

# DST: Dynamic Substitute Training for Data-free Black-box Attack

Wenxuan Wang Xuelin Qian<sup>†</sup> Yanwei Fu Xiangyang Xue  
 Fudan University  
 {wxwang19,xlqian,yanweifu,xyxue}@fudan.edu.cn

## Abstract

With the wide applications of deep neural network models in various computer vision tasks, more and more works study the model vulnerability to adversarial examples. For data-free black box attack scenario, existing methods are inspired by the knowledge distillation, and thus usually train a substitute model to learn knowledge from the target model using generated data as input. However, the substitute model always has a static network structure, which limits the attack ability for various target models and tasks. In this paper, we propose a novel dynamic substitute training attack method to encourage substitute model to learn better and faster from the target model. Specifically, a dynamic substitute structure learning strategy is proposed to adaptively generate optimal substitute model structure via a dynamic gate according to different target models and tasks. Moreover, we introduce a task-driven graph-based structure information learning constrain to improve the quality of generated training data, and facilitate the substitute model learning structural relationships from the target model multiple outputs. Extensive experiments have been conducted to verify the efficacy of the proposed attack method, which can achieve better performance compared with the state-of-the-art competitors on several datasets. Project page: <https://wxwangiris.github.io/DST>

## 1. Introduction

Deep neural network models have achieved the state of the art performances in many challenging computer-vision tasks [9, 16, 29]. These models have been wide-spread adopted in real-world applications, *e.g.*, self-driving cars, license plate reading, disease diagnosis from medical images, and activity classification. However, recent studies [8, 30] show that deep neural networks are highly vulnerable to adversarial examples, which contain small and imperceptible perturbations crafted to fool the target models. This attracts more researchers to study the attack and defense for better

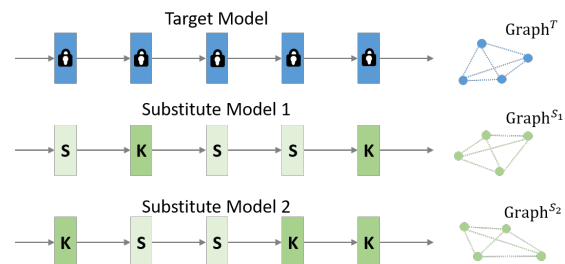


Figure 1. A conceptual overview of our method. Rather than retain all blocks, our approach learns to generate optimal substitute structure according to the black-box attack target. The lighter green blocks with ‘S’ indicate the skipped blocks, and dark green blocks with ‘K’ are the keeping blocks. The graph-based structural information is used to facilitate substitute training.

assessing and improving the robustness of deep models.

Adversarial attack methods can be categorized into two main settings, *i.e.*, white-box attack [2, 5, 17, 21, 24] and black-box attack [3, 3, 4, 6, 7, 12, 13, 35], by whether or not the attackers can have full access to the structure and parameters of the target model. Nowadays, in the era of big data, data is one of the most valuable assets for companies, and much of it also has privacy issues. Thus, in practical cases, it is not only difficult for attackers to know the details of the target model, but also hardly obtain the training data for the target model, even the number of categories.

The purpose of this paper is to successfully achieve a data-free black-box attack according to the given target model. “Data-free” suggests that we cannot access any knowledge about the data distribution (*e.g.*, the type of data, number of categories, *etc.*), which was used for the target model; “Black-box” indicates the target model structure is completely shielded from the attackers, resulting in the limitation on getting model parameters or features of middle layers. The only available thing we can use is the output of probabilities/labels from the target model. Such a strict setting is more in line with the requirements of real-world scenarios, especially at a time when privacy data protection has attracted more attention. Inspired by the substitute training methods in black-box attack, many works [32, 38]

<sup>†</sup>indicates corresponding author.

try to tackle the data-free black-attack through learning a substitute model for the target one with generated training data. However, there exist two main limitations in existing methods, (1) *static substitute model structure for different targets*: due to the lack of prior knowledge, utilizing the same and static substitute model architecture for various target models or tasks definitely can not achieve powerful attack. Meanwhile, it is impractical and cost expensive to train multiple models to find the most suitable substitute model structure for each target variation. (2) *Assumption of knowing the number of categories for the target model*: for the total data-free black-box attack, it is unreasonable for knowing the number of training data classes for the target model. Thus, using the label as the generator guidance information to synthesize diverse and label-controlled data for substitute training is unpractical.

