mathbf{x} - \mathbf{x}_f^*||_2 + \frac{1}{N}\sum\nolimits_{j=1}^{N} L_{adv}(\mathbf{x}_f^*, y, 0, \mathcal{C}_j, k), \end{aligned} \tag{3}$$

where N signifies the cardinality of the surrogate model set $\mathbb{C}$. The l_2 norm measures the Euclidean distance of two images, and L_{adv} is employed to evaluate the adversarial ability, whose definition is (Carlini and Wagner 2017):

$$\begin{aligned} &L_{adv}(\mathbf{x}, y, v, \mathcal{C}_j, k) \\ &= \max\left\{\mathbb{I}(v) \cdot \left(\mathcal{C}_j(\mathbf{x}, y) - \max_{d \neq y} \mathcal{C}_j(\mathbf{x}, d)\right), -k\right\}. \end{aligned} \tag{4}$$

The value of v can be set as 0 or 1. $\mathbb{I}(0) = 1$ means an untargeted attack with a ground-truth label of y, while $\mathbb{I}(1) = -1$ indicates a targeted attack with a target category of y. $\mathcal{C}_j(\mathbf{x}, d)$ is the output score of $\mathcal{C}_j$ at the d-th class. The adjustable constant $k > 0$ controls the adversarial extent.

Reconstruction. For achieving a cycle consistency (Zhu et al. 2017) with regard to feature disentanglement, as depicted in the reconstruction paths of Figure 1(a), the autoencoder is expected to reconstruct an input image well even when its latent feature is processed by the DF module. Given that the image decoupled by $\mathcal{G}$ could be either clean or adversarial, we define the reconstruction loss L_{rec} of $\mathcal{G}$ by:

$$L_{rec} = ||\mathbf{x} - \mathcal{D}\big(\mathrm{DF}(\mathbf{z}, \mathbf{z})\big)||_2 + ||\mathbf{x}^* - \mathcal{D}\big(\mathrm{DF}(\mathbf{z}^*, \mathbf{z}^*)\big)||_2. \tag{5}$$

Algorithm 1: The proposed DifAttack method.

Input: target classifier $\mathcal{F}$; clean image $\mathbf{x}$; max query budget $Q > 0$; distortion budget ϵ; ground-truth label y; learning rate η; variance σ; v, k in Eq.(8); sample scale τ.
Output: Adversarial image $\mathbf{x}'$.

```
Let $q \leftarrow 0, \mathbf{x}' \leftarrow \mathbf{x}, \mathbf{z} \leftarrow \mathcal{E}(\mathbf{x}), \mu \sim \mathcal{N}(0, I)$
while $q + \tau \leq Q$ do
    $\gamma_1, ..., \gamma_i, ..., \gamma_\tau \sim \mathcal{N}(0, I)$
    $\delta_i \leftarrow \mu + \sigma\gamma_i\,, \forall i \in \{1, ..., \tau\}$
    $\mathbf{x}_f^i \leftarrow \Pi_{\epsilon,\mathbf{x}}(\mathcal{D}(\mathrm{DF}(\mathcal{E}(\mathbf{x} + \delta_i), \mathbf{z})))\,, \forall i$
    $l_i \leftarrow L_{adv}(\mathbf{x}_f^i, y, v, \mathcal{F}, k)\,, \forall i$
    $q \leftarrow q + \tau$
    if $\exists\, l_i = -k$ then
        $\mathbf{x}' \leftarrow \mathbf{x}_f^i$
        break
    else
        $\hat{l}_i \leftarrow (l_i - \mathrm{mean}(\{l_i\}))/\mathrm{std}(\{l_i\})\,, \forall i$
        $\mu \leftarrow \mu - \frac{\eta}{\tau\sigma}\sum_{i=1}^{\tau} \hat{l}_i\gamma_i$
    end if
end while
return $\mathbf{x}'$
```

Eventually, the loss function to be minimized during training the network $\mathcal{G}$ can be computed as:

$$L_{all} = \lambda \cdot L_{rec} + L_{dis}, \tag{6}$$

where the hyper-parameter λ balances two loss terms.

