Effectively Improving Data Diversity of Substitute Training for Data-Free Black-Box Attack

Yang Wei, Zhuo Ma, *Member, IEEE*, Zhuoran Ma, Zhan Qin, *Member, IEEE*, Yang Liu, Bin Xiao, Xiuli Bi, and Jianfeng Ma, *Member, IEEE*

***Abstract*—Recent substitute training methods have utilized the concept of Generative Adversarial Networks (GANs) to implement data-free black-box attacks. Specifically, in designing the generators, the substitute training methods use a similar structure to the generators in GANs. However, this design approach ignores the potential situation that the generators in GANs operate under real data supervision, while the generators in substitute training methods lack such supervision. This difference in data-supervised conditions constrain the diversity of data generated by the substitute training methods, resulting in inadequate data to support effective training of the substitute model. This impacts the substitute model's ability to attack the target model further. Consequently, to solve the above issues, we propose three strategies to improve the attack success rates. For the generator, we first propose a dense projection space that projects the input noise into various latent feature spaces to diversify feature information. Then, we introduce a novel disguised natural color mode. This mode improves information exchange between the generator's output layer and previous layers, allowing for more diverse generated data. Besides, we present a regularization method for the substitute model, called noise-based balanced learning, to prevent the potential risk of overfitting due to the lack of diversity of the generated data. In the experimental analysis, extensive experiments are conducted to validate the effectiveness of these proposed strategies.**

***Index Terms*—Data-free black-box attack, substitute training, data diversity, overfitting risk.**

Manuscript received 7 December 2022; revised 20 July 2023; accepted 21 December 2023. Date of publication 28 December 2023; date of current version 11 July 2024. This work was supported in part by the National Natural Science Foundation of China under Grant U21A20464, Grant 61872283, and Grant 62172067, in part by the Natural Science Basic Research Program of Shaanxi under Grant 2021JC-22, in part by the Key Research and Development Program of Shaanxi under Grant 2022GY-029, in part by the China 111Project, and in part by the Natural Science Foundation of Chongqing for Distinguished Young Scholars under Grant CSTB2022NSCQ-JQX0001. *(Corresponding authors: Zhuo Ma; Zhuoran Ma.)*

Yang Wei is with the School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: ywei9395@gmail.com).

Zhuo Ma, Zhuoran Ma, Yang Liu, and Jianfeng Ma are with the School of Cyber Engineering, Xidian University, Xi'an 710071, China (e-mail: mazhuo@mail.xidian.edu.cn; emmazhr@163.com; bcds2018@foxmail.com; jfma@mail.xidian.edu.cn).

Zhan Qin is with the School of Cyber Science and Technology, College of Compute Science and Technology, Zhejiang University, Hangzhou 310027, China, and also with the ZJU-Hangzhou Global Scientific and Technological Innovation Center, Hangzhou 311200, China (e-mail: qinzhan@zju.edu.cn).

Bin Xiao and Xiuli Bi are with the Department of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (e-mail: xiaobin@cqupt.edu.cn; bixl@cqupt.edu.cn).

Digital Object Identifier 10.1109/TDSC.2023.3347753

I. Introduction

DEEP Neural Networks (DNNs) are prominent in numerous domains, including computer vision, speech recognition, language processing, and medical diagnosis, due to their exceptional performance. The development of DNNs requires considerable time and effort to collect massive annotated data samples. Due to the General Data Protection Regulation (GDPR) [1] and the protection of DNN's value, sharing training data, model architecture, and model parameters are not permitted. DNNs as a black-box approach are typically deployed in commercial applications, i.e., inputs in, predictions out.

Unfortunately, the widespread deployment of DNNs creates incentives for adversaries. It has been demonstrated that DNNs are susceptible to adversarial examples [2], [3], [4], [5], [6], [7]. The perturbations in an adversarial example from a benign one are imperceptible, but can lead DNNs to output incorrect predictions [8]. Adversarial attacks utilize the approximations and imperfections in DNN algorithms to control the outputs of DNNs [9]. For instance, in a hand-written digits recognition task, an adversary can prevent a classification model from correctly recognizing a digital image by introducing an imperceptible perturbation. After applying an adversarial example patch to vehicles on the road, the object detection algorithm in the autonomous driving system becomes blind, thus incurring traffic accidents. Therefore, the adversarial attack is dreadful for those domains that applied DNNs, which may cause serious consequences. Based on the properties of adversarial attack methods, they can be divided into two categories, i.e., white-box attack [10], [11] and black-box attack [12], [13], [14], [15], [16], [17], [18]. The white-box attacks are permitted to access any information about DNNs, e.g., the structure and parameters of model. In contrast, the black-box attacks only obtain the output information of model, which is more practical for real-world applications. Substitute training [19], [20], [21], [22], [23] is a typical transferability-based black-box attack that requires many data labeled by the attacked model to train a substitute model. However, data is unobtainable for most attack cases. Therefore, combining the substitution training and the data-free manner yields a more practical method known as data-free substitute training.

The data-free substitute training approaches [9], [24] are a potential way for black-box adversarial attack. It can attack the target DNN model by using only the output information of the target model without knowing the structure or training data of the target model. Even with a small amount of knowledge, the data-free substitute training methods can successfully attack

the target DNN model, which is sufficient to demonstrate their promise. Nevertheless, the limitation of the data-free substitute training approaches is evident that they necessitate a significant volume of queries to obtain a well-trained substitute model. The substitute training consists of three models, including the target model, the generator for generated data, and the substitute model for adversarial examples. In training phase, these generated data are fed into the target model to derive predictions (i.e., labels or class probabilities), and are utilized to train the substitute model to minimize the decision boundary distance between the substitute and target models. In this way, the substitute model are capable of replacing the target model. The data-free substitute training is promising, which is also the focus of this paper.

However, the existingdata-free substitute training methods [24], [25], [26], [27] also have two limitations that will impair the attack performance of the substitute model. One of the priorities to be addressed is the lack of diversity of the generated data. The issue stems from these methods overlooking the different training conditions between themselves and GANs. Specifically, GANs have the supervision of real data, while data-free substitute training lacks such supervision. Additionally, there is the problem of insufficient diversity in the generated data, which already exists in GANs [28]. These methods tend to design the structures of their generators based on GANs' generators as templates, without addressing these differences. Due to the lack of diversity in these generated data, the effective training of the substitute model is not guaranteed which further affects success rates in attacking the target model. Here, we show the two issues that have the largest impact on data diversity in the structure of the generator as follows: i) the majority of generators contain only one projection layer, which causes random noise from Gaussian distribution being projected into the same feature space, which is not conducive to generating features with sufficient diversity; ii) at the output layer of the generator, the number of feature extractors has to corresponds to that of channels of the natural images. Given that the natural image is unknown, this method of determining the number of extractors is plainly illogical. It will greatly limit the diversity of the generated data, hence it is unsuitable for the data-free substitute training. Anther limitation is potential overfitting risk. Under the training setting with insufficient data diversity, the substitute model is incapable of learning a wider data distribution and is prone to overfitting issue. This weakened substitute model cannot be used to generate strong adversarial examples, hence reducing the attack success rate against the target model.

Motivated by the above challenges, we propose three generic strategies to increase the success rates of adversarial attacks. We first decompose a conventional generator into two parts: projection and mapping blocks. Then, the structures of the two blocks are refined to boost the diversity of generated data. Besides, for the substitute model, we propose a regularization method to minimize the overfitting risk so that enhancing its attack ability. The main contributions of the paper can be summarized as follows:

i) We analyze the reason for the lack of diversity of feature information in the projection block. We propose a hybrid space, termed density projection space, which consists of multiple projection layers with random direction transformation and can project random noise into various feature spaces. Compared to a single projection layer, density projection space increases the diversity of feature information vastly.
ii) Without knowledge of the natural image, the mapping block of the generator in the substitute training directly referencing that of the GAN will vastly lose the diversity of the generated data. Therefore, we propose an adversarial color mode to enhance the data diversity by optimizing the structure of the original mapping block. Since the adversarial color mode increases the number of channels of these generated data, we also propose a channel compression operation to lower the number of channels in order to deceive the target model.
iii) To prevent the overfitting risk caused by the lack of diversity in the generated data, we propose a novel regularization method, called Noise-based Balanced Learning (NbBL). NbBL strengthens the robustness of the substitute model by injecting noise properties, hence boosting its attack capability.
iv) On various datasets, we evaluate the proposed three strategies from adversarial attack results, ablation study and the analysis of data diversity. The results of adversarial attack experiments and ablation study show that our strategies are practical to improve the attack ability of diverse data-free substitute training methods. Diversity analysis experiments demonstrate the research core of this study that enhancing the diversity of generated data in substitute training is the key to black-box attacks without any real data.

II. Related Work

DNNs are vulnerable to adversarial examples, which aim to misclassifications during the prediction process [2]. It is noteworthy that adversarial examples are transferable, meaning they can be crafted for one DNN and then be used to attack another DNN with a different architecture [9]. Adversarial attack methods can be divided into two categories, i.e., white-box attacks and black-box attacks. In the white-box setting, the adversary has complete access to the target attack model, which is unfeasible in real-world applications [29], [30], [31], [32], [33], [34], [35]. In the black-box setting, the adversary knows nothing about the target model except the output information. Li et al. [36] proposed a black-box attack approach that finds a probability density distribution in a narrow region centered around the benign inputs for adversarial examples. Since it is necessary to obtain benign inputs, the practicality of the black-box attack method is weakened.

Data-Dependent Substitute Training: Papernot et al. [37] first designed a substitute model based on adversarial example transferability, which mimics the target black-box DNN using real data, and then crafts adversarial examples by attacking the target DNN using the substitute model. Kariyappa et al. [38] presented a scheme for the model stealing attack based on zeroth-order gradient estimation. However, the adversary must obtain a fraction of the training dataset for the target model.