In this study, to address the limitations of existing methods, we propose a novel and task-driven **Dynamic Substitute Training (DST)** attack method for data-free black-box attacking, as illustrated in Fig. 1. Our DST attack adopts the basic substitute learning framework as in [14, 32, 38], which generates the training data via a generator with noise as input, and takes advantage of the knowledge distillation concept to encourage the substitute model to have the same output as the target one when facing the same synthesized training image. To tackle the constant substitute model architecture problem (limitation (1)), in our DST attack algorithm, for the first time, we introduce a dynamic substitute structure learning strategy to automatically generate a more suitable substitute model structure according to different target models and tasks. To achieve such dynamic structure generation, we specially design a learnable dynamic gate to determine which blocks in the deep architecture can be skipped. To deal with being unaware of the prior knowledge about the training data classes issue (limitation (2)), we introduce a graph-based structural information learning strategy in DST, to further improve the generator performance and enhance the substitute training process. Such a learning strategy can facilitate the substitute model learning more implicit and detailed information from the structural relationship among multiple target model outputs. Meanwhile, such structural information can reflect the representation distance among a group of generated training data and stimulate the generator to deliver more valuable training data. Overall, our DST attack can adaptively generate optimal substitute model structure for various targets, improve the consistency between the substitute and target model, and encourage the generator to synthesize better training data via learning structural information. This can promote the data-free black-box attack performance.

The main contributions of this work are summarized below, (1) We propose a novel and task-driven dynamic substitute training attack method to boost the data-free black-box

attacking performance. (2) For the first time, we introduce a dynamic substitute structure learning strategy to adaptively generate optimal substitute model architecture according to the different target models and tasks, instead of adopting the same and static network. (3) To encourage the substitute model to learn more details from the target model and improve the quality of generated training data, we propose a graph-based structural information learning strategy to deeply explore the structural, relational, and valuable information from a bunch of target outputs. (4) The comprehensive experiments over four public datasets and one online machine learning platform demonstrate that our DST method can achieve SOTA attack performance and significantly reduce the query times during the substitute training.

## 2. Related Work

**Adversarial Attack.** Since deep learning models have achieved remarkable success on most computer vision tasks [9, 16, 25, 26, 29, 31], the study for the security of these models has attracted many researchers. [30] illustrates that deep neural networks are susceptible to adversarial perturbations. Subsequently, more and more works [2–5, 7, 8, 11, 17, 19, 21, 24, 34] focus on the adversarial example generation task. In general, the attack task can be divided into white-box and black-box attacks, the former one can know the knowledge of the structure and parameters of the target model, and the latter one only has the access to the simple output of the target. Most white-box algorithms [2, 5, 8, 17, 21, 24] generate adversarial examples based on the gradient of loss function with respect to the inputs. For the black-box attack, some methods [3, 4, 7] iteratively query the outputs of target model and estimate the gradient of target model via training a substitute model; and others [11, 19, 34] focus on improving the transferability of adversarial examples across different models. In this work, we focus on the more practical and challenging scenario, *i.e.*, the data-free black-box attack, which attacks the black-box target model without the need for any real data samples.

**Data-free Black-box Attack.** In practice, attackers can hardly obtain the training data for target model, even the number of categories. Thus, some works [10, 14, 22, 32, 36, 38] begin to study the data-free black-box attack task, which generate adversarial examples without any knowledge about the training data distribution. Mopuri *et al.* [22] propose an attack to corrupt the extracted features at multiple layers to realize independent of the underlying task. Huan *et al.* [10] learn adversarial perturbations based on a mapping connection between fine-tuned model and target model. [32, 38] utilize a generator with noise as input to synthesize data for training a substitute model to learn information from the target one via knowledge distillation. In this paper, we adopt the basic framework as [14, 32, 38]. Different from these works, the structure of our substitute