Attack Stage: Generate the AE $\mathbf{x}'$

Upon obtaining a well-trained network $\mathcal{G}$, in the subsequent attack stage, we can utilize it to generate AEs in a black-box setting. As shown in Figure 1(b), we maintain the visual feature for decoding the perturbed image $\mathbf{x}_f^i$ to be identical to that of the clean image, i.e. $\mathbf{z}_v$. This ensures that $\mathbf{x}_f^i$ remains in close proximity to the clean image. Meanwhile, our objective is to identify an adversarial feature that, when combined with $\mathbf{z}_v$, has the ability to deceive the victim model $\mathcal{F}$. Therefore, we aim to discover a perturbation δ whose corresponding perturbed image $\mathbf{x} + \delta$ can be used to extract such an adversarial feature. Then the final adversarial image $\mathbf{x}'$ can be reconstructed as follows:

$$\mathbf{x}' = \Pi_{\epsilon,\mathbf{x}}\Big(\mathcal{D}\Big(\mathrm{DF}\big(\mathcal{E}(\mathbf{x} + \delta), \mathcal{E}(\mathbf{x})\big)\Big)\Big), \tag{7}$$

where the operation $\Pi_{\epsilon,\mathbf{x}}$ represents the projection of the input image into the l_p ball bounded by ϵ and centered at $\mathbf{x}$. This forces the adversarial perturbation to remain within the specified constraint. For simplicity, we recast the right-hand side of Eq.(7) as $\mathbf{T}(\mathbf{x} + \delta)$. To find the AE $\mathbf{x}'$, by combining Eq.(7) and using the transformation of variable approach, we formulate our attack objective in a unified manner as:

$$\begin{aligned} &\min\nolimits_{\mu} \mathbb{E}_{\mathcal{N}(\delta|\mu,\sigma^2)} L_{adv}(\mathbf{T}(\mathbf{x} + \delta), y, v, \mathcal{F}, k), \\ &\quad s.t.\ ||\mathbf{T}(\mathbf{x} + \delta) - \mathbf{x}|| \leq \epsilon, \end{aligned} \tag{8}$$

where $\mathbf{x}' = \mathbf{T}(\mathbf{x} + \delta)$ satisfies $\arg\max_{\mathrm{d}} \mathcal{F}(\mathbf{x}', d) \neq y$ for untargeted attacks or $\arg\max_{\mathrm{d}} \mathcal{F}(\mathbf{x}', d) = t$ for targeted attacks. Here, t denotes the target class. We sample δ from

Datasets	ImageNet								Cifar-10							
Victim Models	ResNet		VGG		GoogleNet		SqueezeNet		PyramidNet		DenseNet		VGG		ResNet	
Methods	ASR	Avg.Q	ASR	Avg.Q	ASR	Avg.Q	ASR	Avg.Q	ASR	Avg.Q	ASR	Avg.Q	ASR	Avg.Q	ASR	Avg.Q
Signhunter	49.8	7,935	60.8	7,606	28.1	9,021	45.7	8,315	**100.0**	1,193	**100.0**	1,364	99.6	1,792	99.9	1,137
NES	54.3	7,988	47.7	8,328	37.7	8,748	45.7	8,235	96.3	1,742	98.0	1,757	85.8	3,300	96.1	1,745
$\mathcal{N}$Attack	94.5	5,324	93.5	5,924	76.9	6,883	90.0	5,644	**100.0**	1,072	**100.0**	1,185	**100.0**	1,983	**100.0**	1,168
SimBA	97.0	3,873	97.5	3,505	77.5	5,874	94.0	4,230	**100.0**	812	**100.0**	888	99.4	1,182	**100.0**	824
BASES	35.3	7,170	26.7	8,229	12.5	8,901	15.4	9,228	38.5	6,412	29.5	7,433	37.5	6,682	44.5	5,990
P-RGF	51.3	6,393	58.3	6,093	34.2	7,790	65.3	5,433	94.2	1,297	97.7	742	89.6	1,372	91.6	1,546
Subspace	66.3	4,750	68.3	4,964	50.3	6,642	33.2	8,063	88.8	3,035	87.0	2,490	85.1	2,816	80.1	3,603
GFCS	95.0	3,274	93.0	3,358	60.0	6,359	72.5	6,127	96.9	1,045	99.3	635	93.9	1,505	97.1	979
CGA	91.6	4,222	91.1	4,801	90.9	4,645	93.2	3,972	98.5	1,759	99.4	1,953	91.0	2,673	99.0	1,309
DifAttack(**Ours**)	**100.0**	**1,901**	**100.0**	**2,246**	**97.0**	**3,201**	**98.0**	**2,793**	**100.0**	**414**	**100.0**	**498**	**100.0**	**736**	**100.0**	**455**

Table 1: The attack success rate (ASR %), average number of queries (Avg.Q) of test images in targeted attacks with the target class 864 for ImageNet and the target class 0 for Cifar-10. The best and second best Avg.Q are **bold** and underlined, respectively.

