This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Exploring Effective Data for Surrogate Training Towards Black-box Attack

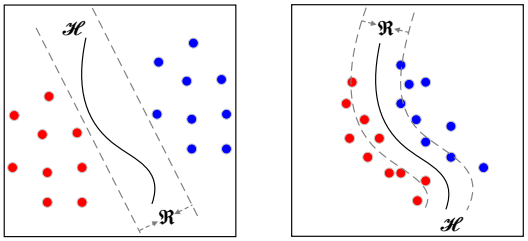
Xuxiang Sun Gong Cheng* Hongda Li Lei Pei Junwei Han
School of Automation, Northwestern Polytechnical University, Xi'an, China
{xuxiangsun,hongda,peilei}@mail.nwpu.edu.cn {gcheng,jhan}@nwpu.edu.cn

Abstract

Without access to the training data where a black-box victim model is deployed, training a surrogate model for black-box adversarial attack is still a struggle. In terms of data, we mainly identify three key measures for effective surrogate training in this paper. First, we show that leveraging the loss introduced in this paper to enlarge the inter-class similarity makes more sense than enlarging the inter-class diversity like existing methods. Next, unlike the approaches that expand the intra-class diversity in an implicit model-agnostic fashion, we propose a loss function specific to the surrogate model for our generator to enhance the intra-class diversity. Finally, in accordance with the in-depth observations for the methods based on proxy data, we argue that leveraging the proxy data is still an effective way for surrogate training. To this end, we propose a triple-player framework by introducing a discriminator into the traditional data-free framework. In this way, our method can be competitive when there are few semantic overlaps between the scarce proxy data (with the size between 1k and 5k) and the training data. We evaluate our method on a range of victim models and datasets. The extensive results witness the effectiveness of our method. Our source code is available at https://github.com/xuxiangsun/ST-Data.

1. Introduction

Over the last decades, we have witnessed the blowout success of Deep Neural Networks (DNNs) in computer vision tasks. Yet they also show pervasive brittleness when they are exposed to adversarial examples [14, 43]. This leaves great safety hazards for the practical deployment of DNNs. Hence, more threatening adversarial examples had been crafted to push for further research [3, 14, 23, 26, 31]. Most of them assume that the prior information of a victim model (denoted by V) can be accessed, *e.g.*, its internal architecture or training data. Albeit this lenient attack scenario (*i.e.*, the white-box setting) can help us to explore the robustness of DNNs, their performance will get hurt drastically no prior knowledge (*i.e.*, the black-box setting) can be accessed. In this case, how to attack DNN has stirred great interest in scholars.



(a) Samples with insufficient imitation ability (b) Samples with sufficient imitation ability

Figure 1. **Illustration of the location relationship in feature space between (a) Samples with insufficient imitation ability, and (b) Samples with sufficient imitation ability.** Here, $\mathscr{H}$ denotes the decision boundary of a model, $\Re$ represents the imitation margin of the synthesized samples.

Under the black-box setting, only the input-output feedback of the victim model V can be accessed. According to the output, the solutions to black-box attack can either be decision-based (only the final label can be accessed) or score-based (the output logits can be obtained). Among them, a feasible way is to design effective searching algorithms [1, 2, 4–6, 9, 15, 27, 32, 36, 37]. However, with a low query budget, their efficiency may be limited exceedingly. While another intuitive idea is training a local surrogate model (denoted by S) to imitate the remote victim model V, and then craft adversarial examples on the trained model S via existing white-box attacks. However, without the training data of model V, training the model S is still strenuous.

Recent advances [46, 58] pointed out that leveraging the synthesized data is more effective than the real proxy data [33, 34]. Specifically, [46] mentioned that this may be attribute to the poor diversity of the proxy data with limited size. In a way, it encounters the underfitting problem. From this perspective of view, the synthesized data can be served as the proxy data with infinite size and relatively large diversity. In this case, an important question is: *what kind of synthesized data are effective to train the model S?*

To deal with this question, the most advanced meth-

*Corresponding author.

ods [46, 58] enlarge the inter-class diversity of the synthesized data. However, as shown in Fig. 1, if the imitation margin of the samples stays away from the decision boundary like Fig. 1a, then the surrogate model S can not learn how the victim model V makes decisions. Things will get worse when it comes to the case of the label-only scenario. Instead, if the synthesized data can lay close to the decision boundary (*i.e.*, with the large inter-class similarity), the imitation ability will be sufficient, like Fig. 1b. This also can be revealed in Fig. 2 (details will be described in Appendix C.1), where the boundary loss measures the distance from the synthesized data to the decision boundary. For most cases when the boundary loss is relatively high, the Attack Success Rate (ASR) of DaST [58] and Knockoff [33] can not grow up sustainably. Compared with them, our method leads to continuous improvement via keeping a lower loss. Besides, as indicated by [46], the intra-class diversity makes sense for surrogate training. Nonetheless, there is no explicit constraint in [46] specified to the surrogate model to enlarge the intra-class diversity, which can not promise the effectiveness of the synthesized data to the model S.

Last but not least, according to [33], when the size of the proxy dataset is large enough, it still can work well. Hence, we make a feasible conjecture, *i.e.*, the distribution of the proxy dataset can be roughly divided into two parts, *i.e.*, the first one contains the samples that are useful for surrogate training while the second one does not. Different datasets contain different proportions of valid data. Based on this assumption, we can use the vanilla Generative Adversarial Networks (GAN) [13, 30, 35] to imitate the distribution of a proxy dataset. As shown in Fig. 2, the synthesized data can still be effective (lower loss and better performance than DaST and Knockoff). Moreover, it is striking that the ASR of vanilla GAN can even exceed DaST. We think that is due to searching from the whole data space like [46, 58] may be ineffective within the same training budget. Consequently, their performance will get stuck.

Spurred by the above observations and assumptions, we make the following contributions: 1) We propose a triple-player framework to train the surrogate model for the black-box adversarial attack. Specifically, based on the traditional data-free training framework, we are the first to introduce a discriminator to limit the searching space of our generator, which can enhance the training efficiency; 2) We identify that enlarging the inter-class similarity makes more sense than the inter-class diversity. Then, a loss function is introduced in this paper to enlarge the inter-class similarity of the synthesized data; 3) We propose a new loss function specified to the surrogate model to boost the intra-class diversity explicitly; 4) We indicate that the effectiveness of our method is competitive when there are few semantic overlaps between the scarce proxy data (*i.e.*, with the size between 1k and 5k) and the training data.