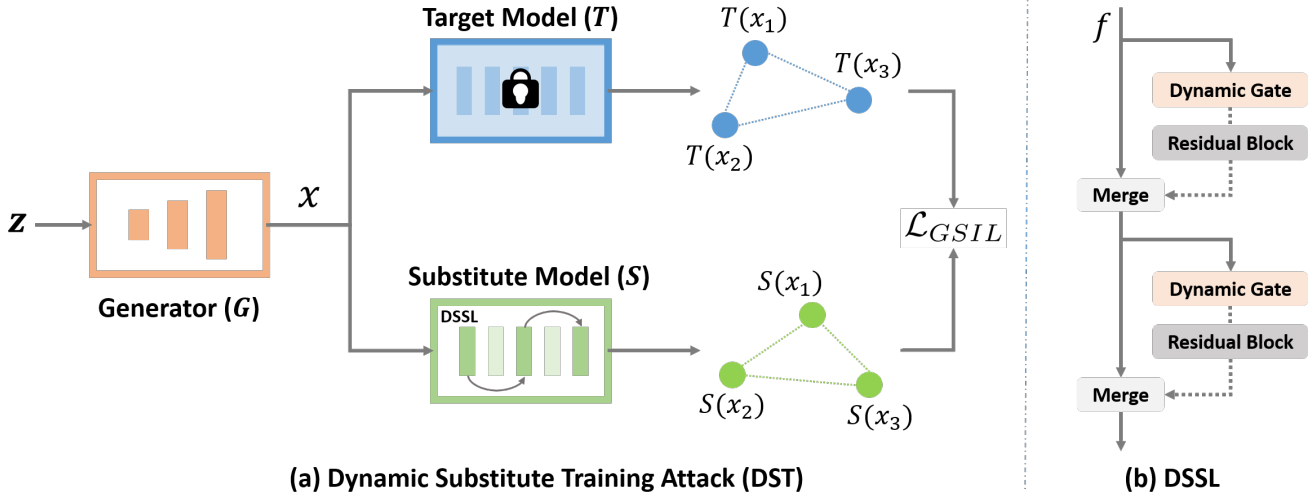


Figure 2. (a) Illustration of our **Dynamic Substitute Training** attack framework (DST). DST utilizes a **Graph-based Structural Information Learning** constrain  $\mathcal{L}_{GSIL}$  to train the generator and substitute model. (b) Schematic diagram of **Dynamic Substitute Structure Learning** strategy (DSSL). The DSSL of DST is applied to automatically generate optimal substitute model structure according to different targets.

model is not fixed, but dynamically generated and optimized according to the different target models and various datasets. Moreover, we train the substitute model and generator not only from a single output of the target model, but also from the detailed and implicit information represented in the graph-based relationship of multiple target outputs.

### 3. Methodology

#### 3.1. Framework Overview

Figure 2(a) illustrates the schematic of our proposed unified **Dynamic Substitute Training** attack framework (DST), which mainly consists of two learnable components, *i.e.*, a generator  $\mathcal{G}$  and a substitute model  $\mathcal{S}$ . More precisely, DST employs a generator  $\mathcal{G}$  to synthesize training samples with Gaussian random noise  $z \sim \mathcal{N}(0, 1)$ ,

$$x = \mathcal{G}(z) \in \mathbb{R}^{3 \times h \times w} \quad (1)$$

where  $h$  and  $w$  denote the height and width of the generated training samples. Subsequently, we feed the synthesized data into a target model  $\mathcal{T}$  and a substitute model  $\mathcal{S}$  simultaneously. The teacher-student strategy is re-purposed here to encourage  $\mathcal{S}$  to learn as similar decision boundary as  $\mathcal{T}$ ,

$$\mathcal{L}_S = d(\mathcal{T}(x), \mathcal{S}(x)) \quad (2)$$

where  $d$  denotes a metric function to measure the distance between the outputs from  $\mathcal{T}$  and  $\mathcal{S}$ . After such substitute training process, attacks can be conducted on the well-trained  $\mathcal{S}$ , and then transferred to  $\mathcal{T}$ .

As proposed in [32, 38], the  $\mathcal{S}$  aims to minimize output discrepancy with the  $\mathcal{T}$ , while the  $\mathcal{G}$  tries to maximize the

discrepancy to explore various hard samples for substitute training. For model parameters learning via gradient descent, such objective maximizing function for  $\mathcal{G}$  learning can be transferred to minimize the loss as the following,

$$\mathcal{L}_G = -d(\mathcal{T}(x), \mathcal{S}(x)) \quad (3)$$

Overall, the main contributions of this paper concentrate on the learning of substitute model. Remarkably, we for the first time present a dynamic substitute searching strategy to learn an optimal structure for the substitute model, rather than manually select one according to priors [14, 32, 38]. We argue that such a design can best ‘mimic’ the characteristics of the target model, both from the structure of the model and the spatial distribution of parameters, which is more flexible, practical, and significant. Furthermore, in order to stimulate the substitute model to learn more details from the target and improve the quality of generated training data, we propose a graph-based structural information learning strategy to deep explore more structural, relational, and valuable information from multiple target outputs.