a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$ where the mean μ is to be optimized and the variance σ is fixed to be the optimal value through performing a grid search. For optimizing μ, we adopt the natural evolution strategies method to estimate the gradient of the expectation of L_{adv} in Eq.(8) w.r.t μ. When using stochastic gradient descent with a learning rate η and a batch size τ, the update rule for μ is as follows:

$$\mu \leftarrow \mu - \frac{\eta}{\tau\sigma} \sum\nolimits_{i=1}^{\tau} \gamma_i L_{adv}(\mathbf{T}(\mathbf{x} + \mu + \sigma\gamma_i), y, v, \mathcal{F}, k). \quad (9)$$

Here, $\gamma_i \sim \mathcal{N}(0, I)$, and by setting $\delta_i = \mu + \sigma\gamma_i$, we obtain $\delta_i \sim \mathcal{N}(\mu, \sigma^2)$.

Finally, our proposed DifAttack is summarized in Algorithm 1. Note that, in line 12, we normalize the L_{adv} of each perturbed image to stabilize convergence by taking inspiration from the $\mathcal{N}$Attack (Li et al. 2019) method.

Experiments

In this section, we compare our DifAttack with SOTA score-based black-box attack methods in both the close-set and open-set scenarios. Ablation experiments on the DF module and τ in Eq.(9) are also conducted. Further experimental results are presented in the supplementary file.

Experiment Setup

Datasets. We mainly conduct experiments on the large-scale ImageNet-1k, small-scale Cifar-10 and Cifar-100 datasets. The images in ImageNet are cropped and resized to 224×224, while images in Cifar-10 and Cifar-100 datasets remain a size of 32×32. When conducting attacks on Cifar-100 in open-set scenarios, we exclude 10 classes that have potential semantic overlap with Cifar-10. The specific classes are presented in the supplementary file.

Classifiers. We collect four pre-trained classifiers with different architectures for each of the above three datasets. In the close-set scenario, when we attack one of the four models, the other three are adopted as surrogate models. When attacking the model of Cifar-100 in the open-set scenario, the surrogate models consist of three classifiers trained on Cifar-10 with different architectures from the victim model. For ImageNet, the four models are ResNet-18, VGG-16, GoogleNet, and SqueezeNet provided by torchvision[2]. For Cifar-10, ResNet-Preact-100, VGG-15, DenseNet-BC-110 and PyramidNet-110 (Feng et al. 2022) are selected. For Cifar-100, its four classifiers are chosen from ResNet-56, VGG-16, DenseNet-121 and GoogleNet (Weiai 2023).

Victim Models	ResNet		VGG		GoogleNet		SqueezeNet	
Methods	ASR	Avg.Q	ASR	Avg.Q	ASR	Avg.Q	ASR	Avg.Q
Signhunter	99.5	931	**100.0**	702	95.5	1,504	**100.0**	363
NES	97.5	1,264	98.0	1,068	**100.0**	1,234	91.0	1,711
$\mathcal{N}$Attck	**100.0**	780	**100.0**	752	**100.0**	878	**100.0**	563
SimBA	99.5	614	99.0	533	**100.0**	865	99.5	428
BASES	86.4	1,549	95.5	564	81.7	2,080	85.1	1,717
P-RGF	96.0	711	98.0	529	93.0	1,162	95.0	796
Subspace	94.5	917	95.5	867	95.5	994	97.0	566
GFCS	**100.0**	739	**100.0**	515	96.5	1,348	98.5	658
CGA	97.3	475	99.4	137	**100.0**	**139**	99.3	202
DifAttack(**Ours**)	**100.0**	**156**	**100.0**	**123**	**100.0**	165	**100.0**	**90**

Table 2: The ASR (%), Avg.Q of test images from ImageNet in an untargeted attack setting.

Comparative Methods. We compare our DifAttack with SOTA approaches, including Signhunter (Al-Dujaili and O'Reilly 2020), NES (Ilyas et al. 2018), $\mathcal{N}$Attack (Li et al. 2019), SimBA-DCT (shorted as SimBA) (Guo et al. 2019), BASES (Cai et al. 2022), P-RGF (Cheng et al. 2019), Subspace (Guo, Yan, and Zhang 2019), GFCS (Lord, Mueller, and Bertinetto 2022) and CGA (Feng et al. 2022). We evaluate them according to their official source codes. Due to the absence of pre-trained models and code, the experimental results of the CGA method on ImageNet and the real-world Imagga API (Imagga 2023) are cited from its published paper, while its results on Cifar-10 and Cifar-100 are derived from the pre-trained models provided by the authors.