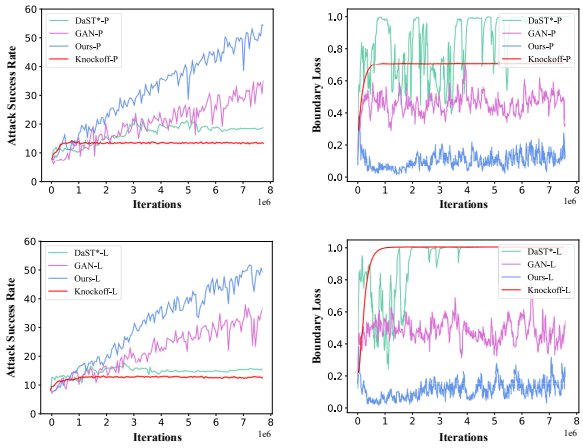


Figure 2. **Iterations vs. Attack Success Rate (Untargeted) and Boundary Loss (BL).** Here, BL denotes the distance from the synthesized data to the decision boundary. "*-P" denotes the probability-only scenario and "*-L" represents the label-only scenario. To eliminate the influence of other factors, we replace the generator of DaST with ours, dubbed "DaST*".

2. Related Work

Adversarial Attacks. As illustrated in Sec. 1, under the white-box setting, early works [3, 14, 23, 26, 31] focus on designing adversarial examples constrained within tight ℓ_p $(p = 1, 2, \infty)$ norm bounds in RGB space. Recently, researchers tried to craft adversarial examples from different points of view. For instance, [55, 56] explored the adversarial examples in different color spaces, [51, 52] revealed that designing adversarial examples in feature space is also remarkable. Besides, in frequency space, [12] identified that dropping useless information yields more imperceptible adversarial examples.

For the black-box attacks, besides what we mentioned in Sec. 1, improving the transferability of adversarial examples is also another subject of intense research. For instance, improving the transferability of the gradient-based attacks via efficient gradient calculation [10, 23, 25, 47] or input transformations [11, 50]. Besides, [18–20] found that designing adversarial examples via the intermediate features yields more transferable adversarial examples. Moreover, [49] indicated that the DNN architecture itself can expose more transferability.

Surrogate Training. If we can access the training data and the internal gradients of the victim model, the methods based on knowledge distillation [17] may be effective to train a surrogate model for black-box adversarial attack. Also, [44, 59] are still remarkable if we can only acquire the training data. Unfortunately, the above assumptions are impractical in reality. Hence, [33, 34] steal the functionality of the victim model V via the proxy dataset. Recent

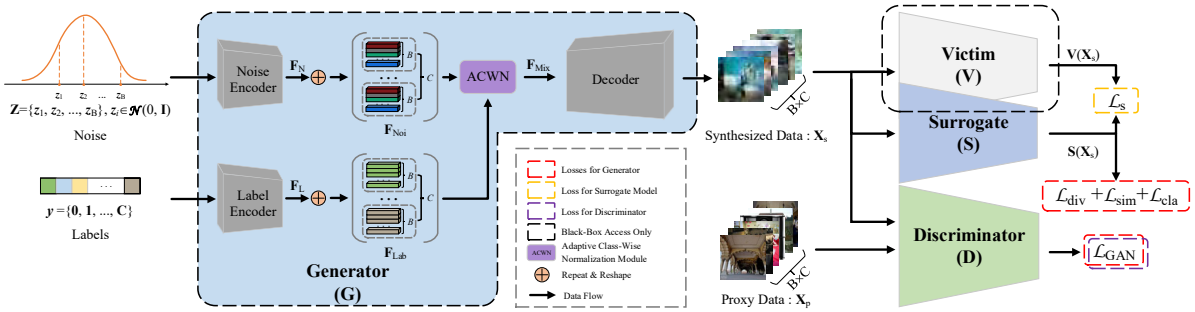


Figure 3. **The framework of the proposed method.** Given a series of noise and labels, our generator synthesizes the data $\mathbf{X}_s$ firstly. Then the victim model V will label $\mathbf{X}_s$ as $V(\mathbf{X}_s)$ (in the form of the final label or the probabilities of each class), which will be leveraged to train the surrogate model S via loss $\mathcal{L}_s$. In addition, the discriminator D will learn to distinguish $\mathbf{X}_s$ with the proxy data $\mathbf{X}_p$ via loss $\mathcal{L}_{GAN}$. Finally, our generator G will be optimized. Here, B is the number of input noise, C represents the class number of a victim dataset. Besides, in $\mathbf{F}_{Lab}$, the features with the same color belong to the same class, which correspondence to the input label with the same color.

works assumed that no proxy images can be leveraged. In this scenario, [46, 53, 54, 58] propose to leverage the idea of GAN [13, 30] to synthesize data for surrogate training. Specifically, [53, 54] synthesize images from noise or recover training data from the victim model. However, under the strict black-box setting, they will be helpless.

Exposed to the challenge of black-box and data-free settings, DaST [58] is the first to steal the functionality of a black-box model without real data. It uses a generator to synthesize data that can cause decision conflicts between the victim model and the surrogate model. Then, DDG [46] goes deeper to explore more effective data. It first modifies the architecture of the generator to compress its size, since the size of the generator in DaST [58] will be enlarged extremely when the class number of the victim dataset grows up. Besides, it introduces a reconstruction network to enhance the intra-class diversity by realizing the one-to-one mapping from random noise to images. Finally, the adversarial training strategy is further deployed.

3. Proposed Method

3.1. Adaptive Class-Wise Normalization

It is worthy of emphasizing that the synthesized data should be label-controllable firstly. In other words, given an input label, the synthesized data should be classified by the surrogate model S into the corresponding class. Otherwise, there may be the problem of mode collapse, since the methods with GAN suffer from this problem usually [13, 39].

Hence, similar to [46], our generator includes a label encoder, which consists of an embedding layer [29] followed by several fully-connected layers. Besides, several fully-connected layers are combined in series to form the noise encoder. In this case, with the noise $\mathbf{Z} = \{\boldsymbol{z}_1, \boldsymbol{z}_2, ..., \boldsymbol{z}_B\}$ where $\boldsymbol{z}_i \in \mathcal{N}(\mathbf{0}, \mathbf{I})$, and the labels $\boldsymbol{y} = \{0, 1, ..., C\}$, we can get the noise features $\mathbf{F}_N \in \mathbb{R}^{B\times L\times H\times W}$ and the label features $\mathbf{F}_L \in \mathbb{R}^{C\times L\times H\times W}$ via the corresponding encoder, where L is the number of channels, B is the number of input noise, C is the class number of the victim dataset, H and W are the height and width of the corresponding features respectively.