#### 3.2. Dynamic Substitute Structure Learning

Compared with static network architectures, dynamic network structure usually has the superiority in network capacity to various tasks and targets. In our black-box attack task, according to different target models and various datasets, our **Dynamic Substitute Structural Learning** strategy (DSSL) can adaptively find more reasonable substitute model architecture to achieve more powerful attack ability.

Considering that most deep networks have adopted a block-based residual learning design following the remark-

able success of ResNet [9] (e.g., MobileNetV2 [28], ShuffleNet [37], and ResNext [33]), we, therefore, construct our DSSL based on the residual design to be generally applicable in the common deep networks.

As shown in Fig. 2(b), the selecting probability of each path is generated by a *dynamic gate*. Such dynamic gate aims at predicting a one-hot vector, which denotes whether to execute or skip the branch of residual block. Here, we adapt a set of light-weight operations to realize the dynamic gate function  $\mathcal{DG}(\cdot)$  with the feature  $f$  as the input,

$$\mathcal{DG}(f) = \mathcal{H}(WP(f) + b) \quad (4)$$

where  $P(\cdot)$  means the global average pooling layer, and  $W$  and  $b$  are the fully-connected layer parameters. To realize discrete binary decisions for the path chosen, we choose hard sigmoid function as  $\mathcal{H}(\cdot)$ , which can be defined as,

$$\mathcal{H}(g) = \max\left(0, \min\left(kg + \frac{1}{2}, 1\right)\right) \quad (5)$$

where we set the threshold as 0.5, which clips the outputs of  $\mathcal{H}(\cdot)$  to 0 or 1. The  $k$  is the crucial parameter used as an approximation of step function that emits binary decisions.

### 3.3. Graph-based Structural Information Learning

To constrain the consistency of the outputs between the  $\mathcal{S}$  and  $\mathcal{T}$  and improve the quality of generated training data by  $\mathcal{G}$ , we argue that it is important to explore the implicit structural relationships among different outputs. The graph-based relational representation can reflect the deep knowledge of  $\mathcal{T}$ , and learning such structural features can help  $\mathcal{S}$  realize a more similar decision boundary as  $\mathcal{T}$  to further improve the attack performance. Meanwhile, such structural relationships can reflect the distance among the multiple generated training images, and improve the data quality generated by  $\mathcal{G}$  in return.

During the training, we propose a novel Graph-based Structural Information Learning strategy (GSIL) to indict the underlying relationships of the model outputs. By virtue of the graph network conception, in each mini-batch during training, the nodes of graph are the model outputs according to inputs, and the corresponding edges are the relation formulated as an adjacent matrix among these nodes. Specially, the structural information graph is defined as,

$$\begin{aligned} \text{Graph} &= (\text{nodes}, \text{edges}) = (\{x_j\}_{j=1}^B, A) \\ A(j, k) &= \|x_j - x_k\|_E, \quad j, k = 1, \dots, B \end{aligned} \quad (6)$$

where  $B$  means the number of training data in each training iteration, *i.e.*, the value of the mini-batch, and the edge is defined as the Euclidean distance  $E$  between the two node representations / model outputs.

Once we obtain the structural information graph, the corresponding constrain  $\mathcal{L}_{GSIL}$  can be produced to further restrict the discrepancy between the  $\text{Graph}^{\mathcal{S}}$  of  $\mathcal{S}$  and the