Parameters. Following many previous works, we also set the maximum perturbation on ImageNet to 12/255, and on CIFAR-10 as well as CIFAR-100 to 8/255. The maximum query number for these three datasets is set to 10,000. For the real-world Imagga API, the maximum query number is limited to 500 due to their query limit. In our DifAttack, we

[2] https://pytorch.org/vision/stable/index.html

Settings	Untargeted								Targeted							
Victim Models	GoogleNet		DenseNet		ResNet		VGG		GoogleNet		DenseNet		ResNet		VGG	
Methods	ASR	Avg.Q	ASR	Avg.Q	ASR	Avg.Q	ASR	Avg.Q	ASR	Avg.Q	ASR	Avg.Q	ASR	Avg.Q	ASR	Avg.Q
Signhunter	99.0	519	99.5	386	**100.0**	121	97.5	704	89.5	3,586	92.0	3,336	96.5	2,076	81.5	4,013
NES	92.5	1,442	94.0	1,454	96.0	914	88.0	1,961	48.5	6,724	53.0	6,488	76.5	4,396	38.5	7,642
$\mathcal{N}$Attack	**100.0**	553	**100.0**	528	**100.0**	311	97.0	952	73.0	6,101	82.0	5,646	95.0	2,853	76.5	5,235
Simba	**100.0**	327	99.5	389	98.0	360	**98.5**	605	79.5	4,203	78.5	4,189	96.0	2,401	77.5	4,156
CGA	96.5	571	99.5	279	**100.0**	59	96.5	691	70.1	5,098	75.6	4,781	90.0	3,189	63.2	5,586
DifAttack(**Ours**)	**100.0**	**146**	**100.0**	**122**	**100.0**	**39**	**98.5**	**386**	**91.0**	**2,506**	**94.0**	**2,740**	**97.0**	**1,382**	**86.0**	**2,980**

[1] Methods including BASES, P-RGF, Subspace, and GFCS, which rely on the output scores of surrogate models to update perturbed images, are unable to perform attacks in this open-set scenario. The reason is that the categories outputted by these models do not encompass the classes of the test images from Cifar-100.

Table 3: Untargeted and targeted black-box attacks in the open-set scenario. The test images are sourced from CIFAR-100, and the victim model is trained on CIFAR-100. Meanwhile, the autoencoder and surrogate models are trained on CIFAR-10.

Settings	Metrics	DifAttack	CGA	GFCS	SimBA	$\mathcal{N}$Attack	Signhunter
Untargeted	ASR	**90**	85	65	45	65	70
	Avg.Q	**90.8**	139.2	221.2	287.6	207.2	187.2
Targeted	ASR	**70**	-	-	35	30	60
	Avg.Q	**183.5**	-	-	338.3	417.8	189.5

[1] In untargeted attacks, we aim to remove the top three predicted classes of clean images. In targeted attacks, we set the target class to be the 2nd-highest scoring class among the API's predicted classes.

Table 4: The ASR and Avg.Q when attacking Imagga API.

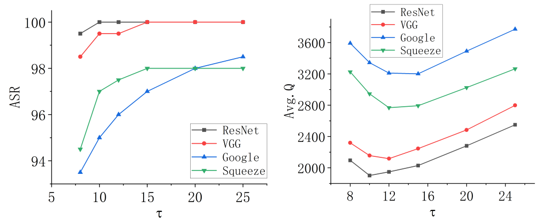

Figure 2: The influence of the parameter τ on the ASR and Avg.Q when performing targeted attacks on ImageNet.

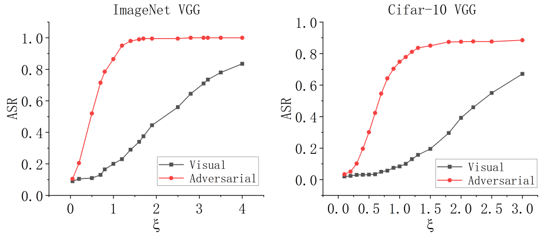

Figure 3: The sensitivity comparison of the ASR between the adversarial and visual features.

set $\lambda = 1$, $\sigma = 0.1$, $\eta = 0.01$, $k = 5$ or 0 for Eq.(3) and (9).