Data-Free Substitute Training: Considering data privacy [1] and inaccessibility of collecting real data, data-free substitute training [24], [25], [27] is one of the fastest-growing black-box attacks. The data-free substitute training approach is based on the adversarial example transferability, which implies adversarial

TABLE I
LIST OF ABBREVIATIONS USED IN THIS PAPER

Abbreviation	Description
GANs	Generative Adversarial Nets
$\mathcal{D}$	discriminator in GANs
$\mathcal{G}_D$	generator in GANs
$\mathcal{G}_S$	generator in data-free substitute training
S	substitue model in data-free substitute training
$\mathcal{T}$	target model in data-free substitute training
DPS	Density Projection Space
AdCM	Adversarial Color Mode
DNCM	Disguised Natural Color Mode
NbBL	Noise-based Balanced Learning

examples that misclassify one model can mislead other models, even if their architectures are vastly different [9], [39], [40]. Consequently, the adversarial attack can be launched on the substitute model and subsequently transmitted to the target model. Zhou et al. [24] proposed a data-free black-box attack method without learning any information about training data and the target model. Zhang et al. [41] presented the adversarial perturbation and black-box attack, which alleviates the dependence on the original training samples by exploiting artificial Jigsaw images as the training samples. Wang et al. [25] designed an efficient substitute training for black-box attacks, which uses a diverse data generation module to generate large-scale data with broad distribution to effectively construct the substitute model. Wang et al. [27] introduced a dynamic substitute structure learning strategy that constructs an optimal substitute model structure for different target models and tasks. However, the above schemes exploit the generator of Generative Adversarial Network (GAN) [42], which incurs the lack of diversity in generated data and mode collapsing [43], [44]. Therefore, these schemes seldom learn how to recover the consistent data distribution and decision boundary of the target model.

III. PROBLEM ANALYSIS

For an adversarial substitute attack against the target model $\mathcal{T}$, the key to success is varied data supplied by the generator $\mathcal{G}_S$ in the data-free substitute training in order to train a powerful substitute model S. However, since $\mathcal{G}_S$ is trained without the supervision of natural images, the diversity of data it generates is diminished if its structure totally follows that of $\mathcal{G}_D$ (the generator in GANs). In this section, we analyze the issue and improve $\mathcal{G}_S$. The abbreviations used in this paper are listed in Table I.

A. Diversity of Generators in Data-Free Substitute Training

First, we discuss the differences between Generative Adversarial Nets (GANs) and data-free substitute training, and then we assess the diversity of data generated by $\mathcal{G}_S$ if these differences are neglected.

1) Differences Between GANs and Substitute Training: GANs consist of a generator $\mathcal{G}_D$ and a discriminator $\mathcal{D}$. $\mathcal{G}_D$ is responsible for generating realistic images, whereas $\mathcal{D}$ needs to assess the authenticity of these images to the greatest extent feasible. Based on the concept of GANs, researchers have proposed some data-free substitute training methods to attack target models. A data-free substitute training method contains three roles: a generator $\mathcal{G}_S$, a substitute model S and a target model $\mathcal{T}$. In the training process, $\mathcal{G}_S$ provides a variety of data to adequately support the training of S, enabling S to effectively steal the decision boundary of $\mathcal{T}$. Although the objectives of both $\mathcal{G}_D$ and $\mathcal{G}_S$ are to generate data, there are some differences between the two. We summarize these differences as follows:

i) *Supervision condition:* In the training course, $\mathcal{G}_D$ has to be supervised by natural images in order to understand their data distribution. However, for the data-free substitute training, $\mathcal{G}_S$ generates data without the supervision of natural images.

ii) *Type of the generated images:* The goal of $\mathcal{G}_D$ is to generate images that are as realistic as possible and deceive $\mathcal{D}$. However, $\mathcal{G}_S$ is to generate images with adequate diversity and effectively train S.

Overall, both the conditions and objectives of $\mathcal{G}_D$ and $\mathcal{G}_S$ are different. If $\mathcal{G}_D$'s structure is directly applied to the design of $\mathcal{G}_S$, the diversity of the data generated by $\mathcal{G}_S$ is affected. We will discuss in detail if this affection is negative or positive in the next paragraph.

2) Impact on Diversity: For a training dataset, its diversity consists of two parts: *the inter-class and intra-class diversities*. The inter-class diversity DS_{inter} is the total of the differences between any two different categories at class level, and the intra-class diversity DS_{intra} is the sum of the differences between any two different samples within a class at pixel level.

Since $\mathcal{G}_D$ is trained under the supervision of a training dataset, the diversity of data generated by $\mathcal{G}_D$ stems from the two types of diversities in the dataset. We assume a training dataset has n_r classes and n_a samples in each class. DS_{inter} and DS_{intra} are respectively defined as follows.

$$DS_{inter} = \sum_{i=1}^{n_r}\sum_{j=1}^{n_r} d(c_i, c_j), i \neq j \tag{1}$$

$$DS_{intra} = \sum_{v=1}^{n_a}\sum_{u=1}^{n_a} d(x_v, x_u), v \neq u \tag{2}$$

where $d(*,*)$ is a metric, e.g., cosine similarity, to measure the difference at class or pixel levels. c_i and c_j represent any two different classes in a training dataset. x_v and x_u are any two diverse samples from some class of the dataset.

In contrast, $\mathcal{G}_S$ is trained without the supervision of real samples in a training dataset, resulting in the loss of DS_{intra} of the dataset. It only obtains the class information of the dataset indirectly, via the output of $\mathcal{T}$, therefore its generated data only contain DS_{inter}. Using CIFAR10 dataset [45] as a illustration, $n_r = 10$ and $n_a = 10^4$. Clearly, the loss of the substantial intra-class diversity will impact the diversity of data generated by $\mathcal{G}_S$ negatively.

B. Structural Decomposition of Generators

Under the absence of the intra-class diversity, the diversity of the data generated by $\mathcal{G}_S$ is diminished. Therefore, to increase data diversity, it is necessary that decomposing the structure of $\mathcal{G}_D$ and proposing targeted improvements. We first partition $\mathcal{G}_D$ into two submodules based on their generation processes, and then we analyze their deficiencies. We consider a $\mathcal{G}_D$ to

be composed of two blocks: *projection* and *mapping blocks*. As seen in Fig. 2(a), we use DCGAN [46] generator as an illustration of $\mathcal{G}_D$.

- *Projection block:* A one-dimensional random noise z from Gaussian distribution is fed into the projection block which only has a single Fully Connected (FC) layer. The FC layer project z into the same feature space, and the space will produce similar feature information.
- *Mapping block:* In the mapping block, the feature information from the projection block is mapped to a distribution that is analogous to the data distribution of the natural images. Ultimately, we obtain the images as realistic as possible.

Correspondingly, the two blocks each have their own issues, which is concluded as follows.

- *Issue of projection block:* The single FC layer can only project z to the same feature space. Therefore, when using the projection block to design $\mathcal{G}_S$, the diversity of feature information outputted by single feature space is restricted.
- *Issue of mapping block:* In the mapping block of $\mathcal{G}_D$, the number of the feature extractors of output layer corresponds to that of color channels of natural images. However, these channel-specific details are inaccessible to the data-free substitute training. Under this circumstance, it is rough that the structure of output layer of $\mathcal{G}_D$ is used to design that of $\mathcal{G}_S$ without adapting, hence restricting the diversity of the data generated by $\mathcal{G}_S$.

Overall, if $\mathcal{G}_S$ follows the structures of $\mathcal{G}_D$ directly, the diversities of its projection and mapping blocks are constrained by a single FC layer and color channel information, respectively. Since the color channel information is related to the natural color mode of images, we will describe the color mode in depth in the next subsection.

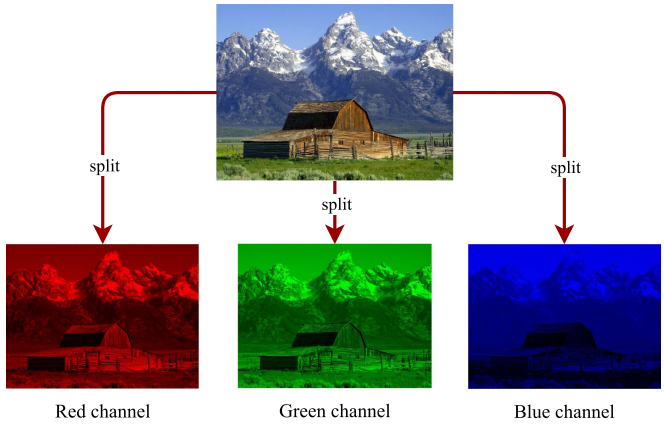


Fig. 1. Three color channels are interrelated, and various colors are generated by mixing them in different proportions.

C. Natural Color Mode

Here, we explain in detail why the natural color mode carried by images determines the structure of the output layer and why the mapping block of $\mathcal{G}_D$ is not suitable to totally transferred into $\mathcal{G}_S$.

In the mapping block of $\mathcal{G}_D$, the output layer plays a crucial role. It receives the feature information from all the preceding layers and generates final images. The output layer consists of only a few several feature extractors (i.e., convolution kernels), whose number is determined by the color mode of the natural images in the generating task, e.g., the number of convolution kernels is 3 when natural images have RGB color mode. Fig. 1 shows a natural image with RGB Color Mode, which indicates the correlation between the channels of RGB. We refer to the color mode carried by these natural images as *natural color mode*. Natural color mode provides the essential properties of an image, including the number of color channels, the distribution of pixels per channel, and the correlation among channels.

Each kernel in the output layer is responsible to learn the pixel distribution of corresponding channel in natural color mode and the correlation among channels. Then, the output layer generates images with natural color mode. However, the way that employs the natural color mode to determine the structure of the output layer is not suited for the data-free substitute training. In contrast, for the data-free substitute training, the kernels in the output layer of $\mathcal{G}_S$ are incapable of learning any information contained by natural color mode. If we transfer the learning way of $\mathcal{G}_D$ to $\mathcal{G}_S$ directly, $\mathcal{G}_S$ will progressively cause the output of the convolution kernels in the output layer to be same. The reason for this is that throughout the training process, $\mathcal{G}_S$ will make the channel information outputted by each kernel as similar as feasible in order to converge fast under unconstrained conditions, resulting in simple data with minimal complexity in the final composition. These simple data enables $\mathcal{G}_S$ to find the local optimal solution faster, but they lack the enough data diversity required to fully train the substitute model.

D. Summary and Motivation

In the previous sections, we have decomposed the generator $\mathcal{G}_D$'s structure into projection and mapping blocks and analyzed the root causes of lacking diversity in these two modules. We summarize these reasons as follows:

i) The existing projection block in $\mathcal{G}_D$ contains only a single fully connected (FC) layer. However, a single FC layer cannot project the input noise into a richer feature space, resulting in limited diversity of feature information.

ii) In the mapping block of $\mathcal{G}_D$, the number of feature extractors in the output layer is predefined to be equal to the channel count of an image's natural color mode. However, the number of channels in a natural color mode is very small, usually 1 or 3. This implies that the corresponding number of feature extractors is also small, directly leading to limited diversity in the generated data.

Under both of these limiting conditions, $\mathcal{G}_S$ based on $\mathcal{G}_D$ obviously cannot generate sufficiently diverse data to effectively support adequate training of $\mathcal{S}$. Therefore, to address these problems, we need to propose targeted improvements to the structure of $\mathcal{G}_S$. The core idea of these improvements is to enhance the diversity of the generated data by modifying $\mathcal{G}_S$'s structure so that $\mathcal{S}$ can produce more powerful adversarial samples under sufficient training, thus increasing the attack success rate for $\mathcal{T}$.