$\text{Graph}^{\mathcal{T}}$  of  $\mathcal{T}$ . The difference of such graphs contains the node discrepancy and edge discrepancy formulated as,

$$\begin{aligned} \mathcal{L}_{GSIL} &= \text{Disc}(\text{Graph}^{\mathcal{T}}, \text{Graph}^{\mathcal{S}}) \\ &= \alpha_1 \sum_{j=0}^B \cdot \text{KL}(x_j^{\mathcal{T}}, x_j^{\mathcal{S}}) + \alpha_2 \cdot \text{MSE}(A^{\mathcal{T}}, A^{\mathcal{S}}) \end{aligned} \quad (7)$$

where  $x_j^{\mathcal{T}}, x_j^{\mathcal{S}}$  and  $A^{\mathcal{T}}, A^{\mathcal{S}}$  refer to the node and edge set for  $\mathcal{S}$  and  $\mathcal{T}$ . The Kullback-Leibler Divergence function  $\text{KL}$  is used to measure the discrepancy between two nodes, and the MSE loss is applied to restrict the differences between the two edges. And  $\alpha_1$  and  $\alpha_2$  are the hyper-parameters to balance the discrepancy of the nodes and edges. In DST model learning, we apply the  $\mathcal{L}_{GSIL}$  to measure the discrepancy between the  $\mathcal{S}$  and  $\mathcal{T}$  used in Eq. 2 and Eq. 3.

## 4. Experiment

### 4.1. Experiment Setup

**Datasets and model structure.** 1) MNIST [18]: The target model is pre-trained on AlexNet [16], VGG-16 [29], and ResNet-18 [9]. 2) CIFAR-10 [15]: The target model is pre-trained on AlexNet, VGG-16, and ResNet-18. 3) CIFAR-100 [15]: The target model is pre-trained on VGG-19 and ResNet-50. 4) Tiny Imagenet [27]: The target model is pre-trained on ResNet-50. For these four datasets, the default substitute model basic structure is ResNet-34.

**Competitors.** To verify the efficacy of our DST, we compare our attacking results with the existing state-of-the-art data-free black-box attacks, *i.e.*, GD-UAP [22], Cosine-UAP [36], DaST [38], MAZE [14], and DDG+AST [32].

**Implementation details.** We use Pytorch for Implementation. We utilize Adam to train our substitute model and generator from scratch, and all weights are randomly initialized using a truncated normal distribution with std of 0.02. The initial learning rates of the generator and substitute model are set as 0.0001 and 0.001, respectively, they are gradually decreased to zero from the 80th epoch, and stopped at the 150th epoch. We set the mini-batch size as 500, the hyper-parameters  $\alpha_1$  and  $\alpha_2$  are equally as 1. The  $k$  in Eq. 5 is set as 1 in the following experiments. Our model is trained by one NVIDIA GeForce GTX 1080Ti GPU. We apply PGD [21] as the default white-box attack method to generate adversarial images over the well-trained substitute model during the evaluation. We also utilize several classic attack methods for extensive experiments, such as FGSM [8], BIM [17] and C&W [2]. During the model optimizing stage, there are no real images used for model learning, only random noise is utilized as input. For evaluation, the adversarial samples to conduct attack crafted only on test set over four datasets.

**Evaluation metrics.** Following the two scenarios proposed in DaST [38], *i.e.*, only getting the output label from the

	Dataset	MNIST			CIFAR-10			CIFAR-100		Tiny ImageNet
	Target Model	AlexNet	VGG-16	ResNet-18	AlexNet	VGG-16	ResNet-18	VGG-19	ResNet-50	ResNet-50
Non-Target	GD-UAP [22]	33.28	29.54	30.81	18.39	16.43	20.65	12.57	14.90	8.93
	Cosine-UAP [36]	38.92	35.11	28.48	38.20	23.44	35.73	15.62	17.83	11.96
	DaST [38]	63.34	60.38	56.21	43.64	56.25	49.36	32.94	27.32	26.85
	MAZE [14]	65.82	67.68	59.32	44.72	50.13	52.99	29.41	24.83	25.40
	DDG+AST [32]	68.29	65.03	61.47	44.87	53.91	50.30	31.88	26.56	30.81
	DST (Ours)	<b>70.48</b>	<b>72.49</b>	<b>63.72</b>	<b>47.20</b>	<b>58.21</b>	<b>54.93</b>	<b>34.03</b>	<b>31.39</b>	<b>32.28</b>
Target	GD-UAP [22]	28.10	39.51	29.48	14.22	18.43	16.49	10.55	6.31	7.50
	Cosine-UAP [36]	36.91	48.23	37.88	20.46	17.97	24.31	12.40	12.11	10.53
	DaST [38]	58.28	67.33	54.29	29.48	40.29	46.10	15.82	22.48	20.37
	MAZE [14]	60.48	67.39	60.43	33.28	29.83	41.26	18.22	20.21	19.25
	DDG+AST [32]	62.20	66.45	61.94	35.91	34.87	45.25	17.04	19.57	17.48
	DST (Ours)	<b>64.82</b>	<b>68.49</b>	<b>65.77</b>	<b>38.29</b>	<b>44.71</b>	<b>47.90</b>	<b>20.59</b>	<b>23.01</b>	<b>22.94</b>