Once we get $\mathbf{F}_N$ and $\mathbf{F}_L$, they are repeated and reshaped to acquire $\mathbf{F}_{Noi} \in \mathbb{R}^{(B\times C)\times L\times H\times W}$ and $\mathbf{F}_{Lab} \in \mathbb{R}^{(B\times C)\times L\times H\times W}$, as shown in Fig. 3. Then $\mathbf{F}_{Noi}$ and $\mathbf{F}_{Lab}$ will be fed into our Adaptive Class-wise Normalization (ACWN) module followed by a decoder to generate the synthesized images $\mathbf{X}_s$. Details about ACWN can be accessed in our source code.

Formally, we formulate $\mathbf{X}_s$ as shown in Eq. (1):

$$\begin{cases} \mathbf{X}_s = \{\boldsymbol{x}_j^k\}(k = \{1, 2, ..., C\}, j = \{1, 2, ..., B\}) \\ \quad\; = \mathrm{G}(\mathbf{Z}, \boldsymbol{y}) \end{cases}. \tag{1}$$

Here, $\boldsymbol{y}$ represents the input labels and $\mathbf{Z}$ denotes the input noise. Besides, the variable k in the upper right corner of $\boldsymbol{x}_j^k$ represents the index of its input label, and j in its bottom right corner denotes the index of its input noise. For the sake of expression, we use $\mathbf{Y} \in \mathbb{R}^{(B\times C)\times 1}$ to denote the corresponding original input labels of $\mathbf{X}_s$.

As we noted at the beginning of this subsection, the generator in our framework should be label-controllable. Thus, one of the optimization objective functions for the generator G is the cross-entropy loss, as shown in Eq. (2).

$$\mathcal{L}_{cla} = \mathrm{CE}(\mathrm{S}(\mathbf{X}_s), \mathbf{Y}). \tag{2}$$

Here, $\mathrm{S}(*) \in \mathbb{R}^C$ denotes the output logits except for the softmax layer of the surrogate model S.

3.2. Inter-Class Similarity

Since Eq. (2) has provided a label-controllable constraint, we now introduce a loss function to boost the inter-class similarity of the synthesized data. Recall that [3] improves the adversarial ability via Eq. (3):

$$\max\{\max\{f(\boldsymbol{x})_i : i \neq t\} - f(\boldsymbol{x})_t, -\kappa\}. \tag{3}$$

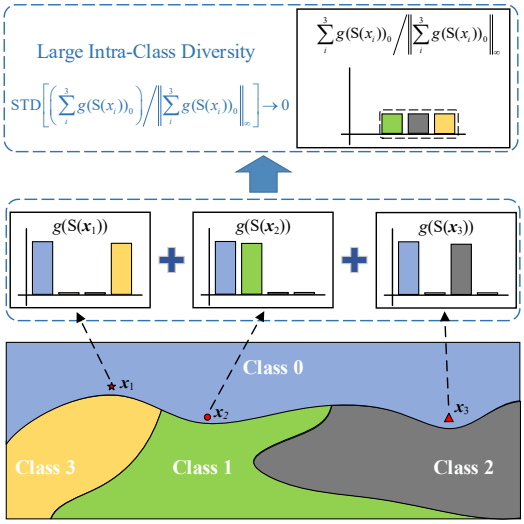


Figure 4. **Illustration of the proposed intra-class diversity loss.** Here, $g(*)_0$ represents applying the softmax function to $*$, of which the 0-th entry is excepted. In the bottom part, different colors represent different classes. In the middle part, the output probabilities from the surrogate model S are shown. Each bar with a certain color is the probability of the class with the same color in the bottom. In the top part, the function STD calculates the standard deviation of all entries in its input.

Here, given the input data $\boldsymbol{x}$, $f(\boldsymbol{x})$ denotes the output of a DNN f except for the softmax layer, and $f(\boldsymbol{x})_i$ represents the i-th entry of $f(\boldsymbol{x})$, and t is the target label (for the targeted attack) or the original label (for the untargeted attack).

In this paper, we revisit Eq. (3) as the one that aims at pushing $\boldsymbol{x}$ move close to the decision boundary between class t and j, where $j = \max_{\theta}\{f(\boldsymbol{x})_\theta : \theta \neq t\}$. Thus, to enlarge the inter-class similarity, we introduce a loss based on Eq. (3) into the optimization of the generator. To be specific, as shown in Eq. (4),

$$\mathcal{L}_{\text{sim}} = \frac{||\sum_{k}^{C}\sum_{j}^{B}\{\max\{\mathrm{S}(\boldsymbol{x}_j^k)_i : i \neq k\} - \mathrm{S}(\boldsymbol{x}_j^k)_k\}||}{B \times C}, \quad (4)$$

where $\mathrm{S}(*)_i$ means the i-th entry of $\mathrm{S}(*)$.

Associating Eq. (2) with Eq. (4), we can push the generated samples as closely as possible to their decision boundary of the corresponding class while keeping these samples to be label-controllable for our generator. Moreover, it is worth emphasizing that enhancing the inter-class similarity does not mean confusing the class-specific features, instead, Eq. (2) has provided a guarantee about this. Also, the visualization of the generated samples in Appendix C.3 can still provide empirical evidence for this.

3.3. Intra-class Diversity

Suppose we optimize the generator by Eq. (2) and Eq. (4), we can make an ideal assumption that the generated samples are label-controllable with large inter-class similarity. In this scenario, to deal with the intra-class diversity, let us consider a typical case shown in Fig. 4. Here, all these three samples belong to class 0. We consider the location as the metric that measures the inter-class diversity. In other words, if they are distributed discretely near the decision boundary of class 0 and the other three classes, as depicted at the bottom of Fig. 4, then the inter-class diversity of class 0 will be the ideal solution. In this case, these three samples can explore all the class regions.

Now, we start to analyze how to formulate our loss function mathematically. Recall that in Sec. 3.2, we introduce Eq. (4) to describe the distance between an input sample and the decision boundary. Thus, as shown at the top of Fig. 4, we can find that the probabilities except class 0 are almost equal. Unfortunately, we do not know what the exact value is. Instead, we turn to another way of thinking about it, *i.e.*, the standard deviation. Specifically, in this case, the standard deviation is close to zero. Thus, we design Eq. (5) as the loss function to enlarge the intra-class diversity, as comes below:

$$\begin{aligned}\mathcal{L}_{\text{div}} &= \frac{1}{C}\sum_{k}^{C}\frac{1}{m}\mathrm{STD}[\sum_{j}^{C-1} Norm(g(\mathrm{S}(x_j^k))_k)] \\ s.t.\ B &= m \times (C-1), m \in \mathrm{N}_+\end{aligned}, \quad (5)$$

where $g(*)_k$ represents applying the softmax function to $*$, of which the k-th entry is excepted, and $\mathrm{STD}[*]$ means calculating the standard deviation of $*$ along the first dimension. $Norm(x)$ denotes normalizing x by Eq. (6):

$$Norm(x) = \frac{x - \min(x)}{\max(x - \min(x))}. \quad (6)$$

Here, $\max(x)$ and $\min(x)$ mean extracting the maximum and the minimum of x along its first dimension, respectively. It is worth noting that in Eq. (5), B should be an integer multiple of $C-1$ (N_+ is the set of positive integers). However, when C gets increased, B cannot be set to a large number limited by the hardware conditions (*i.e.*, $B < (C-1)$). We will give the formulation about this case in Appendix A.