Algorithm 1: Feature Projection in DPS.

Input:
k: the number of FC layers in DPS;
w_i and b_i, $i \in k$: the weight and bias of FC_i;
β_i: alterable direction factor for FC_i sampled from the beta distribution $Beta(\alpha, \alpha)$;
$concat()$: a splice operation;

Output:
F: final feature information;

Sample a random noise vector z with length m;
Divide z into k sub-noise vectors;
for $i = 1$ to k **do**
  Input sub-noise vector z_i into FC_i with β_i, and get $f_i \leftarrow (w_i z_i + b_i) \cdot \beta_i$;
end for
$F \leftarrow concat(x_i)$;
return F;

IV. Methodology

We have proved that the loss of intra-class diversity decreases the diversity of data generated by $\mathcal{G}_S$. To increase the diversity and enhance the attack capability of S for $\mathcal{T}$, we propose two strategies for refining the structure of $\mathcal{G}_S$. Besides, we also present a regularization method to solve the potential overfitting problem caused by the loss of diversity. Note that our proposed scheme is generic and applicable to any data-free substitute training methods.

A. Density Projection Space

Here, we first solve the lack of diversity of generated data caused by the projection block.

In the projection block of $\mathcal{G}_D$, a single FC layer only can project a random noise z sampled from standard Gaussian distribution into a single latent feature space, and this space only outputs a type of feature information. Therefore, increasing the number of FC layers is a logical way to improve the diversity of feature information. However, to converge rapidly, these supplemental layers will be optimized toward the same direction, leading z to be projected into neighboring feature spaces and still restricting the diversity of the feature information. Consequently, to further settle the optimized direction problem of the multiple FC layers, we sample some direction factors from the beta distribution to change the optimization directions of these FC layers. These FC layers with random direction transformation constitute the hybrid space known as Density Projection Space (DPS).

Diverse latent feature spaces in DPS enable z to be projected into various types of feature information. Clearly, compared to a single projection layer, DPS can substantially increase the diversity of generated data by small computation cost. As a replacement module, DPS can replace the projection blocks in most of $\mathcal{G}_S$, as seen in the left dark grey box of Fig. 2(b). DPS outputs richer feature information than the projection block of DCGAN generator. The projection steps of DPS are explained as Algorithm 1, which is described as follows in detail.

i) We sample a random noise vector z with length m from Gaussian distribution.
ii) To project z into different feature spaces, it is partitioned into k sub-noise vectors with length $\frac{m}{k}$. k is the number of classes for the attack task.
iii) To adjust the optimized directions of these FC layers, we sample k variable direction factors from $Beta(\alpha, \alpha)$. In all of our experiments, we set $\alpha = 1$. These sub-noises are inputted into DPS, each layer FC_i with β_i output feature information by computing $f_i \leftarrow (w_i z_i + b_i) \cdot \beta_i, i \in k$.
iv) Along a given dimension, all sub-feature information from k feature spaces is concatenated using $F \leftarrow concat(f_i)$.

The rationale for sampling the direction factors from a beta distribution is as follows. The input noise of DPS follows Gaussian distribution. To differentiate the input and the direction factors, we sample the direction factors from a distribution different from the Gaussian distribution. Here, we sample direction factors from the Beta distribution since it provides continuous values in the range (0, 1) allowing us to randomly adjust the optimization direction of each FC layer. The Beta distribution also has a parameter α that controls the shape, so we can flexibly adjust how direction factors are sampled. We are able to change the FC layers' optimized directions by multiplying the sampled Beta random variable β_i with the output of each layer. This introduces controlled randomness into the optimization direction, leading to more diverse feature spaces.

B. Disguised Natural Color Mode

To improve the diversity of data generated by the mapping block, we propose a Disguised Natural Color Mode (DNCM) more suitable for the data-free substitute training, which consists of an Adversarial Color Mode (AdCM) and a channel compression operation.

1) Adversarial Color Mode: The natural color mode determines the structure of the output layer in the mapping block of $\mathcal{G}_D$, as described in Section III-C. Therefore, a novel AdCM is proposed to refine the output layer so that the reformative output layer could be transferred into $\mathcal{G}_S$.

In the output layer of $\mathcal{G}_D$, the purpose of each convolutional kernel is to output a corresponding color channel information after receiving the feature information from the previous layers. Since the output layer follows the rule that one kernel corresponds to one channel of information, increasing the number of kernel allows it to generate data with richer channel information. Therefore, in order to improve the mapping block of $\mathcal{G}_D$ and make it more suitable for $\mathcal{G}_S$, we expand the number of kernels in the output layer, which improves the diversity of the generated images and supports the adequate training of S. The color mode carried by these generated data is referred to as Adversarial Color Mode (AdCM). AdCM is responsible to enhance data diversity and hence improving the attack success rate of S for $\mathcal{T}$.

2) Channel Compression Operation: $\mathcal{T}$ is trained to learn the data distribution of natural images, thus it only recognizes the images with natural color mode. However, AdCM provides more color channel information than natural color mode. Therefore, to effectively steal the information of $\mathcal{T}$, we disguise the generated

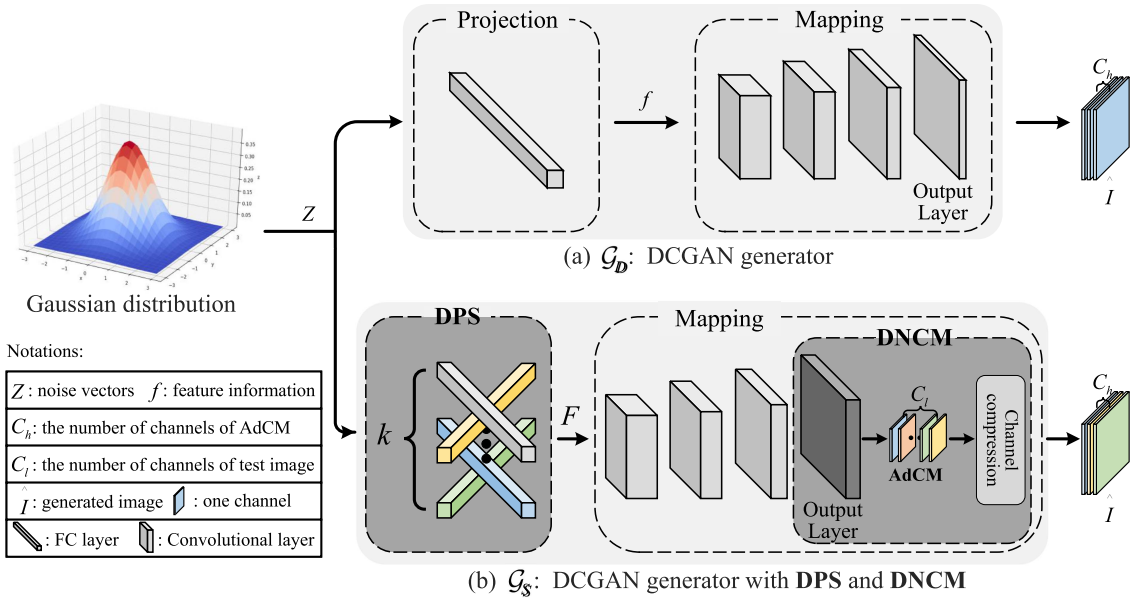


Fig. 2. Diagrams of (a) DCGAN generator ($\mathcal{G}_D$) and (b) DCGAN generator with DPS and DNCM ($\mathcal{G}_S$). The differences between $\mathcal{G}_D$ and $\mathcal{G}_S$ are the projection and mapping structures, which are depicted in the gray regions of the subgraph (b).

images with AdCM as natural color mode images by compressing their channel information. As demonstrated in Algorithm 2, we propose a channel compression operation for the disguised objective. The details of Algorithm 2 are described as follows.

i) Given a generated image I with AdCM, we use a Global Average Pooling (GAP) operation [47] to calculate the global information for each channel of I, respectively.
ii) These channels are arranged from least to greatest based on the magnitude of their global information.
iii) We divide the sorted channels into C_h intervals, e.g., $C_h = 1$ and 3 for Gray and RGB natural color modes, respectively.
iv) To disguise I as a natural image, we need to convert AdCM to natural color mode. We randomly select one channel from each interval to form a disguised natural image $\hat{I}$.

In the right dark grey box of Fig. 2(b), we clearly demonstrate the difference between the mapping block + DNCM and the origin mapping block. This distinction allows for the generation of more diversified data.

Algorithm 2: Channel Compression Operation.

Input:
C_l: the number of channels of AdCM;
C_h: the number of channels of natural color mode;
Output:
$\hat{I}$: disguised natural image;
1: **for** $i = 1$ to C_l **do**
2: Calculate the global information of the ith channel by a GAP operation;
3: **end for**
4: Sort the C_l channels according to the magnitude of their global information;
5: Divide the sorted channels into C_h intervals;
6: Randomly select one channel from each interval to form a disguised natural image $\hat{I}$;
7: **return** $\hat{I}$ containing C_h channels;

C. Noise-Based Balanced Learning

For the data-free substitute training, if the substitute model is at the overfitting risk, it cannot be utilized to create powerful adversarial examples, resulting in a lower rate of success when attacking the target model. Even though we have proposed two improvements to increase the diversity of the generated samples above, we still need to prevent S from overfitting the $\mathcal{G}_S$-generated data. Hence, we propose a novel regularization method: Noise-based Balanced Learning (NbBL).

Adding random noise into training data is a data augmentation technique that can effectively increase the robustness of models and reduce the risk of overfitting in image classification task. However, the augmentation method is unsuitable for the $\mathcal{G}_S$-generated data due to the following factors: i) since $\mathcal{G}_S$ receives random noise as input, the data generation process can also be seen as a denoising process. The technique of introducing noise contradicts the denoising process of $\mathcal{G}_S$; ii) $\mathcal{G}_D$ is susceptible to unstable training, as demonstrated in [28]. For $\mathcal{G}_S$, introducing noise will increase its randomness, which causes the instability will be further exacerbated. However, S is unaffected by the above issues. Therefore, we embed the noise-based thinking into S.