The Close-set Scenario

Black-box Attack on ImageNet and Cifar-10. Table 1 presents the ASR and average query numbers (Avg.Q) in challenging targeted attacks on ImageNet and Cifar-10. We randomly select a target class and 1,000 images, excluding those belonging to the selected class, as the test set. It is observed that our method achieves a 100% ASR in most cases, which is much better than other approaches when images come from ImageNet. Additionally, our DifAttack significantly reduces Avg.Q when obtaining the same or even higher ASR compared to other competitors. For instance, when attacking ResNet on ImageNet, the Avg.Q of DifAttack is reduced by 41.9% compared to the second-lowest query numbers obtained by GFCS (from 3,274 to 1,901), meanwhile achieving an additional 5% increase in ASR. In apart from this, for Cifar-10, DifAttack also reduces the Avg.Q by 43.3% compared to the second-lowest queries obtained by SimBA (from 812 to 414) when against PyramidNet. The experimental results of untargeted attacks on ImageNet are presented in Table 2, where DifAttack maintains a 100% ASR in all cases and achieves lower query numbers in most cases. Given the inherent simplicity of the untargeted attack task, the Avg.Q of all competitors remains relatively low; but DifAttack is also almost the best. As presented earlier, in more challenging targeted attacks, DifAttack significantly outperforms other methods. Further experimental results for untargeted attacks on Cifar-10 and targeted attacks with other target classes can be found in supplementary file.

The Open-set Scenario

Black-box Attack on Cifar-100. We use the autoencoder and surrogate models pre-trained on Cifar-10 to generate AEs for images from Cifar-100, then evaluate the performance of these AEs when attacking victim models pre-trained on Cifar-100. The experimental results in Table 3 demonstrate that in both untargeted and targeted attacks, DifAttack can always achieve fewer query numbers and the same or higher ASR simultaneously, compared to other methods. For example, when against GoogleNet in an untargeted setting, DifAttack reduces the minimum Avg.Q for other methods from 327 to 146; in targeted attacks against VGG and GoogleNet, our approach decreases the lowest Avg.Q obtained by other methods by over 1,000 while maintaining a higher ASR. These experimental results in such an open-set scenario illustrate the favorable generalizability of our disentangled adversarial feature on unseen datasets.

Black-box Attack against Real-world API. To evaluate the attack performance in a more practical open-set scenario, we assess comparative methods by attacking the real-world Imagga API (Imagga 2023). This API utilizes a recognition model trained on a diverse dataset of daily images spanning over 3,000 unknown categories. The Imagga provides multiple predicted classes and corresponding scores for each query image. Due to the API's call limits, we choose the top-performing five methods in the above experiments as the comparative methods, which are CGA, GFCS, SimBA, $\mathcal{N}$Attack and Signhunter. We use the autoencoder and surrogate models pre-trained on ImageNet to extract disentangled features of test images. The experimental results in Table 4 indicate that even when the actual training dataset is completely unknown, our DifAttack exhibits superior query efficiency and ASR, e.g., an Avg.Q of 90.8 for the untargeted attack, and 90%/70% ASR in untargeted/targeted attacks. We also observe that Signhunter, which has the second best targeted query efficiency in Table 3, achieves an Avg.Q comparable to ours in the targeted attack against Imagga, yet suffering an ASR that is 10% lower than ours. Moreover, in the untargeted attack against Imagga and closed-set scenarios presented in Table 1-2, Signhunter largely lags behind our method in both ASR and Avg.Q. Notably, GFCS successfully generates AEs in untargeted attacks but not in targeted attacks, since GFCS is unable to craft AEs when the target class does not occur in the output of surrogate models. Apart from this, the missing result of CGA in the targeted attack is due to the absence of code and this result in their work.

Black-box Attack against Defensive Methods

It is also important to evaluate the performance of attack methods against defensive strategies. We compare the ASR of comparative methods in untargeted attacks when encountering several SOTA defensive techniques, including Adversarially Training (AT) and pre-processing based methods, such as Bit-depth Reduction (Bit-Red) (Xu, Evans, and Qi 2018), Neural Representation Purifier (NRP) (Naseer et al. 2020) and Diffusion Purifier (DiffPure) (Nie et al. 2022). For attacking AT models pre-trained on ImageNet, we adopt adversarially trained ResNet-50 as the surrogate model and the more powerful classifier Wide-ResNet-50-2 as the victim model (Salman et al. 2020). For other defensive techniques, we use them to process AEs from ImageNet generated for attacking standardly trained ResNet-18, and then evaluate the ASR of these AEs after being processed. The experimental results shown in Table 5 showcase that the average ASR of DifAttack surpasses that of other methods, standing as the sole approach with an average ASR exceeding 40%. Additionally, across each defensive strategy, DifAttack achieves 2 second and 1 third places. The good performance indicates that DifAttack can effectively disentangle the adversarial feature from the AT model. To be consistent with previous works (Cheng et al. 2019; Feng et al. 2022), and due to the poor performance, the evaluation of targeted attacks under defense is temporarily omitted.