3.4. Optimization

For the optimization of the model S in Fig. 3, we use Eq. (7) as its objective function, which comes as below:

$$\mathcal{L}_{\text{s}} = \mathrm{CE}(\mathrm{S}(\mathbf{X}_{\text{s}}), \mathbf{Y}_{\text{v}}) + \alpha||\mathrm{S}(\mathbf{X}_{\text{s}}), \mathrm{V}(\mathbf{X}_{\text{s}})||_{\text{F}}, \quad (7)$$

where $\mathbf{Y}_{\text{v}}$ are the labels corresponding to $\mathbf{X}_{\text{s}}$ output by the victim model V, *i.e.*, $\mathbf{Y}_{\text{v}} = \mathrm{V}(\mathbf{X}_{\text{s}})$. $||\cdot||_{\text{F}}$ denotes the MSE

Table 2. **Performance comparison in terms of untargeted attack success rate over several victim models deployed on four datasets.** Here, the experimental settings are reported in Tab. 1. "*-P" denotes the probability-only attack scenario and "*-L" represents the label-only attack scenario. **RED**/**BLUE** indicate the best/the second best. The same as Tab. 3 and Tab. 4.

DataSet	MNIST [24]			CIFAR-10 [21]			CIFAR-100 [21]		Tiny-ImageNet [38]
Victim Model	AlexNet	VGG-16	ResNet-18	AlexNet	VGG-16	ResNet-18	VGG-19	ResNet-50	ResNet-50
Knockoff$_{P1}$-P [33]	42.47	40.75	40.16	27.17	21.36	24.40	15.36	12.52	11.47
Knockoff$_{P2}$-P [33]	45.53	45.14	42.97	28.71	22.04	25.96	16.09	14.24	13.86
DaST-P [58]	58.86	54.82	59.62	50.28	32.45	42.77	27.39	26.18	28.81
DDG-P [46]	66.31	62.84	70.27	55.76	42.31	46.82	35.48	39.29	34.28
Ours$_{P1}$-P	**91.70**	**90.14**	**85.59**	**62.23**	**75.51**	**74.24**	**57.28**	**61.42**	**59.65**
Ours$_{P2}$-P	**94.95**	**93.63**	**85.66**	**64.89**	**77.54**	**76.16**	**60.83**	**64.08**	**62.81**
Knockoff$_{P1}$-L [33]	27.45	28.48	31.38	18.29	15.65	16.38	8.11	7.81	7.25
Knockoff$_{P2}$-L [33]	28.66	29.82	32.21	19.52	16.86	17.47	9.74	8.73	8.46
DaST-L [58]	26.51	29.22	35.81	25.18	19.34	23.01	17.34	17.27	16.28
DDG-L [46]	31.74	32.70	40.96	29.44	26.92	23.38	23.48	27.88	28.31
Ours$_{P1}$-L	**90.00**	**88.34**	**84.56**	**58.39**	**73.57**	**71.23**	**56.15**	**58.08**	**57.81**
Ours$_{P2}$-L	**92.04**	**88.65**	**84.76**	**63.32**	**75.00**	**73.50**	**59.81**	**61.16**	**60.98**

Table 1. **Default Settings on each victim dataset.** Here, VD for the victim dataset, S for the default surrogate model, B for batch-size, LR for the learning rate of model S, P1 and P2 are two proxy datasets, and N are the number of images for both P1 and P2.

VD	MNIST	CIFAR-10	CIFAR-100	Tiny-ImageNet
N	4k	4k	4k	4k
B	135	135	10	8
S	Small	VGG-13	ResNet-18	ResNet-34
LR	0.0001	0.0001	0.0002	0.0002
P1	EMNIST	CUB-200	CUB-200	CUB-200
P2	KMNIST	Places365	Places365	Places365

loss. For the label-only scenario, we set $\alpha = 0$. While for the probability-only scenario, $\alpha = 1$.

For the optimization of the discriminator, we use the loss function proposed by [28], *i.e.*, comes as Eq. (8):

$$\mathcal{L}_{\text{GAN}}^{\text{D}} = \mathbb{E}_{\boldsymbol{x}\sim\mathbf{X}_{\text{p}}}[\text{D}(\boldsymbol{x}) - 1]^2 + \mathbb{E}_{\boldsymbol{x}\sim\mathbf{X}_{\text{s}}}[\text{D}(\boldsymbol{x})]^2. \quad (8)$$

Here, $\mathbf{X}_{\text{p}}$ represents the proxy dataset. Also, the adversarial loss should be included to optimize the generator, *i.e.*, Eq. (9):

$$\mathcal{L}_{\text{GAN}}^{\text{G}} = \mathbb{E}_{\boldsymbol{x}\sim\mathbf{X}_{\text{s}}}[\text{D}(\boldsymbol{x}) - 1]^2. \quad (9)$$

Finally, the total loss function for the generator is Eq. (10):

$$\mathcal{L}_{\text{G}} = \mathcal{L}_{\text{GAN}}^{\text{G}} + \beta_1\mathcal{L}_{\text{div}} + \beta_2\mathcal{L}_{\text{sim}} + \beta_3\mathcal{L}_{\text{cla}}. \quad (10)$$

4. Experiments

4.1. Experimental Settings

In this section, we will give the introductions about the main settings in our experiments, including the datasets, model architectures, attacks that are utilized to evaluate the performance and the evaluation metrics.

Datasets and Model Architectures. In general, there are a total of four victim datasets (*i.e.*, MNIST [24], CIFAR-10 [21], CIFAR-100 [21], and Tiny-ImageNet [38]) are deployed to varify the effectiveness of our method. Besides, we also provide a total of four proxy datasets (*i.e.*, EMNIST [8], KMNIST [7], Places365 [57], and CUB-200 [48]) for our method and Knockoff [33] to carry out surrogate training. For the model architecture, we leverage a total of nine models, which belong to five different types (*i.e.*, a model with 3 convolution layers, dubbed "Small" in this paper, AlexNet [22], GoogleNet [42], MobileNet-V2 [40], VGG-Net [41], ResNet [16]). On each victim dataset, the default settings can be seen in Tab. 1. It is worth noting that for all the proxy datasets leveraged in this paper, there are almost no semantic overlaps between them and the victim datasets.