Usually, for stealing the inside information of $\mathcal{T}$, some classification models are used as S, whose effectiveness has been extensively proven, such as VGG [48] and ResNet [49] series. These models consist of multiple convolutional and FC layers. The objective of all layers before the last FC layer is to extract feature information while reducing feature dimensionality, while

the mission of the last layer, also known as the classification layer, is to classify these extracted features. We consider inserting noise-based thinking into the layers before the classification layer to improve the robustness and attack performance of S. Whereas, some models contain vast feature extraction layers, e.g., ResNet-34 has 34 feature extraction layers. Introducing noise to each layer will result in a burdensome and excessive quantity of noise, which further impacts the convergence of S. Therefore, we limit the number of layers to be injected with noise and just introduce noise in the layer P preceding the classification layer. P as a part of S, its robustness is propagated back to all preceding layers during back propagation.This ensures the convergence of S and strengthens its robustness.During the substitute training, there is information interaction between $\mathcal{G}_S$ and S. To avoid introducing additional noise information, we use $\mathcal{G}_S$'s input noise z as one of the input terms of NbBL, as stated follows.

$$\mathcal{L}_{NbBL} = \left(KL\left(f_P \| M\right) + KL\left(z \| M\right)\right) \times \frac{\alpha}{2} \tag{3}$$

$$M = \frac{f_P + z}{2} \tag{4}$$

where f_p is the feature extracted by P. KL is the Kullback-Leibler divergence function [50]. α is a hyperparameter used to adjust the constraint energy of NbBL, which is set as $0 \leq \alpha \leq 5$ in general.

As a regularization method, once NbBL is inserted into the loss function of S, which can effectively constrain the overfitting of S to training data. A more robust S can create more aggressive adversarial examples, which can also attack the target model more effectively.

V. Experiment

A. Experiment Setting

Datasets and Model Structures: We choose some public datasets and classic models to conduct the experiments of image classification. 1) MNIST [51]: The target model is pre-trained on VGG-16 [48], and ResNet-18 [49]. The substitute model is a small network with 3 convolutional layers. 2) CIFAR-10 [45]: The target model is pre-trained on VGG-16, and ResNet-18. The substitute model is VGG-13. 3) CIFAR-100 [45]: The target model is pre-trained on VGG-19 and ResNet50. The substitute model is ResNet-18. 4) Tiny Imagenet [52]: The target model is pre-trained on ResNet-50. The substitute model is ResNet-34.We evaluate the robustness of our proposed strategies on two publicly available datasets (i.e., CIFAR-10-C and CIFAR-100-C [53]). The two corruption datasets are constructed based on CIFAR-10 and CIFAR-100 datasets, and are primarily utilized for evaluating the robustness of deep learning models against image corruption. CIFAR-10-C dataset contains 19 distinct types of image corruptions, including brightness reduction, contrast reduction, color alteration, blurring, scaling up/down, rotation, translation, horizontal flipping, among others. Each type of corruption is characterized by 5 varying intensity levels, resulting in a total of 95 different varieties of corrupted images. CIFAR-100-C dataset is identical to CIFAR-10-C in terms of both corruption types and intensity levels.

Comparison Methods: To assess the efficacy of the proposed strategies, they are incorporated into three data-free substitute training methods: DaST [24], DDG [25] and FE-DaST [26]. For comparison, several black-box attacks requiring real data are selected, e.g., PBBA [19] and Knockoff [20], along with a data-free black-box attack TEDF [54]. Furthermore, substitute training is performed using both the original training data of the target model and ImageNet [52] for training the substitute model.

Defense Methods: We primarily consider three state-of-the-art disruption-based defense methods, i.e., Prediction Poisoning (PP) [55], Reverse Sigmoid (RS) [56] and Adaptive Misinformation (AM) [57]. The three methods disrupt query results in different manners. In PP, the prediction accuracy of the target model decreases as more noise is added to the query results, rendering it an accuracy-constrained defense. In RS, the top-1 prediction of the target model remains unchanged, thus constituting an accuracy-preserving defense. AM combines detection-based and disruption-based defenses. In AM, the defender adaptively provides misleading predictions to the identified out-of-distribution (OOD) queries.

Implementation Details: We use the deep learning library Pytorch as an implementation platform. Adam is used to train substitute models and generators from scratch, and all weights are randomly initialized. The initial learning rates for the generator and substitute model are both 0.0001. The substitute models for 120 epochs on MNIST dataset, 300 epochs on CIFAR-10/100 datasets, and 400 epochs on Tiny-ImageNet dataset, respectively. We set the mini-batch size as 500, and C_l as 12 in AdCM. The hyperparameter α in NbBL is specified as 1, 5 and 2 on MNIST, CIFARS and Tiny-ImageNet, respectively. All models are trained on one NVIDIA GeForce GTX 3090 GPU. During the evaluation phase, we apply PGD [58] as the default white-box attack method to generate transferable adversarial examples over the well-trained substitute model. Besides, we also utilize several classic attack methods for extensive experiments, e.g., FGSM [59], BIM [60] and C&W [61]. In PP, it perturbs the query results based on a hyperparameter ϵ, which denotes the magnitude of perturbation and is set as 1.1. For RS, we set its hyperparameters $\beta = 0.8$ and $\gamma = 0.2$. In AM, hyperparameter τ balances security and accuracy, being set as 0.99.

Evaluation Metrics: We evaluate the methods under two scenarios proposed in DaST, i.e., only getting the output label from the target model and accessing the output probability well. We name these two scenarios as Probability-based and Label-based. The Attack Success Rates (ASRs) is used to evaluate the performance of transfer-based black-box attacks. As in DaST, in the non-targeted attack setting, we only generate adversarial examples on the images classified correctly by the attacked model. For targeted attack, we only generate adversarial examples on the images which are not classified to the specific wrong labels. In order to reduce the effect of random seeds on the experiment, each experiment is repeated three times and the average results are recorded. Furthermore, to assess the diversity of the generated data, we employ two evaluation metrics: information entropy [62] and structural similarity index measurement (SSIM) [63], ranging from 0 to 1. Information entropy measures the complexity of a single image, with higher

TABLE II
COMPARING ASRS RESULTS USING THE PROBABILITY AS THE TARGET MODEL OUTPUT AMONG DIFFERENT ATTACK METHODS ON THE FOUR DATASETS

	Dataset	MNIST		CIFAR-10		CIFAR-100		Tiny-ImageNet	Average
	Target Model	VGG-16	ResNet-18	VGG-16	ResNet-18	VGG-19	ResNet-50	ResNet-50	advance
Non-targeted	Training Data	29.25	34.81	23.15	32.66	14.47	18.33	12.86	—
	ImageNet	34.86	31.39	22.94	34.01	17.26	20.93	21.75	—
	PBBA	50.31	59.77	30.19	33.91	22.34	28.11	26.54	—
	Knockoff	58.38	65.82	31.58	39.40	27.73	29.55	29.99	—
	TEDF	74.56	81.34	67.81	74.25	46.72	54.90	41.28	—
	DaST	50.26	68.35	35.52	51.52	35.94	44.34	35.00	—
	DaST + Ours	63.01 (↑**12.75**)	75.47 (↑**7.13**)	54.27 (↑28.75)	62.41 (↑10.89)	41.70 (↑5.76)	50.31 (↑5.97)	40.90 (↑5.90)	11.02
	DDG	52.63	72.53	35.92	49.31	32.65	39.11	32.81	—
	DDG + Ours	57.72 (↑5.09)	77.01 (↑4.48)	78.73 (↑**42.81**)	64.94 (↑15.63)	44.52 (↑**11.87**)	47.52 (↑8.41)	37.68 (↑4.87)	13.31
	FE-DaST	69.01	70.73	49.61	53.39	34.46	43.60	34.67	—
	FE-DaST + Ours	76.55 (↑10.54)	73.8 (↑3.07)	76.58 (↑26.97)	76.38 (↑**22.99**)	43.54 (↑9.08)	58.32 (↑**14.72**)	42.19 (↑**7.52**)	13.56
Targeted	Training Data	40.27	43.94	10.35	11.22	5.02	8.66	6.17	—
	ImageNet	43.88	41.72	10.28	13.43	5.82	10.39	11.25	—
	PBBA	55.66	49.24	15.38	20.44	6.73	17.22	13.88	—
	Knockoff	52.89	54.27	16.92	19.56	12.83	22.37	15.26	—
	TEDF	60.34	67.29	55.65	57.12	36.42	33.68	37.35	—
	DaST	51.59	51.39	22.65	26.70	33.95	21.30	15.19	—
	DaST + Ours	63.19 (↑11.60)	53.18 (↑1.79)	32.69 (↑10.04)	51.37 (↑**24.67**)	39.82 (↑5.87)	28.2 (↑6.90)	25.53 (↑10.34)	10.17
	DDG	38.15	55.57	31.30	30.93	15.24	15.48	13.23	—
	DDG + Ours	52.53 (↑**14.38**)	59.93 (↑**4.36**)	60.16 (↑**28.86**)	42.97 (↑12.04)	37.84 (↑**22.60**)	23.69 (↑8.21)	32.53 (↑19.30)	15.68
	FE-DaST	53.34	56.62	35.74	31.53	18.08	19.12	19.65	—
	FE-DaST + Ours	65.7 (↑12.36)	58.25 (↑1.63)	46.59 (↑10.85)	42.22 (↑10.69)	29.50 (↑11.42)	36.59 (↑**17.47**)	55.66 (↑**36.01**)	14.35

Notes. We use PGD as the attack method for all substitute training to ensure a fair comparison. The number within each pair of parentheses represents the improvements of our proposed strategies for their embedded methods.