Ablation Studies

DF Module. To assess the DF module, we train an autoencoder $\mathcal{G}$ w/o DF with the same training mode and architecture as $\mathcal{G}$ w/ DF. Additionally, the disentanglement of adversarial and visual features implemented by DF is realized by randomly dividing the feature $\mathbf{z}$ into two halves. Afterward, we examine the image reconstruction loss and ASR of these two network variants. The experimental results presented in Table 6 demonstrate that the usage of the DF significantly reduces the l_2 reconstruction loss from an average value of 50 to 10, meanwhile improving the attack efficiency. For instance, in the case of ResNet, the ASR increases by 16%, accompanied by a reduction of over 2,000 queries.

Methods	AT	BitRed	NRP	DiffPure	Avg. ASR
Signhunter	36.7	38.0	28.9	12.6	26.5
NES	18.6	21.0	31.3	20.0	24.1
$\mathcal{N}$Attack	39.2	45.0	34.9	24.0	34.6
SimBA	39.5	14.5	25.8	13.1	17.8
BASES	0.0	**58.5**	28.5	8.6	31.9
P-RGF	43.2	38.0	37.5	**30.4**	35.3
Subspace	**54.8**	54.0	29.5	14.8	32.8
GFCS	6.8	21.5	32.9	22.5	25.6
CGA	-	-	-	-	-
DifAttack(**Ours**)	52.5	46.5	**41.5**	28.5	**42.3**

Table 5: The ASR of AEs from ImageNet against various SOTA defensive techniques in the untargeted setting.

Settings	l_2	ASR	Avg.Q
w/o DF	57/48/42/48	84/96/92/100	2,324/1,807/1,779/930
w/ DF	**10/9/9/14**	**100/100/100/100**	**156/123/179/90**

Table 6: The ablation study on the DF module when performing untargeted attacks against ResNet/VGG/-GoogleNet/SqueezeNet in ImageNet.

Sampling Scale τ. Another factor in DifAttack that significantly affects the attack performance is the sampling scale τ in Eq.(9). We evaluate the ASR and Avg.Q when changing the value of τ in targeted attacks, as depicted in Figure 2. Generally, as the value of τ increases, the ASR continuously improves, while the Avg.Q initially decreases and then increases. Based on the trend of Figure 2, in targeted attacks, we generally set τ to the optimal value of most classifiers, which is 12. In untargeted attacks, τ is usually set to 8, of which the details are provided in the supplementary file.

Disentangled Features. To evaluate the effect of the disentangled adversarial and visual features on the adversarial capability, we randomly sample a noise from a Gaussian distribution $\mathcal{N}(0, \xi^2)$ to perturb either the adversarial or visual feature, while keeping the other feature unchanged. Then we employ our pre-trained autoencoder to decode an image from the fusion of these two features. The resulting ASR of these decoded images is presented in Figure 3. With the increasing ξ, the adversarial feature exhibits a faster growth in ASR compared to the visual feature. This manifestation underscores the high sensitivity of an image's adversarial capabilities to modifications in its adversarial features.

Conclusion

This work presents a novel score-based black-box attack method named DifAttack, which aims to decouple the adversarial and visual features from an image's latent feature and perform black-box attacks via the disentangled feature space. The main idea is to optimize the adversarial feature while keeping the visual feature unchanged until a successful AE is obtained. Experimental results demonstrate the superior attack efficiency of DifAttack in both close-set and open-set scenarios, as well as against a real-world API.

References

Al-Dujaili, A.; and O'Reilly, U.-M. 2020. Sign Bits are All You Need for Black-Box Attacks. In *Proc. Int. Conf. Learn. Representat.*

Alemi, A. A.; Fischer, I.; Dillon, J. V.; and Murphy, K. 2016. Deep Variational Information Bottleneck. In *Proc. Int. Conf. Learn. Representat.*

Arora, J.; Tushir, M.; Kherwa, P.; and Rathee, S. 2023. Generative Adversarial Networks: A Comprehensive Review. *Data Wrangling: Concepts, Applications and Tools*, 213.

Brendel, W.; Rauber, J.; and Bethge, M. 2018. Decision-Based Adversarial Attacks: Reliable Attacks Against Black-Box Machine Learning Models. In *Proc. Int. Conf. Learn. Representat.*

Cai, Z.; Song, C.; Krishnamurthy, S.; Roy-Chowdhury, A.; and Asif, S. 2022. Blackbox Attacks via Surrogate Ensemble Search. In *Proc. Int. Conf. Neural Inf. Process. Sys.*, volume 35, 5348–5362.

Carlini, N.; and Wagner, D. 2017. Towards Evaluating the Robustness of Neural Networks. In *Proc. IEEE Conf. Symp. Security Privacy.*, 39–57.