Attack Methods and Evaluation Metrics. In our experiments, we evaluate the performance mainly via four attack methods, including FGSM [14], BIM [23], PGD [26], and C&W [3]. PGD [26] is the default attack method to evaluate the performance of our method, unless specified. For the evaluation metrics, we use the targeted Attack Success Rate (ASR_{tar}) and the untargeted attack success rate ($\text{ASR}_{\text{untar}}$). The calculation of these two metrics can be seen in Appendix B.

Implementation Details. For the implementation, our code is based on Pytorch deep learning framework. We use Adam optimizer to train all the networks of our method. The hyper-parameters in Eq. (10) are $\beta_1 = \beta_2 = 2, \beta_3 = 1.2$. The other default settings are reported in Tab. 1. All the experiments in this paper are conducted by one NVIDIA GeForce RTX 3090 GPU. Besides, all the proxy images are randomly chosen at the beginning and frozen during training. In addition, the learning rates of both the generator

Table 3. **Performance comparison in terms of targeted attack success rate over several victim models deployed on four datasets.**

DataSet	MNIST [24]			CIFAR-10 [21]			CIFAR-100 [21]		Tiny-ImageNet [38]
Victim Model	AlexNet	VGG-16	ResNet-18	AlexNet	VGG-16	ResNet-18	VGG-19	ResNet-50	ResNet-50
Knockoff$_{P1}$-P [33]	24.07	23.32	28.96	16.59	11.04	13.13	4.34	5.26	4.97
Knockoff$_{P2}$-P [33]	25.43	24.89	30.08	17.36	11.29	13.89	4.38	5.32	5.08
DaST-P [58]	50.17	52.84	51.29	29.93	16.28	21.44	10.84	15.81	13.92
DDG-P [46]	39.29	**57.28**	**64.46**	**33.81**	29.89	25.77	17.23	21.44	19.37
Ours$_{P1}$-P	**55.36**	54.63	54.17	29.00	**40.31**	**39.35**	**25.42**	**28.32**	**27.43**
Ours$_{P2}$-P	**61.05**	**57.93**	**57.11**	**34.07**	**41.21**	**40.92**	**29.22**	**32.60**	**30.93**
Knockoff$_{P1}$-L [33]	10.10	15.99	8.98	7.99	6.23	7.35	3.37	2.25	2.22
Knockoff$_{P2}$-L [33]	14.28	17.41	10.37	8.61	7.42	8.68	3.44	2.25	2.26
DaST-L [58]	20.03	21.48	19.33	15.72	15.92	14.83	7.48	10.39	10.31
DDG-L [46]	25.56	27.64	21.83	21.66	18.67	17.90	12.47	16.26	13.39
Ours$_{P1}$-L	**53.28**	**50.33**	**40.77**	**26.53**	**37.95**	**35.73**	**24.47**	**26.29**	**25.61**
Ours$_{P2}$-L	**54.33**	**51.92**	**44.96**	**30.46**	**39.93**	**36.60**	**26.19**	**29.85**	**28.62**

Table 4. **Method Ablation Studies on CIFAR-10 [21] and CIFAR-100 [21] datasets.** Here, "Base" means using Eq. (2) only to train the surrogate model. The default proxy dataset for both is Places365 [57], and the victim models are VGG-16 [41] (CIFAR-10) and ResNet-50 [16] (CIFAR-100). The other settings are the corresponding ones in Tab. 1

DataSet	CIFAR-10-P					CIFAR-10-L					CIFAR-100-P					CIFAR-100-L				
Base	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
$\mathcal{L}_{sim}$				✓	✓				✓	✓				✓	✓				✓	✓
$\mathcal{L}_{div}$			✓		✓			✓		✓			✓		✓			✓		✓
$\mathcal{L}_{GAN}$		✓	✓	✓	✓		✓	✓	✓	✓		✓	✓	✓	✓		✓	✓	✓	✓
ASR$_{untar}$	18.60	28.59	49.70	**50.92**	**77.54**	10.16	24.73	46.26	**50.20**	**75.00**	3.31	27.40	42.84	**45.67**	**64.08**	2.38	20.28	41.38	**44.65**	**61.16**
ASR$_{tar}$	3.52	12.68	23.86	**26.84**	**41.21**	1.98	11.21	20.38	**25.05**	**39.93**	0.31	15.94	18.46	**20.34**	**32.60**	0.21	13.63	17.22	**18.11**	**29.85**

Table 5. **Performance evaluation on MNIST dataset [24] via different attacks (shown in the first column) with two proxy datasets in terms of both ASR$_{untar}$ and ASR$_{tar}$ (separated with a double line).** Here, the victim model is AlexNet [22], and the other settings are the reported in Tab. 1.

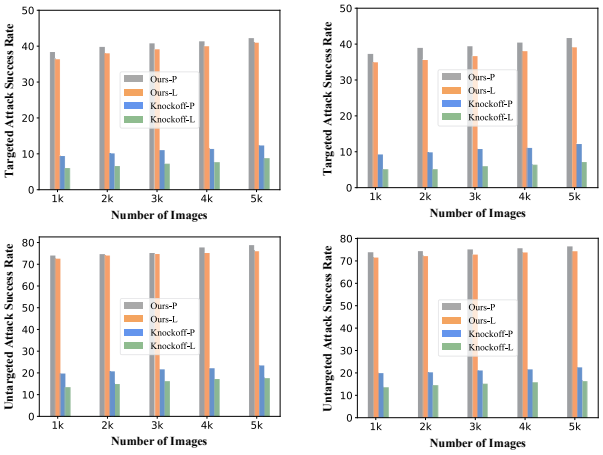
Attacks	Probability-Only			Label-Only		
	P2	P1	DDG	P2	P1	DDG
FGSM$_{untar}$	**83.71**	**82.86**	57.35	**81.33**	**81.19**	33.10
BIM$_{untar}$	**94.15**	**90.76**	68.45	**90.87**	**88.37**	29.58
PGD$_{untar}$	**94.95**	**91.70**	66.31	**92.04**	**90.00**	31.74
C&W$_{untar}$	**68.70**	**55.74**	46.93	**62.30**	**55.63**	22.02
FGSM$_{tar}$	**32.00**	**31.10**	29.48	**24.79**	**23.52**	19.25
BIM$_{tar}$	**62.57**	**56.45**	44.82	**56.27**	**55.21**	18.14
PGD$_{tar}$	**61.05**	**55.36**	39.29	**54.33**	**53.28**	25.56
C&W$_{tar}$	**40.00**	**38.26**	28.57	**38.44**	**33.48**	19.66

and the discriminator are all $\times 5$ as the learning rate of the surrogate model that is reported in Tab. 1. For the experiments in Tabs. 2 to 5, the learning rates of all the three networks in Fig. 3 (*i.e.*, the generator, the discriminator, and the surrogate model) decrease linearly to zero from the 75-th epoch and stop after the 150-th epoch. Moreover, most of the experimental items in this paper follow the protocol of state-of-the-art baseline [46].