TABLE III
COMPARING ASRS RESULTS USING THE LABEL AS THE TARGET MODEL OUTPUT AMONG DIFFERENT ATTACK METHODS ON THE FOUR DATASETS

	Dataset	MNIST		CIFAR-10		CIFAR-100		Tiny-ImageNet	Average
	Target Model	VGG-16	ResNet-18	VGG-16	ResNet-18	VGG-19	ResNet-50	ResNet-50	advance
Non-targeted	Training Data	20.11	24.50	10.43	13.05	5.01	8.58	7.32	—
	ImageNet	23.77	22.56	12.73	14.11	8.38	11.28	13.29	—
	PBBA	28.18	29.00	13.63	17.66	11.48	16.33	15.37	—
	Knockoff	33.18	37.72	20.74	19.87	16.48	18.31	22.33	—
	TEDF	46.72	62.29	45.85	41.72	32.34	48.25	31.76	—
	DaST	27.72	25.08	19.35	25.3	24.14	32.41	23.5	—
	DaST + Ours	38.80 (↑11.08)	61.00 (↑**35.92**)	54.09 (↑34.74)	30.15 (↑4.85)	28.12 (↑**3.98**)	42.14 (↑9.73)	32.28 (↑8.78)	15.58
	DDG	37.87	49.32	42.32	16.17	25.79	32.94	20.59	—
	DDG + Ours	45.39 (↑7.52)	63.19 (↑13.87)	65.76 (↑23.44)	40.39 (↑24.22)	29.65 (↑3.86)	42.78 (↑9.84)	27.56 (↑6.97)	12.82
	FE-DaST	42.22	56.49	32.54	22.12	25.51	35.74	23.1	—
	FE-DaST + Ours	54.57 (↑**12.35**)	71.56 (↑15.07)	82.00 (↑**49.46**)	63.6 (↑**41.48**)	29.29 (↑3.78)	56.72 (↑**20.98**)	40.52 (↑**17.42**)	22.93
Targeted	Training Data	12.55	10.88	10.24	9.09	3.97	6.44	4.92	—
	ImageNet	14.81	15.70	12.22	9.32	4.82	8.56	7.02	—
	PBBA	19.86	18.53	11.33	10.48	6.91	7.33	8.61	—
	Knockoff	23.74	17.85	12.80	13.91	9.48	9.52	10.65	—
	TEDF	43.39	38.72	41.32	36.79	23.54	31.16	21.86	—
	DaST	14.69	13.77	16.21	13.04	16.42	16.03	7.04	—
	DaST + Ours	24.90 (↑10.21)	15.93 (↑2.16)	43.59 (↑27.38)	31.43 (↑**18.39**)	24.40 (↑7.98)	30.30 (↑14.27)	14.85 (↑7.81)	12.60
	DDG	30.65	27.69	40.40	20.91	15.68	14.69	6.34	—
	DDG + Ours	37.08 (↑6.43)	31.38 (↑**3.69**)	71.06 (↑**30.66**)	28.44 (↑7.53)	26.49 (↑**10.81**)	28.30 (↑13.61)	12.87 (↑6.53)	11.32
	FE-DaST	32.37	22.66	26.98	17.39	15.37	20.23	7.96	—
	FE-DaST + Ours	51.61 (↑**19.24**)	24.97 (↑2.31)	47.77 (↑20.79)	25.65 (↑8.29)	23.29 (↑7.92)	34.95 (↑**14.72**)	28.58 (↑**14.72**)	12.57

entropy values denoting greater levels of image complexity. The formula for computing information entropy is as follows:

$$H = -\sum_{i=1}^{L} p_i \log_2 p_i \tag{5}$$

where L represents the number of gray levels (in an 8-bit grayscale image, L is 256) and p_i is the probability of occurrence of the ith gray level.

SSIM is utilized to evaluate dissimilarities among all the generated data in the same class, where lower values represent higher levels of difference. SSIM is defined as follows:

$$\mathrm{SSIM}(\mathbf{x}, \mathbf{y}) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_2)} \tag{6}$$

where μ_x and μ_y represent the mean deviation of x and y, respectively, while σ_x and σ_y denote their standard deviations. σ_{xy} represents the covariance between x and y. Constant terms C_1, C_2 and C_3 are introduced to avoid the denominator approaching zero or producing unreliable numerical values.

B. Attack Results

Substitute Training With the Real Data: As shown in Tables II and III, we train the substitute model using the training dataset of the target model or ImageNet dataset, and conduct attack experiments. The results indicate that real data may lead to higher accuracy but lower ASRs for the substitute models than the generated data. We consider this is because, despite the fact that the substitute models are trained with the real data, there is no interaction between the substitute and target models during the training process, hindering the substitute model from approaching the decision boundary of the target model.

Comparisons Among the State-of-the-Art Attacks: As shown in Tables II and III, the performance of several black-box

TABLE IV
ACCURACY COMPARISON OF VARIOUS ATTACK METHODS UNDER PROBABILITY-BASED AND LABEL-BASED SCENARIOS

	Dataset	MNIST		CIFAR-10		CIFAR-100		Tiny-ImageNet
	Target Model	VGG-16	ResNet-18	VGG-16	ResNet-18	VGG-19	ResNet-50	ResNet-50
	Training Data	99.56	99.53	93.8	95.5	70.92	78.03	54.89
Probability-based	DaST	43.33	34.21	30.58	29.76	14.71	21.94	7.51
	DaST + Ours	66.89 (↑23.56)	38.35 (↑4.14)	53.02 (↑22.44)	40.88 (↑11.12)	27.82 (↑13.11)	34.38 (↑12.44)	12.15 (↑4.64)
	DDG	53.66	49.51	34.82	32.26	12.25	23.92	7.04
	DDG + Ours	65.82 (↑12.16)	72.98 **(↑23.47)**	64.70 **(↑29.88)**	48.41 **(↑16.15)**	28.16 **(↑15.91)**	34.51 (↑10.59)	12.70 (↑5.66)
	FE-DaST	51.94	54.53	41.82	42.08	17.23	28.12	10.12
	FE-DaST + Ours	81.78 **(↑29.84)**	64.21 (↑9.68)	71.68 (↑29.86)	55.14 (↑13.06)	29.75 (↑12.52)	53.15 **(↑25.03)**	18.52 **(↑8.40)**
Label-based	DaST	25.06	21.00	19.05	23.73	12.95	16.61	5.71
	DaST + Ours	58.92 **(↑36.86)**	31.53 (↑10.53)	48.05 (↑29.00)	49.73 **(↑26.00)**	26.54 **(↑13.59)**	32.42 (↑15.81)	12.10 (↑6.39)
	DDG	53.72	38.53	31.00	25.71	19.12	21.33	6.83
	DDG + Ours	68.52 (↑14.80)	63.24 **(↑24.71)**	67.95 (↑36.95)	45.83 (↑20.12)	26.47 (↑7.35)	36.03 (↑14.70)	12.12 (↑5.29)
	FE-DaST	51.49	42.42	27.1	32.88	17.41	24.12	8.51
	FE-DaST + Ours	79.23 (↑27.74)	63.21 (↑20.79)	65.95 **(↑38.85)**	48.56 (↑15.68)	28.81 (↑11.40)	53.16 **(↑29.04)**	17.88 **(↑9.31)**

TABLE V
COMPARING ASRS RESULTS USING PROBABILITY AS THE TARGET MODEL OUTPUT AMONG DIFFERENT ATTACK METHODS WITH VARIOUS WHITE-BOX ADVERSARIAL EXAMPLE GENERATION METHODS

	Dataset	CIFAR-10				CIFAR-100			
	Attack Method	FGSM	BIM	PGD	C&W	FGSM	BIM	PGD	C&W
Non-targeted	DaST	41.9	40.1	35.5	25.3	27.9	33.6	44.3	26.8
	DaST + Ours	64.0 (↑22.1)	69.4 **(↑29.3)**	54.3 (↑18.8)	46.0 (↑20.7)	30.3 (↑2.4)	39.7 (↑6.1)	50.3 (↑6.0)	33.8 (↑7.0)
	DDG	46.3	53.1	35.9	30.6	26.6	26.9	39.1	26.3
	DDG + Ours	77.3 **(↑31.0)**	81.8 (↑28.7)	78.7 **(↑41.8)**	52.7 (↑22.1)	29.8(↑3.2)	38.2 (↑11.3)	47.5 (↑8.4)	32.1 (↑5.8)
	FE-DaST	55.4	74.0	49.7	33.0	28.4	32.2	43.7	29.4
	FE-DaST + Ours	75.0 (↑19.6)	91.7 (↑17.7)	77.1 (↑27.4)	63.3 **(↑30.3)**	34.4 **(↑6.0)**	53.2 **(↑21.0)**	58.2 **(↑14.5)**	46.1 **(↑16.7)**
Targeted	DaST	8.4	30.4	22.6	13.6	4.8	19.5	21.4	8.8
	DaST + Ours	14.2 (↑5.8)	58.6 (↑28.2)	32.7 (↑10.1)	19.0 (↑5.4)	5.9 **(↑1.1)**	24.8 (↑5.3)	28.1 (↑6.7)	9.0 (↑0.2)
	DDG	10.74	37.9	31.3	15.6	3.6	12.8	15.5	5.3
	DDG + Ours	21.88 **(↑11.14)**	71.5 **(↑33.6)**	60.2 **(↑28.9)**	42.4 **(↑26.8)**	4.5 (↑0.9)	23.8 (↑11.0)	23.7 (↑8.2)	6.0 (↑0.7)
	FE-DaST	12.3	52.4	35.8	23.0	3.6	17.0	18.8	6.4
	FE-DaST + Ours	20.5 (↑8.2)	81.6 (↑29.2)	46.6 (↑10.8)	34.0 (↑11.0)	3.9 (↑0.3)	36.9 **(↑19.9)**	36.0 **(↑17.2)**	8.4 **(↑2.0)**

attack methods is compared, including two attacks requiring real data (PBBA and Knockoff) and a data-free attack (TEDF). The proposed strategies are incorporated into three data-free substitute training methods: DaST, DDG, and FE-DaST. Our strategies can improve ASRs of these data-free methods by up to 35.92%, 49.46%, 20.98% and 36.01% on the four datasets, respectively. And our strategies can also improve their ASRs by at least 1.63%, 4.85%, 3.78% and 4.87%. Obviously, whether under Probability-based and Label-based scenarios, our strategies significantly improve ASRs for the three methods under both targeted and non-targeted attacks. However, on MNIST dataset, the minimum ASRs improvement is only 1.63%, which is smaller than that of the other datasets. We believe that the reason is that the images on MNIST dataset contain only one color channel, which severely limits the capacity of the generator to maximize the diversity of the produced samples.

Besides, under Probability-based and Label-based scenarios, we calculate the average improvements of our strategies on the four datasets for the three data-free attacks, which are displayed in the rightmost column of Tables II and III. Our strategies improve ASRs by total of 78.09% under Probability-based scenario, but by 87.82% under Label-based scenario. This demonstrates that our strategies can enhance ASRs under Label-based scenario more effectively than under under Probability-based scenario.

The results prove that that enhancing the diversity of the generated samples and the generalization of the substitute models can bring the substitute models closer to the decision boundary of the target model and generate more powerful transferable adversarial examples.

Accuracy Comparisons Between Substitute and Target Models: The prerequisite for transferable adversarial examples to successfully attack the target model is that the substitute model can fit the decision boundary of the target model. Consequently, the test accuracies of the substitute and the target models can measure the similarity of the decision boundaries of the two to some extent. As seen in Table IV, for the data-free substitute training methods, our proposed strategies can improve their test accuracy significantly. On CIFAR-10 dataset, our strategies improve the accuracy of VGG-16 by up to 38.85% under Label-based scenario. Meanwhile, we discover that the improvement gets higher when the architectures of the substitute and target models are similar, e.g., the substitute model VGG-13 being superior than ResNet-18 for the target model VGG-16 on CIFAR-10 dataset.