Cheng, S.; Dong, Y.; Pang, T.; Su, H.; and Zhu, J. 2019. Improving Black-Box Adversarial Attacks with a Transfer-Based Prior. In *Proc. Int. Conf. Neural Inf. Process. Sys.*, volume 32.

Croce, F.; and Hein, M. 2020a. Minimally Distorted Adversarial Examples with a Fast Adaptive Boundary Attack. In *Proc. Int. Conf. Mach. Learn.*, 2196–2205. PMLR.

Croce, F.; and Hein, M. 2020b. Reliable Evaluation of Adversarial Robustness with an Ensemble of Diverse Parameter-Free Attacks. In *Proc. Int. Conf. Mach. Learn.*, 2206–2216.

Dong, Y.; Fu, Q.-A.; Yang, X.; Pang, T.; Su, H.; Xiao, Z.; and Zhu, J. 2020. Benchmarking Adversarial Robustness on Image Classification. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 321–331.

Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; and Li, J. 2018. Boosting Adversarial Attacks with Momentum. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 9185–9193.

Du, J.; Zhang, H.; Zhou, J. T.; Yang, Y.; and Feng, J. 2020. Query-efficient Meta Attack to Deep Neural Networks. In *Proc. Int. Conf. Learn. Representat.*

Feng, Y.; Wu, B.; Fan, Y.; Liu, L.; Li, Z.; and Xia, S.-T. 2022. Boosting Black-Box Attack with Partially Transferred Conditional Adversarial Distribution. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 15095–15104.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative Adversarial Nets. In *Proc. Int. Conf. Neural Inf. Process. Sys.*, volume 27.

Google. 2023. Google Cloud Vision API. https://cloud.google.com/vision. Accessed:2023-08-15.

Guo, C.; Gardner, J.; You, Y.; Wilson, A. G.; and Weinberger, K. 2019. Simple Black-Box Adversarial Attacks. In *Proc. Int. Conf. Mach. Learn.*, 2484–2493.

Guo, C.; Rana, M.; Cisse, M.; and van der Maaten, L. 2018. Countering Adversarial Images using Input Transformations. In *Proc. Int. Conf. Learn. Representat.*

Guo, Y.; Yan, Z.; and Zhang, C. 2019. Subspace Attack: Exploiting Promising Subspaces for Query-Efficient Black-Box Attacks. In *Proc. Int. Conf. Neural Inf. Process. Sys.*, volume 32.

Hinton, G. E.; and Zemel, R. 1993. Autoencoders, Minimum Description Length and Helmholtz Free Energy. In *Proc. Int. Conf. Neural Inf. Process. Sys.*, volume 6.

Huang, Z.; and Zhang, T. 2020. Black-Box Adversarial Attack with Transferable Model-based Embedding. In *Proc. Int. Conf. Learn. Representat.*

Ilyas, A.; Engstrom, L.; Athalye, A.; and Lin, J. 2018. Black-box Adversarial Attacks with Limited Queries and Information. In *Proc. Int. Conf. Mach. Learn.*, 2137–2146.

Imagga. 2023. AI-Powered Image Tagging API. "https://imagga.com/solutions/auto-tagging". Accessed:2023-08-15.

Kim, J.; Lee, B.-K.; and Ro, Y. M. 2021. Distilling Robust and Non-Robust Features in Adversarial Examples by Information Bottleneck. In *Proc. Int. Conf. Neural Inf. Process. Sys.*, volume 34, 17148–17159.

Li, P.; Pei, Y.; and Li, J. 2023. A Comprehensive Survey on Design and Application of Autoencoder in Deep Learning. *Applied Soft Computing*, 110176.

Li, Y.; Li, L.; Wang, L.; Zhang, T.; and Gong, B. 2019. Nattack: Learning the Distributions of Adversarial Examples for an Improved Black-Box Attack on Deep Neural Networks. In *Proc. Int. Conf. Mach. Learn.*, 3866–3876.

Lin, J.; Chen, Z.; Xia, Y.; Liu, S.; Qin, T.; and Luo, J. 2019. Exploring Explicit Domain Supervision for Latent Space Disentanglement in Unpaired Image-to-Image Translation. *IEEE Trans. on Pattern Anal. and Mach. Intell.*, 43(4): 1254–1266.

Lord, N. A.; Mueller, R.; and Bertinetto, L. 2022. Attacking Deep Networks with Surrogate-Based Adversarial Black-Box Methods is Easy. In *Proc. Int. Conf. Learn. Representat.*

Lu, S.; Xian, Y.; Yan, K.; Hu, Y.; Sun, X.; Guo, X.; Huang, F.; and Zheng, W.-S. 2021. Discriminator-Free Generative Adversarial Attack. In *Proc. ACM Int. Conf. Multimedia.*, 1544–1552.

Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; and Vladu, A. 2018. Towards Deep Learning Models Resistant to Adversarial Attacks. In *Proc. Int. Conf. Learn. Representat.*

Mohaghegh Dolatabadi, H.; Erfani, S.; and Leckie, C. 2020. Advflow: Inconspicuous Black-Box Adversarial Attacks using Normalizing Flows. In *Proc. Int. Conf. Neural Inf. Process. Sys.*, 15871–15884.

Mustafa, A.; Khan, S. H.; Hayat, M.; Goecke, R.; Shen, J.; and Shao, L. 2020. Deeply Supervised Discriminative Learning for Adversarial Defense. *IEEE Trans. on Pattern Anal. and Mach. Intell.*, 43(9): 3154–3166.

Naseer, M.; Khan, S.; Hayat, M.; Khan, F. S.; and Porikli, F. 2020. A Self-Supervised Approach for Adversarial Robustness. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 262–271.

Nie, W.; Guo, B.; Huang, Y.; Xiao, C.; Vahdat, A.; and Anandkumar, A. 2022. Diffusion Models for Adversarial Purification. In *Proc. Int. Conf. Mach. Learn.*

Qin, Y.; Xiong, Y.; Yi, J.; and Hsieh, C.-J. 2023. Training Meta-Surrogate Model for Transferable Adversarial Attack. In *Proc. of the AAAI Conf. on Artif. Intell.*, volume 37, 9516–9524.

Salman, H.; Ilyas, A.; Engstrom, L.; Kapoor, A.; and Madry, A. 2020. Do Adversarially Robust ImageNet Models Transfer Better? arXiv:2007.08489.

Vaze, S.; Han, K.; Vedaldi, A.; and Zisserman, A. 2021. Open-Set Recognition: A Good Closed-Set Classifier is All You Need. In *Proc. Int. Conf. Learn. Representat.*

Wang, G.; Han, H.; Shan, S.; and Chen, X. 2020. Cross-domain Face Presentation Attack Detection via Multi-Domain Disentangled Representation Learning. In *Proc. IEEE Conf. Comput. Vis. Pattern Recogn.*, 6678–6687.

Wang, S.; Nepal, S.; Rudolph, C.; Grobler, M.; Chen, S.; Chen, T.; and An, Z. 2021. Defending Adversarial Attacks via Semantic Feature Manipulation. *IEEE Trans. on Serv. Comput.*, 15(6): 3184–3197.

Weiai, C. 2023. Pytorch-cifar100. https://github.com/weiaicunzai/pytorch-cifar100. Accessed:2023-08-15.

Wierstra, D.; Schaul, T.; Glasmachers, T.; Sun, Y.; Peters, J.; and Schmidhuber, J. 2014. Natural Evolution Strategies. *The Journal of Mach. Learn. Resear.*, 15(1): 949–980.

Xu, W.; Evans, D.; and Qi, Y. 2018. Feature Squeezing: Detecting Adversarial Examples in Deep Neural Networks. In *Proc. Netw. Distrib. Syst. Secur. Symp.*

Yang, S.; Guo, T.; Wang, Y.; and Xu, C. 2021. Adversarial Robustness through Disentangled Representations. In *Proc. of the AAAI Conf. on Artif. Intell.*, volume 35, 3145–3153.

Yin, F.; Zhang, Y.; Wu, B.; Feng, Y.; Zhang, J.; Fan, Y.; and Yang, Y. 2023a. Generalizable Black-Box Adversarial Attack With Meta Learning. *IEEE Trans. on Pattern Anal. and Mach. Intell.*, 1–13.

Yin, F.; Zhang, Y.; Wu, B.; Feng, Y.; Zhang, J.; Fan, Y.; and Yang, Y. 2023b. Generalizable Black-Box Adversarial Attack with Meta Learning. *IEEE Trans. on Pattern Anal. and Mach. Intell.*

Zhang, H.; Avrithis, Y.; Furon, T.; and Amsaleg, L. 2021. Walking on the Edge: Fast, Low-Distortion Adversarial Examples. *IEEE Trans. Inf. Forensics and Security*, 16: 701–713.

Zhu, J.-Y.; Park, T.; Isola, P.; and Efros, A. A. 2017. Unpaired Image-to-Image Translation using Cycle-Consistent Adversarial Networks. In *Proc. IEEE Conf. Comput. Vis.*

Zou, Y.; Yang, X.; Yu, Z.; Kumar, B. V.; and Kautz, J. 2020. Joint Disentangling and Adaptation for Cross-Domain Person Re-Identification. In *Euro. Conf. on Comput. Vis.*, 87–104.