4.2. Peer Comparisons

In this section, we evaluate our approach on four victim datasets under two attack scenarios, *i.e.*, the probability-only case and the label-only case. The competitors in this part include a method based on proxy dataset (*i.e.*, Knockoff [33]) and two data-free methods (*i.e.*, DaST [58] and DDG [46]). For a fair comparison, we evaluate the performance of Knockoff with the two proxy datasets leveraged by our method in this paper instead of those utilized in their paper. We run our methods five times over each evaluation, and remove the maximum and the minimum to calculate the average of the remaining three. The results in this part are summarized in Tab. 2 (for ASR$_{untar}$) and Tab. 3 (for ASR$_{tar}$). The experimental settings on each dataset are those in Tab. 1.

Associating Tab. 2 with Tab. 3, four conclusions can be established preliminarily. First, when there are almost no semantic overlaps between the scarce proxy data (*e.g.*, 4k images) and the training data, the performance of Knockoff will be very poor. By contrast, leveraging synthesized samples is a wise approach. Next, compared to state-of-the-



(a) Places365 dataset (b) CUB-200 dataset

Figure 5. **Data ablation studies on CIFAR-10 dataset with (a) Places365 [57] and (b) CUB-200 [48].** Here, we set the number of proxy images between 1k and 5k to evaluate the performances of our method and Knockoff under two attack scenarios. The victim model is VGG-16 [41].

art method [46], we can outperform it by a large margin in most cases in terms of both targeted and untargeted attack scenario, especially in untargeted attack scenario. These results demonstrate that in the large majority of cases, all of our measures are quiet effective for surrogate training. Besides, as expected, for the label-only attack scenario, we can exceed all our competitors with a large margin in all cases. For the untargeted attack, the performance of our method in label-only case can even be as good as that in probability-only case. Moreover, equipped with different proxy datasets, our method exhibits different performances. Recall that in Sec. 1, we assumed that different proxy datasets contain different numbers of valid samples. The results in Tabs. 2 and 3 then give us the empirical validations of this idea. That is to say, with the same searching algorithm, P1 dataset may contain more regions that belong to effective data than P2 dataset.

4.3. Further Analyses

Method Ablation Studies. In this part, we further analyze the effects of different components in our method. Specifically, the ablation terms are shown in Tab. 4.

The results in Tab. 4 can be summarized as the following: 1) Searching over the whole image space within limited training steps is very inefficient. However, if we limit the seeking space by introducing the proxy dataset, things can get better greatly (*i.e.*, optimizing the generator with Eq. (2) and Eq. (9)); 2) Based on Eq. (2) and Eq. (9)), improving the intra-class diversity or enhancing the inter-class similarity are all effective. Compared to those two losses with each other, the inter-class similarity seems to make more sense than the intra-class diversity. That is, the prerequisite for living up to the potential of the intra-class diversity is that we have enlarged the inter-class similarity; 3) Compared to probability-only scenario, pushing the synthesized data towards the decision boundary is more effective in label-only scenario, especially when the class number is small; 4) Bring all the components of our method together, the imitation efficiency can be mined excitedly.

Table 6. **Performance evaluation on CIFAR-10 dataset [21] with different surrogate models (shown in the first column) in terms of both $\text{ASR}_{\text{untar}}$ and ASR_{tar}.** Here, the victim model is VGG-16 [41], and the proxy dataset is Places365 [57].

Surrogate	$\text{ASR}_{\text{untar}}$		ASR_{tar}	
	Ours-P	Ours-L	Ours-P	Ours-L
GoogleNet [42]	78.87	77.08	45.91	43.32
MobileNet-V2 [40]	73.04	71.91	39.33	37.75
VGG-16 [41]	74.60	73.35	40.43	38.07
VGG-19 [41]	75.79	74.99	41.00	38.98
ResNet-18 [16]	78.40	76.41	45.65	42.45
ResNet-34 [16]	79.31	78.70	46.93	44.74

Data Ablation Studies. Since we leverage the proxy data in our method, we now give an ablation study on the number of proxy images w.r.t the ASR (both targeted and untargeted are reported). The results on CIFAR-10 are summarized in Fig. 5 (extended results on CIFAR-100 can be found in Appendix C.2.). Looking through Fig. 5, we can find that our method is not very sensitive to the length of the proxy images. That is, the data space established by the proxy images with the size between 1k and 5k seems to not have an impressive impact on the efficiency of our method. But with the same size of the proxy images, we can outperform Knockoff by a large margin. From this perspective of view, the way we utilize the proxy data is more effective.

4.4. Extended Evaluations

Evaluations with different attacks. Here, we evaluate our method on MNIST dataset [24] with different attacks, *i.e.*, FGSM [14], BIM [23], PGD [26], and C&W [3]. As shown in Tab. 5, we can see that different attacks give consistent results for measuring the strength of different methods. Our method can still keep its performance under various attacks. Thus, in consistent with the suggested by DDG [46], we can see that there is no need to restrict the attack method for the evaluation of the performance.

Evaluations with different surrogate model. In reality, we do not know exactly the architecture of the victim model. Thus, to give a further study in the light of the architecture of the surrogate model, we equip the same victim model with various surrogate models. It is worth noting that here we only train the surrogate model for 75 epochs with

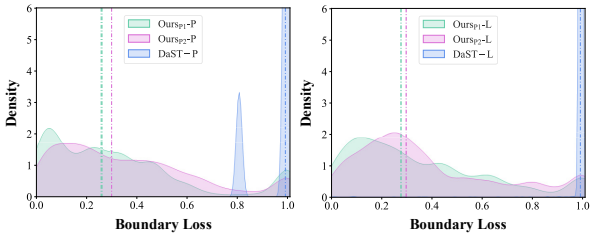


Figure 6. **Kernel Density Estimation (KDE) curves for the distribution of boundary loss on CIFAR-10 dataset [21].** Here, we leverage the Gaussian kernel to draw the curves. The dashed line represents the median boundary loss.

the learning rate decay from 37-th epoch to zero. The results in this part are epitomized in Tab. 6. We can see that various architectures indeed have different imitation abilities. In most cases, deeper networks can provide more powerful performance than shallower ones.