Comparisons Among Various White-Box Adversarial Example Generation Methods: We embed the proposed three strategies into the three data-free substitute training methods (DaST, DDG and FE-DaST). We use four attacks, including FGSM, BIM, PGD and C&W, to generate adversarial examples over substitute models of six methods for attacking the target models on CIFAR-10/100 datasets. VGG-16 and ResNet-50 are used as the target models on CIFAR-10/100 datasets, respectively. Then, we compare ASRs of several methods under Probability-based scenario, as given in Table V. In the most cases, our strategies can significantly improve ASRs of these data-free training methods, However, on CIFAR-100 dataset, our strategies can only slightly

TABLE VI
ASRs IMPROVEMENTS OF OUR STRATEGIES FOR ATTACKING THE MICROSOFT AZURE EXAMPLE MODEL

	Scenario	Probability-based	Label-based
Non-targeted	DaST	88.28	89.33
	DaST + Ours	92.29 (↑**4.01**)	91.98 (↑2.65)
	DDG	99.56	97.08
	DDG + Ours	99.87 (↑0.31)	98.39 (↑1.31)
	FE-DaST	97.84	91.41
	FE-DaST + Ours	99.64 (↑1.80)	98.47 (↑**7.06**)
Targeted	DaST	44.66	64.25
	DaST + Ours	48.29 (↑3.63)	70.90 (↑6.65)
	DDG	70.21	87.39
	DDG + Ours	75.98 (↑5.77)	89.14 (↑1.75)
	FE-DaST	55.80	57.01
	FE-DaST + Ours	70.90 (↑**15.10**)	84.32 (↑**27.31**)

Notes. PGD is used as the attack method for all substitute training.

TABLE VII
ASRs IMPROVEMENT OF EACH PROPOSED COMPONENT FOR VARIOUS DATA-FREE SUBSTITUTE TRAINING METHODS

	Scenario	Probability-based		Label-based	
	Dataset	CIFAR-10	CIFAR-100	CIFAR-10	CIFAR-100
Non-targeted	DaST	35.5	44.3	19.4	32.4
	+ DPS	41.3 (↑5.8)	45.6 (↑1.3)	27.8 (↑**8.4**)	35.5 (↑3.1)
	+ DNCM	49.4 (↑8.1)	48.8 (↑3.2)	49.0 (↑**21.2**)	39.6 (↑4.1)
	+ NbBL	54.3 (↑4.9)	50.3 (↑1.5)	54.1 (↑**5.1**)	42.1 (↑2.5)
	DDG	35.9	39.1	42.3	32.9
	+ DPS	48.2 (↑**12.3**)	42.4 (↑3.3)	51.8 (↑9.5)	36.0 (↑3.1)
	+ DNCM	70.3 (↑**22.1**)	46.5 (↑4.1)	60.4 (↑8.6)	42.3 (↑6.3)
	+ NbBL	78.7 (↑**8.4**)	47.5 (↑1.0)	65.8 (↑5.4)	42.8 (↑0.5)
	FE-DaST	49.6	43.6	32.5	35.7
	+ DPS	57.2 (↑7.6)	50.3 (↑6.7)	47.7 (↑**15.2**)	45.1 (↑9.4)
	+ DNCM	72.3 (↑15.1)	57.8 (↑7.5)	74.0 (↑**26.3**)	53.5 (↑8.4)
	+ NbBL	76.6 (↑4.3)	58.3 (↑0.5)	82.0 (↑**8.0**)	56.7 (↑3.2)
Targeted	DaST	22.7	21.3	16.2	16.0
	+ DPS	25.0 (↑2.3)	22.2 (↑0.9)	22.5 (↑**6.3**)	18.3 (↑2.3)
	+ DNCM	31.8 (↑6.8)	26.9 (↑4.7)	38.9 (↑**16.4**)	28.4 (↑10.1)
	+ NbBL	32.7 (↑0.9)	28.2 (↑1.3)	43.6 (↑**4.7**)	30.3 (↑1.9)
	DDG	31.3	15.5	40.4	14.7
	+ DPS	42.4 (↑11.1)	19.0 (↑3.5)	52.5 (↑**12.1**)	17.4 (↑2.7)
	+ DNCM	56.7 (↑14.3)	23.1 (↑4.1)	67.8 (↑**15.3**)	25.7 (↑8.3)
	+ NbBL	60.2 (↑**3.5**)	23.7 (↑0.6)	71.1 (↑3.3)	28.3 (↑2.6)
	FE-DaST	35.7	19.1	27.0	20.2
	+ DPS	39.3 (↑3.6)	26.6 (↑**7.5**)	33.1 (↑6.1)	26.8 (↑6.6)
	+ DNCM	45.5 (↑6.2)	34.7 (↑8.1)	44.4 (↑**11.3**)	32.4 (↑5.6)
	+ NbBL	46.6 (↑1.1)	36.6 (↑1.9)	47.8 (↑**3.4**)	35.0 (↑2.6)

Notes. PGD is used as the attack method for all substitute training. Our proposed strategies contains the following components: Density Projection Space (DPS), Disguised Natural Color Mode (DNCM) and Noise-based Balanced Learning (NbBL).

improve ASRs of these methods under FGSM and C&W with targeted attacks. For FE-DaST, its ASRs is 3.6% under FGSM with targeted attack, and our strategies can only improve it by 0.3%. Therefore, we deduce that ASRs of the substitute training method will influence the degree it is enhanced to some extent.

ASRs Comparisons on Microsoft Azure: To evaluate the attack ability of our proposed strategies under the real-world applications, we conduct experiments for attacking the online model on Microsoft Azure in two scenarios. The example MNIST model of the machine learning tutorial on Azure is used as the target model, whose architecture and parameters are unknowable. Access is restricted to this model's output information alone. As shown in Table VI, these methods use same substitute model with five convolutional layers. Our strategies can improve their ASRs under Probability-based and Label-based scenarios. On non-targeted attacks, since ASRs of these substitute attacks is above 88%, the improvements of our strategies are limited, which reaches up to 7.06% for FE-DaST. By comparison, our strategies enhance ASRs of FE-DaST by up to 27.31% on targeted attacks.

C. Ablation Study

We conduct extensive ablative experiments to validate the performance of each component of our strategies. The target models are VGG-16 and ResNet-50 on CIFAR-10/100 datasets, respectively. We apply three components (DPS, DNCM and NbNL) to DaST, DDG and FE-DaST. Noting that DaST already contains multiple projected feature spaces, we only add random direction transformation to its projection block. We demonstrate the improvement of each component in Table VII. Our proposed two components effectively improve DaST's ASRs under both Probability-based and Label-based scenarios. DPS, DNCM and NbBL enhance its ASRs by up to 8.4%, 21.2% and 5.1%, respectively. Our strategies also improves ASRs for DDG and FE-DaST. DPS, DNCM and NbBL can boost their ASRs by up to 15.2%, 26.3% and 8.4%, respectively. Overall, DNCM has more improvement for different attacks than DPS, and DPS has more improvement than NbBL.

It is of paramount importance to note that even though the projection block of DaST and DPS are capable of projecting noise into different feature spaces, they still exhibit some distinct differences. The projection block of DaST encompass multiple branches, with each branch being a complex network comprising multiple layers, including convolutional layers and BN layers, resulting in substantial computational complexity and time consumption. With an increase in the number of branches, the computational burden of DaST also increases. In contrast, DPS utilizes multiple FC layers for noise projection, creating a more streamlined structure. The structure mitigates an excessive computational load while ensuring enhanced diversity in the generated data.

D. Diversity Analysis

The diversity of data generated by $\mathcal{G}_S$ is a crucial element for S to successfully attack $\mathcal{T}$. We conduct a series of experiments to prove that our proposed strategies can effectively improve the diversity of the generated data. Since NbBL is responsible to solve the overfitting issue for S, these experiments are only for DPS and DNCM.

Classes of Generated Data: CIFAR-10/100 datasets contain 10 and 100 classes, respectively. There are 200 classes on Tiny-ImageNet dataset. Our strategies are embedded into three data-free substitute training methods (i.e, DaST, FE-DaST

TABLE VIII
COMPARISONS OF UNGENERATED CLASSES ON DIFFERENT DATASETS

Datasets	Attacks	Number of ungenerated classes	Number of total classes
CIFAR-10	DaST	0	10
	DaST + Ours	0	10
CIFAR-100	FE-DaST	1	100
	FE-DaST + Ours	0	100
Tiny-ImageNet	DDG	27	200
	DDG + Ours	3	200

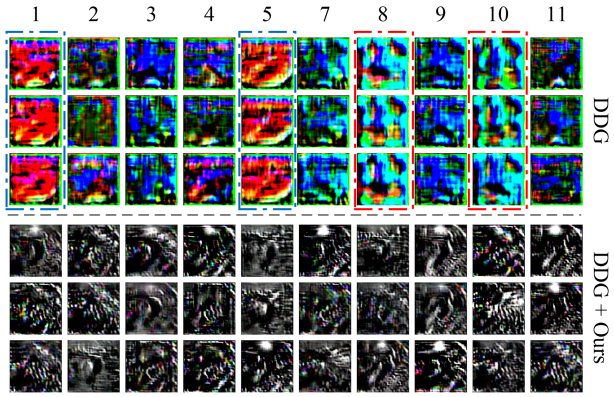


Fig. 3. Diversity comparison between DDG and DDG + Ours. In DDG, the dotted boxes with the same color represent the classes that have similar color and texture information.

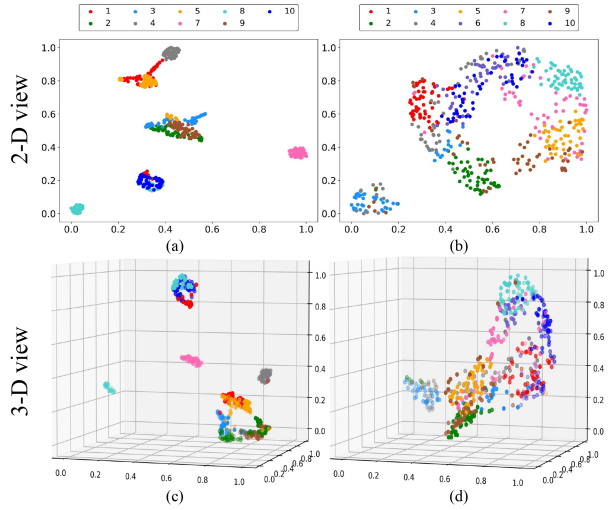
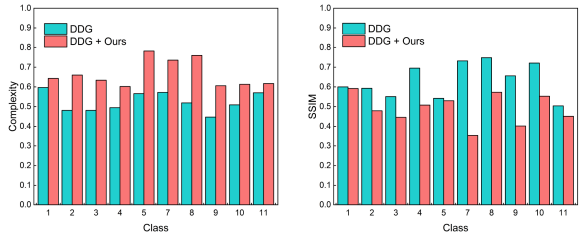


Fig. 4. Using umap [64] to reduce the dimension of generated data for first ten classes on Tiny-ImageNet dataset. One color represents one data class. (a) and (c) show the feature distributions of data generated by DDG from 2-D and 3-D views, respectively. (b) and (d) demonstrate that of DDG + Ours.

and DDG) to generate data, and then we use target models (i.e., ResNet-18 on CIFAR-10, VGG-19 on CIFAR-100, and ResNet-50 on Tiny-ImageNet) to recognize the classes of these generated data. As shown in Table VIII, we tally the number of data categories that could not be generated on different datasets. Since CIFAR-10 dataset comprises just 10 classes, both DaST and DaST + Ours are capable of generating data for all classes. As the number of classes increases on CIFAR-100 dataset, an issue arises in which the diversity of data generated by $\mathcal{G}_S$ in



(a) Complexity comparison. (b) SSIM comparison.

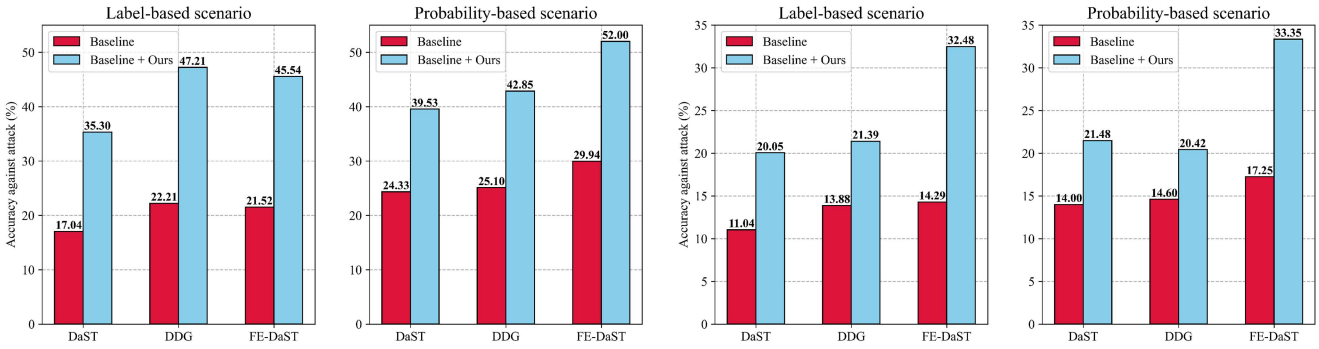
Fig. 5. Our strategies for diversity improvement of generated data. Subgraph (a) illustrates complexity comparisons among different classes, where a larger numerical value indicates higher complexity in an image. Subgraph (b) shows SSIM comparisons among different classes, where a smaller numerical value indicates a greater difference between images within the same class.

FE-DaST is restrained. FE-DaST is unable to generate data for a certain class. On Tiny-ImageNet dataset, where the size of images and the number of classes are far larger than those of CIFAR datasets, the diversity of $\mathcal{G}_S$ is further diminished, and the number of un-generated classes reaches a maximum of 27 in DDG. However, our strategies mitigate the issue effectively. DDG + Ours can generate data for 24 more classes than DDG, meaning that our strategies can improve the inter-class diversity of generated data effectively.

Visualization of generated data: As shown in Fig. 3, we demonstrate the diversity difference between DDG and DDG + Ours on Tiny-ImageNet dataset. We consider generating images for the first ten classes. However, DDG cannot generate images for the 6th category, which hence is replaced by the 11th class. From the richness of the color, the inter-class diversity of data generated by DDG is superior to that of DDG + Ours. Whereas, on closer inspection, there are two obvious issues in DDG: i) within each class, the difference among these generated data is inconspicuous, which means that the intra-class diversity of data is extremely poor; ii) the data of some classes is very similar, e.g., the 1st and 5th classes, the 8th and 10th classes. This represents that the inter-class diversity of generated data is also limited. In contrast, the color of DDG + Ours-generated data is closer to black. Although the color difference among data generated by DDG + Ours is not as pronounced as DDG, our strategies improve the richness of the texture information of data effectively. These texture information composes various data and enriches the complexity and diversity of these data.

Further, we use umap [64] to lower the dimension of generated data and demonstrate the improvements of our strategies for intra-class diversity. As shown in Fig. 4, we visualize the feature distributions of generated data extracted by $\mathcal{T}$ for first ten classes. Note that DDG cannot generate data for the 6th class. Whether viewed in 2-D and 3-D, the data generated by DDG is highly concentrated for each class, demonstrating that the intra-class diversity of data generated DDG is insufficient. However, for DDG + Ours, the same-colored data are more dispersed, which indicates our strategies can increase the intra-class diversity of DDG significantly.

Diveristy of Generated Data: Figs. 3 and 4 visualize the data generated by DDG and DDG + Ours. We further perform two sets of experiments to illustrate the improvement of our proposed strategies for the diversity of generated data.



(a) Robustness comparsion on CIFAR-10-C dataset. (b) Robustness comparsion on CIFAR-100-C dataset.

Fig. 6. Robustness improvements of our proposed strategies for various methods on CIFAR-10-C and CIFAR-100-C datasets.

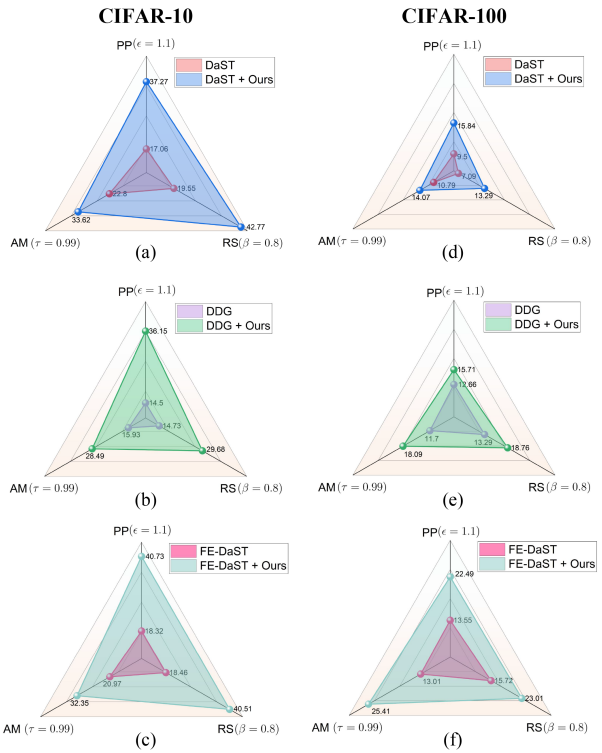


Fig. 7. Robustness improvements of our strategies against three disruption-based defense methods. Subgraphs (a)–(c) and (d)–(f) represent the comparison results of Baseline and Baseline + Ours on CIFAR-10 and CIFAR-100 datasets, respectively.

As shown in Fig. 5(a), we first test the improvement of our strategies on the complexity of the data generated by DDG by calculating the information entropy [62] of an image to demonstrate its complexity. For the 8th class, our strategies improve the complexity of the generated data by up to 24.1%. The last set of experiments only focus on evaluating the diversity of single image. To further test the diversity across all generated data with the same class, we employ the structural similarity (SSIM) measurement [63]. As illustrated in Fig. 5(b), our proposed strategies effectively reduce SSIM scores for different classes of generated data. Notably, for the 7th class, our strategies lowered SSIM score by up to 37.9%.

In sum, the aforementioned experiments validate the efficacy of our approach in ameliorating data diversity from various perspectives.

E. Robustness Improvement

Improvement on Curruption Datasets: Fig. 6 illustrates the robustness enhancements achieved by our strategies for three data-free substitute training techniques (i.e., DaST, DDG, and FE-DaST). Specifically, under the label-based scenario, our strategies are capable of enhancing the robustness of these three attack methods by more than one-fold on CIFAR-10-C dataset. Moreover, our strategies exhibit significant improvements in the robustness of these techniques under the probability-based scenario as well. On CIFAR-100-C dataset, the accuracies against attacks for the three methods do not exceed 18%. However, our strategies are able to elevate their accuracies by up to 33%.

Improvement Against Model Extraction Defenses: Further, we show the robustness improvements of our strategies against the three disruption-based defenses, i.e., PP, RS and AM. The training durations for various methods on CIFAR-10 and CIFAR-100 datasets are 30 and 50 epochs, respectively. As illustrated in Fig. 7, we use radar charts to represent the accuracy improvements of our strategies over the Baselines. Each angle axis denotes one defense strategy. On CIFAR-10 dataset, our proposed strategies improve the accuracy of DaST against RS and AM defenses by up to 23.22% and at minimum 10.82%, respectively. On CIFAR-100 dataset, our strategies raise the accuracy of FE-DaST against AM defense by up to 12.4%, while promoting the accuracy of DDG against PP defense by at least 3.05%. Overall, our proposed strategies are highly effective in enhancing the robustness of Baselines against various defense methods.

VI. Conclusion

In this paper, we analyze the lack of data diversity and the overfitting problems caused by the insufficient diversity, which are prevalent in existing substitute training methods and can lead to a reduced success rate of attacking the target model. Therefore, we propose two strategies to address the lack of diversity, including Dense Projection Space (DPS), and Disguised Natural

Color Mode (DNCM). DPS promotes the diversity of features by projecting random noise into various latent feature spaces. DNCM is able to increase the information exchange between the output layer and the preceding layers, thus generating more diverse data. In addition, to solve the potential overfitting issue, we also propose a noise-based balancing learning to improve the robustness of the substitute model. Extensive experiments are conduct to demonstrate the effectiveness of the proposed three generic strategies.

ACKNOWLEDGMENT

The authors would like to thank NIO company for their financial support.