Statistical Analyses. We now provide the statistical results of the synthesized samples by our method and DaST [58]. First, we calculate the distribution for the boundary loss of the synthesized data and plot the Kernel Density Estimation (KDE) curves of the distribution of boundary loss. The results on CIFAR-10 dataset [21] are summarized in Fig. 6. From Fig. 6 we can see that in terms of the median boundary loss, the data synthesized by our medthod can outperform those synthesized by DaST. That is to say, the synthesized data of our method does have larger inter-class similarity than DaST [58]. Besides, the medians of our method equipped with Places365 [57] are lower than CUB-200 [48]. Together with Fig. 2, Tabs. 2 and 3, the importance of improving the inter-class similarity comes back clearly.

Qualitative Analyses. Besides the statistical analyses, we also visualize the synthesized data in three classes to see the difference of intra-class diversity between our method and DaST [58]. As shown in Fig. 7, the synthesized data of our methods with arbitrary proxy dataset indeed have a larger intra-class diversity than DaST, *i.e.*, wider distribution range (zoom-in to see the range of coordinates) and more sparse distribution density. This provides a powerful varification for the effectiveness of Eq. (5). Also, we can see the intra-class diversity of our method equipped with Places365 is still better than that equipped with CUB-200. Associating Fig. 6 with Fig. 7, we think the reason why the performance of our method equipped with Places365 can outperform the one equipped with CUB-200 is that both the intra-class diversity and the inter-class similarity of the synthesized data are larger for Places365 than CUB-200. The other qualitative results are available in Appendix C.3.

5. Conclusions and Discussions

In this paper, we first illustrated empirically that the effective data for surrogate training may have large inter-class

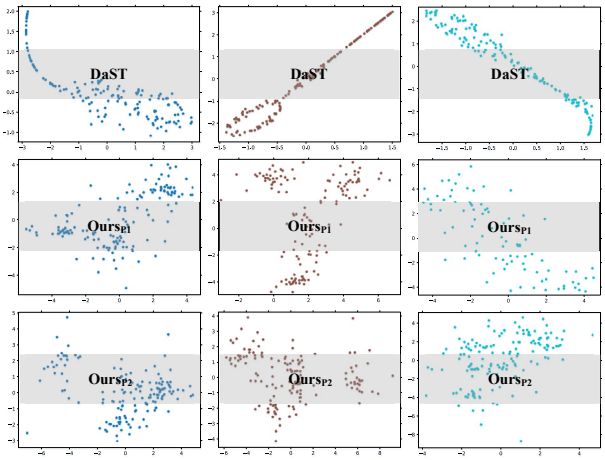


Figure 7. **Visualization of the generated data in 3 classes (distinguished by different colors) via t-SNE [45].** Each column represents one class, and each row denotes one method. Zoom-in for better view of details.

similarity instead of the inter-class diversity. Based on this assumption, we introduced a loss for the generator to enlarge the inter-class similarity. Secondly, unlike existing methods that enlarge the intra-class diversity in a model-agnostic way, we designed a new loss function to enhance the intra-class diversity explicitly in a model-specific way. Finally, based on careful observations, we proposed a triple-player framework to mine the great potential of the proxy data. With the proposed framework, our method can maintain its efficiency even when there are almost no semantic overlaps between the training data and the scarce proxy data (*i.e.*, with the number of the proxy images between 1k and 5k). According to extensive evaluations, the effectiveness of our method can be greatly verified.

Besides the achievements of this paper, there are still some challenges to be solved. For instance, first, in a more practical setting, in which the proxy dataset is often the mixed one where the samples are randomly chosen from different datasets, and there are few semantic overlaps among these datasets. In this scenario, can we achieve more stable performance? Besides, what is the lower bound in terms of the number of proxy data that our method allows? Moreover, from the perspective of the optimization for the generator, given the determined data distribution built by the fixed proxy samples, can we design a more efficient way to search for the valid data within a lower query regime? We will leave them to our future work.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China under Grants 62136007 and U20B2065, and in part by the Shaanxi Science Foundation for Distinguished Young Scholars under Grant 2021JC-16.

References

[1] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In *Proc. Int. Conf. Learn. Represent.*, 2018. 1

[2] Thomas Brunner, Frederik Diehl, Michael Truong Le, and Alois Knoll. Guessing smart: Biased sampling for efficient black-box adversarial attacks. In *Proc. Int. Conf. Comput. Vis.*, pages 4958–4966, 2019. 1

[3] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *IEEE Symp. Security Privacy*, pages 39–57. IEEE, 2017. 1, 2, 3, 5, 7

[4] Jianbo Chen, Michael I Jordan, and Martin J Wainwright. Hopskipjumpattack: A query-efficient decision-based attack. In *IEEE Symp. Security Privacy*, pages 1277–1294. IEEE, 2020. 1

[5] Pin-Yu Chen, Huan Zhang, Yash Sharma, Jinfeng Yi, and Cho-Jui Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *ACM AISec*, pages 15–26, 2017. 1

[6] Minhao Cheng, Simranjit Singh, Patrick H. Chen, Pin-Yu Chen, Sijia Liu, and Cho-Jui Hsieh. Sign-opt: A query-efficient hard-label adversarial attack. In *Proc. Int. Conf. Learn. Represent.*, 2020. 1

[7] Tarin Clanuwat, Mikel Bober-Irizar, Asanobu Kitamoto, Alex Lamb, Kazuaki Yamamoto, and David Ha. Deep learning for classical japanese literature. *arXiv:1812.01718*, 2018. 5

[8] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and Andre Van Schaik. Emnist: Extending mnist to handwritten letters. In *Int. Joint Conf. Neural Netw.*, pages 2921–2926. IEEE, 2017. 5

[9] Hadi M Dolatabadi, Sarah Erfani, and Christopher Leckie. Advflow: Inconspicuous black-box adversarial attacks using normalizing flows. In *Proc. Adv. Neural Inform. Process. Syst.*, 2020. 1

[10] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 9185–9193, 2018. 2

[11] Yinpeng Dong, Tianyu Pang, Hang Su, and Jun Zhu. Evading defenses to transferable adversarial examples by translation-invariant attacks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4312–4321, 2019. 2

[12] Ranjie Duan, Yuefeng Chen, Dantong Niu, Yun Yang, AK Qin, and Yuan He. Advdrop: Adversarial attack to dnns by dropping information. In *Proc. Int. Conf. Comput. Vis.*, pages 7506–7515, 2021. 2