REFERENCES

[1] M. Goddard, "The eu general data protection regulation (GDPR): European regulation that has a global impact," *Int. J. Market Res.*, vol. 59, no. 6, pp. 703–705, 2017.
[2] C. Szegedy et al., "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*.
[3] M. Shen, Z. Liao, L. Zhu, K. Xu, and X. Du, "VLA: A practical visible light-based attack on face recognition systems in physical world," *Proc. ACM Interactive Mobile Wearable Ubiquitous Technol.*, vol. 3, no. 3, pp. 1–19, 2019.
[4] X. Feng et al., "{Off-Path} network traffic manipulation via revitalized {ICMP} redirect attacks," in *Proc. 31st USENIX Secur. Symp.*, 2022, pp. 2619–2636.
[5] M. Shen, H. Yu, L. Zhu, K. Xu, Q. Li, and J. Hu, "Effective and robust physical-world attacks on deep learning face recognition systems," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 4063–4077, 2021.
[6] Y. Zhao, K. Xu, H. Wang, B. Li, and R. Jia, "Stability-based analysis and defense against backdoor attacks on edge computing services," *IEEE Netw.*, vol. 35, no. 1, pp. 163–169, Jan./Feb. 2021.
[7] X. Feng, Q. Li, K. Sun, Y. Yang, and K. Xu, "Man-in-the-middle attacks without rogue AP: When WPAs meet ICMP redirects," in *Proc. IEEE Symp. Secur. Privacy*, 2022, pp. 694–709.
[8] N. Papernot, P. McDaniel, S. Jha, M. Fredrikson, Z. B. Celik, and A. Swami, "The limitations of deep learning in adversarial settings," in *Proc. IEEE Eur. Symp. Secur. Privacy*, 2016, pp. 372–387.
[9] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: From phenomena to black-box attacks using adversarial samples," 2016, *arXiv:1605.07277*.
[10] Y. Dong et al., "Boosting adversarial attacks with momentum," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9185–9193.
[11] N. Carlini and D. A. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy*, 2017, pp. 39–57.
[12] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, "ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models," in *Proc. 10th ACM workshop Artif. Intell. Secur.*, 2017, pp. 15–26.
[13] A. Ilyas, L. Engstrom, A. Athalye, and J. Lin, "Black-box adversarial attacks with limited queries and information," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 2137–2146.
[14] J. Mu, B. Wang, Q. Li, K. Sun, M. Xu, and Z. Liu, "A hard label black-box adversarial attack against graph neural networks," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2021, pp. 108–125.
[15] F. Li, X. Liu, X. Zhang, Q. Li, K. Sun, and K. Li, "Detecting localized adversarial examples: A generic approach using critical region analysis," in *Proc. IEEE Conf. Comput. Commun.*, 2021, pp. 1–10.
[16] B. Zheng et al., "Black-box adversarial attacks on commercial speech platforms with minimal information," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2021, pp. 86–107.
[17] Q. Wang, B. Zheng, Q. Li, C. Shen, and Z. Ba, "Towards query-efficient adversarial attacks against automatic speech recognition systems," *IEEE Trans. Inf. Forensics Security*, vol. 16, pp. 896–908, 2020.
[18] D. Chen et al., "CARTL: Cooperative adversarially-robust transfer learning," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 1640–1650.
[19] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, 2016, pp. 506–519.
[20] T. Orekondy, B. Schiele, and M. Fritz, "Knockoff nets: Stealing functionality of black-box models," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4949–4958.
[21] Y. Chen, R. Guan, X. Gong, J. Dong, and M. Xue, "D-DAE: Defense-penetrating model extraction attacks," in *Proc. IEEE Symp. Secur. Privacy*, 2022, pp. 432–449.
[22] X. Gong, Y. Chen, W. Yang, G. Mei, and Q. Wang, "InverseNet: Augmenting model extraction attacks with training data inversion," in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 2439–2447.
[23] S. Pal, Y. Gupta, A. Shukla, A. Kanade, S. Shevade, and V. Ganapathy, "ActiveThief: Model extraction using active learning and unannotated public data," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 865–872.
[24] M. Zhou, J. Wu, Y. Liu, S. Liu, and C. Zhu, "DaST: Data-free substitute training for adversarial attacks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 234–243.
[25] W. Wang et al., "Delving into data: Effectively substitute training for black-box attack," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 4761–4770.
[26] M. Yu and S. Sun, "FE-DaST: Fast and effective data-free substitute training for black-box adversarial attacks," *Comput. Secur.*, vol. 113, 2022, Art. no. 102555.
[27] W. Wang, X. Qian, Y. Fu, and X. Xue, "DST: Dynamic substitute training for data-free black-box attack," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 14341–14350.
[28] M. Arjovsky and L. Bottou, "Towards principled methods for training generative adversarial networks," 2017, *arXiv: 1701.04862*.
[29] R. Tang et al., "ZeroWall: Detecting zero-day web attacks through encoder-decoder recurrent neural networks," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 2479–2488.
[30] M. Zhang et al., "Poseidon: Mitigating volumetric DDoS attacks with programmable switches," in *Proc. 27th Netw. Distrib. Syst. Secur. Symp.*, 2020.
[31] J. Cao et al., "The {CrossPath } attack: Disrupting the { SDN} control channel via shared links," in *Proc. 28th USENIX Secur. Symp.*, 2019, pp. 19–36.
[32] Y. Du, H. Duan, L. Xu, H. Cui, C. Wang, and Q. Wang, "PEBA: Enhancing user privacy and coverage of safe browsing services," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 5, pp. 4343–4358, Sep.–Oct. 2023.
[33] C. Lin, J. He, C. Shen, Q. Li, and Q. Wang, "CrossBehaAuth: Cross-scenario behavioral biometrics authentication using keystroke dynamics," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 3, pp. 2314–2327, May/Jun. 2023.
[34] X. Gong, Y. Chen, Q. Wang, M. Wang, and S. Li, "Private data inference attacks against cloud: Model, technologies, and research directions," *IEEE Commun. Mag.*, vol. 60, no. 9, pp. 46–52, Sep. 2022.
[35] D. Liu et al., "SoundID: Securing mobile two-factor authentication via acoustic signals," *IEEE Trans. Dependable Secure Comput.*, vol. 20, no. 2, pp. 1687–1701, Mar./Apr. 2023.
[36] Y. Li, L. Li, L. Wang, T. Zhang, and B. Gong, "NAttack: Learning the distributions of adversarial examples for an improved black-box attack on deep neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2019, pp. 3866–3876.
[37] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM Asia Conf. Comput. Commun. Secur.*, 2017, pp. 506–519.
[38] S. Kariyappa, A. Prakash, and M. K. Qureshi, "Maze: Data-free model stealing attack using zeroth-ordergradient estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 13814–13823.
[39] J. Yang, Y. Jiang, X. Huang, B. Ni, and C. Zhao, "Learning black-box attackers with transferable priors and query feedback," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 12288–12299.
[40] N. Papernot, P. McDaniel, X. Wu, S. Jha, and A. Swami, "Distillation as a defense to adversarial perturbations against deep neural networks," in *Proc. IEEE Symp. Secur. Privacy*, 2016, pp. 582–597.
[41] C. Zhang, P. Benz, A. Karjauv, and I. S. Kweon, "Data-free universal adversarial perturbation and black-box attack," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 7868–7877.
[42] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein generative adversarial networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 214–223.
[43] G. Mordido, H. Yang, and C. Meinel, "microbatchGAN: Stimulating diversity with multi-adversarial discrimination," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2020, pp. 3061–3070.
[44] Y. Bai, T. Ma, and A. Risteski, "Approximability of discriminators implies diversity in GANs," 2018, *arXiv: 1806.10586*.
[45] A. Krizhevsky et al., "Learning multiple layers of features from tiny images," 2009.

[46] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*.
[47] M. Lin, Q. Chen, and S. Yan, "Network in network," 2013, *arXiv:1312.4400*.
[48] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*.
[49] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
[50] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.
[51] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[52] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, 2014.
[53] D. Hendrycks and T. G. Dietterich, "Benchmarking neural network robustness to common corruptions and perturbations," in *Proc. 7th Int. Conf. Learn. Representations*, New Orleans, LA, USA, 2019.
[54] J. Zhang et al., "Towards efficient data free black-box adversarial attack," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 15115–15 125.
[55] T. Orekondy, B. Schiele, and M. Fritz, "Prediction poisoning: Towards defenses against DNN model stealing attacks," in *Proc. Int. Conf. Learn. Representations*, 2019.
[56] T. Lee, B. Edwards, I. Molloy, and D. Su, "Defending against neural network model stealing attacks using deceptive perturbations," in *Proc. IEEE Secur. Privacy Workshops*, 2019, pp. 43–49.
[57] S. Kariyappa and M. Qureshi, "Defending against model stealing attacks with adaptive misinformation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 770–778.
[58] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proc. Int. Conf. Learn. Representations*, 2018.
[59] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *Proc. Int. Conf. Learn. Representations*, 2015. [Online]. Available: http://arxiv.org/abs/1412.6572
[60] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," 2017, *arXiv:1607.02533*.
[61] N. Carlini and D. A. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy*, 2017, pp. 39–57.
[62] C. E. Shannon, "A mathematical theory of communication," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 5, no. 1, pp. 3–55, 2001.
[63] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.
[64] L. McInnes, J. Healy, and J. Melville, "UMAP: Uniform manifold approximation and projection for dimension reduction," 2018, *arXiv: 1802.03426*.

Yang Wei received the PhD degree in cybersecurity from Xidian University, in 2023. He is currently a postdoctoral research associate with the Chongqing University of Posts and Telecommunications, Chongqing, China. His research interests include information and AI security, and deep learning.

Zhuo Ma (Member, IEEE) received the PhD degree in computer architecture from Xidian University, Xi'an, China, in 2010. He is currently an Associate Professor with the School of Cyber Engineering, Xidian University. His research interests include cryptography, machine learning in cyber security, and the Internet of things security.

Zhuoran Ma is currently working toward the PhD degree with the Department of Cyber Engineering, Xidian University. Her current research interests include data security and secure computation outsourcing.

Zhan Qin (Member, IEEE) received the PhD degree from the Department of Computer Science and Engineering, University at Buffalo, State University of New York, in 2017. He is currently an Assistant Professor with the School of Cyber Science and Technology, Zhejiang University. His research interests include intersection of AI security and differential privacy.

Yang Liu received his doctor degree in cybersecurity from Xidian University, in 2022. He is currently an Associate Professor in the School of Cyber Engineering, Xidian University. His research interests are privacy-preserving computation and AI security.

Bin Xiao received the BS and MS degrees in electrical engineering from Shanxi Normal University, Xi'an, China, in 2004 and 2007, respectively, and the PhD degree in computer science from Xidian University, Xi'an. He is currently a professor with the Chongqing University of Posts and Telecommunications, Chongqing, China. His research interests include information security and pattern recognition.

Xiuli Bi received the BSc and MSc degrees from Shanxi Normal University, Xi'an, China, in 2004 and 2007, respectively, and the PhD degree in computer science from the University of Macau, Macau, in 2017. She is currently a professor with the College of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, China. Her research interests include medical image processing, multimedia security, and image forensics.

Jianfeng Ma (Member, IEEE) received the BS degree in mathematics from Shaanxi Normal University, Xi'an, China, in 1985, and the MS and PhD degrees in computer software and telecommunication engineering from Xidian University, Xi'an, in 1988 and 1995, respectively. He is currently a professor with the School of Cyber Engineering, Xidian University. He is also the director of the Shaanxi Key Laboratory of Network and System Security. His current research interests include information and network security and mobile computing systems.