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In *Proc. Adv. Neural Inform. Process. Syst.*, 2014. 2, 3

[14] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv:1412.6572*, 2014. 1, 2, 5, 7

[15] Chuan Guo, Jacob Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Weinberger. Simple black-box adversarial attacks. In *Proc. Int. Conf. Mach. Learn.*, pages 2484–2493. PMLR, 2019. 1

[16] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. Int. Conf. Comput. Vis.*, pages 770–778, 2016. 5, 6, 7

[17] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv:1503.02531*, 2015. 2

[18] Nathan Inkawhich, Kevin J Liang, Lawrence Carin, and Yiran Chen. Transferable perturbations of deep feature distributions. In *Proc. Int. Conf. Learn. Represent.*, 2020. 2

[19] Nathan Inkawhich, Kevin J Liang, Binghui Wang, Matthew Inkawhich, Lawrence Carin, and Yiran Chen. Perturbing across the feature hierarchy to improve standard and strict blackbox attack transferability. In *Proc. Adv. Neural Inform. Process. Syst.*, 2020. 2

[20] Nathan Inkawhich, Wei Wen, Hai Helen Li, and Yiran Chen. Feature space perturbations yield more transferable adversarial examples. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 7066–7074, 2019. 2

[21] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. *Tech Report*, 2009. 5, 6, 7, 8

[22] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Proc. Adv. Neural Inform. Process. Syst.*, volume 25, pages 1097–1105, 2012. 5, 6

[23] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv:1607.02533*, 2016. 1, 2, 5, 7

[24] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. In *Proc. IEEE*, pages 2278–2324, 1998. 5, 6, 7

[25] Jiadong Lin, Chuanbiao Song, Kun He, Liwei Wang, and John E Hopcroft. Nesterov accelerated gradient and scale invariance for adversarial attacks. In *Proc. Int. Conf. Learn. Represent.*, 2020. 2

[26] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *Proc. Int. Conf. Learn. Represent.*, 2018. 1, 2, 5, 7

[27] Thibault Maho, Teddy Furon, and Erwan Le Merrer. Surfree: a fast surrogate-free black-box attack. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 10430–10439, 2021. 1

[28] Xudong Mao, Qing Li, Haoran Xie, Raymond YK Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. In *Proc. Int. Conf. Comput. Vis.*, pages 2794–2802, 2017. 5

[29] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Proc. Adv. Neural Inform. Process. Syst.*, pages 3111–3119, 2013. 3

[30] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *arXiv:1411.1784*, 2014. 2, 3

[31] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2574–2582, 2016. 1, 2

[32] Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4467–4477, 2017. 1

[33] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Knockoff nets: Stealing functionality of black-box models. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4954–4963, 2019. 1, 2, 5, 6

[34] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *ACM Asia Conf. Comput. Commun. Security*, pages 506–519, 2017. 1, 2

[35] Alec Radford, Luke Metz, and Soumith Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *Proc. Int. Conf. Learn. Represent.*, 2016. 2

[36] Ali Rahmati, Seyed-Mohsen Moosavi-Dezfooli, Pascal Frossard, and Huaiyu Dai. Geoda: a geometric framework for black-box adversarial attacks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8446–8455, 2020. 1

[37] Binxin Ru, Adam Cobb, Arno Blaas, and Yarin Gal. Bayesopt adversarial attack. In *Proc. Int. Conf. Learn. Represent.*, 2019. 1

[38] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. In *Int. J. Comput. Vis.*, pages 211–252, 2015. 5, 6

[39] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. In *Proc. Adv. Neural Inform. Process. Syst.*, volume 29, pages 2234–2242, 2016. 3

[40] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4510–4520, 2018. 5, 7

[41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. Int. Conf. Learn. Represent.*, 2015. 5, 6, 7

[42] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1–9, 2015. 5, 7

[43] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv:1312.6199*, 2013. 1

[44] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *Proc. USENIX Conf. Security Symp.*, pages 601–618, 2016. 2

[45] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *J. Mach. Learn. Res.*, 9(11), 2008. 8

[46] Wenxuan Wang, Bangjie Yin, Taiping Yao, Li Zhang, Yanwei Fu, Shouhong Ding, Jilin Li, Feiyue Huang, and Xiangyang Xue. Delving into data: Effectively substitute training for black-box attack. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 4761–4770, 2021. 1, 2, 3, 5, 6, 7

[47] Xiaosen Wang and Kun He. Enhancing the transferability of adversarial attacks through variance tuning. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1924–1933, 2021. 2

[48] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Serge Belongie, and Pietro Perona. Caltech-ucsd birds 200. 2010. 5, 7, 8

[49] Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey, and Xingjun Ma. Skip connections matter: On the transferability of adversarial examples generated with resnets. In *Proc. Int. Conf. Learn. Represent.*, 2020. 2

[50] Cihang Xie, Zhishuai Zhang, Yuyin Zhou, Song Bai, Jianyu Wang, Zhou Ren, and Alan L Yuille. Improving transferability of adversarial examples with input diversity. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 2730–2739, 2019. 2

[51] Qiuling Xu, Guanhong Tao, Siyuan Cheng, and Xiangyu Zhang. Towards feature space adversarial attack. *arXiv:2004.12385*, 2020. 2

[52] Qiuling Xu, Guanhong Tao, Siyuan Cheng, and Xiangyu Zhang. Towards feature space adversarial attack by style perturbation. In *AAAI*, volume 35, pages 10523–10531, 2021. 2

[53] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 8715–8724, 2020. 3

[54] Jaemin Yoo, Minyong Cho, Taebum Kim, and U Kang. Knowledge extraction with no observable data. In *Proc. Adv. Neural Inform. Process. Syst.*, pages 2705–2714, 2019. 3

[55] Zhengyu Zhao, Zhuoran Liu, and Martha Larson. Adversarial robustness against image color transformation within parametric filter space. *arXiv:2011.06690*, 2020. 2

[56] Zhengyu Zhao, Zhuoran Liu, and Martha Larson. Towards large yet imperceptible adversarial image perturbations with perceptual color distance. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 1039–1048, 2020. 2

[57] Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(6):1452–1464, 2017. 5, 6, 7, 8

[58] Mingyi Zhou, Jing Wu, Yipeng Liu, Shuaicheng Liu, and Ce Zhu. Dast: Data-free substitute training for adversarial attacks. In *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, pages 234–243, 2020. 1, 2, 3, 5, 6, 8

[59] Yuankun Zhu, Yueqiang Cheng, Husheng Zhou, and Yantao Lu. Hermes attack: Steal dnn models with lossless inference accuracy. In *Proc. USENIX Conf. Security Symp.*, 2021. 2