Table 1. Comparing ASRs results using probability as the target model output among our method and competitors over four datasets. For a fair comparison, we use PGD as the attack method and ResNet-34 as the default substitute model for all substitute training.

	Dataset	MNIST				CIFAR-10				CIFAR-100			
	Attack Method	FGSM	BIM	PGD	C&W	FGSM	BIM	PGD	C&W	FGSM	BIM	PGD	C&W
Non-Target	DaST [38]	59.32	81.52	63.34	59.37	42.41	53.92	56.25	57.28	27.48	34.56	27.32	25.37
	MAZE [14]	56.25	74.83	65.82	70.44	46.57	62.91	50.13	48.12	27.86	30.51	24.83	27.66
	DDG+AST [32]	62.48	76.70	68.29	69.33	44.12	59.03	53.91	56.28	30.79	33.62	26.56	25.83
	DST (Ours)	<b>63.92</b>	<b>82.45</b>	<b>70.48</b>	<b>73.22</b>	<b>49.21</b>	<b>64.90</b>	<b>58.21</b>	<b>59.17</b>	<b>33.80</b>	<b>37.74</b>	<b>31.39</b>	<b>29.33</b>
	Target	DaST [38]	59.28	70.47	58.28	59.33	30.24	45.10	40.29	42.13	16.49	<b>27.86</b>	22.48
	MAZE [14]	61.42	67.29	60.48	57.32	26.35	40.49	29.83	38.10	20.42	24.81	20.21	23.85
	DDG+AST [32]	57.84	71.90	62.20	52.11	37.58	42.06	34.87	45.39	17.82	26.33	19.57	28.95
	DST (Ours)	<b>64.23</b>	<b>73.85</b>	<b>64.82</b>	<b>60.57</b>	<b>39.53</b>	<b>47.20</b>	<b>44.71</b>	<b>48.06</b>	<b>20.48</b>	27.31	<b>23.01</b>	<b>29.57</b>

Table 2. Comparing ASRs results using probability as the target model output among our method and competitors with various white-box adversarial example generation methods. For a fair comparison, we utilize ResNet-34 as the substitute model for all substitute training. The target models are the AlexNet for MNIST, VGG-16 for CIFAR-10, and ResNet-50 for CIFAR-100.

target model and accessing the output probability well, we name these two scenarios as Probability-based and Label-based. In the experiments, we report the attack success rates (ASRs) of the adversarial examples generated by the substitute model to attack the target model. As in DaST [38], for non-target attacking, we only attack the images classified correctly by the target model. For target attacking, we only generate adversarial examples on the images which are not classified to the specific wrong labels. We conduct ten times over each testing, and report the average results.

## 4.2. Black-box Attack Results

**Comparisons with the state-of-the-art attacks.** As shown in Tab. 1, our DST attack beats all competitors with significant margins. We conduct extensive comparisons with these competitors from several aspects, *i.e.*, the diverse tasks (datasets), various target models, and different attack goals (target/non-target). The results show that our DST method can achieve the best attack ability for black-box attack without using any real data as input.

**Comparisons with competitors using various white-box adversarial sample generation methods.** Under the

probability-based scenario, we compare the ASRs with competitors when using different white-box attack methods via well-trained substitute models. As illustrated in Tab. 2, we apply four classic attacks to generate adversarial samples over substitute models for attacking the target models on three datasets. Compared to other data-free black-box attack algorithms, our DST can dynamically generate suitable substitute model structure and learn structural information from the target output, thus, DST achieves the best ASRs over most experiments.

**Comparisons with competitors which use different deep networks as their substitute model.** As shown in Tab. 3, we compare our DST attack performance with competitors which can choose a bunch of networks as their substitute model. Even our DST can only use ResNet-34 as the basic substitute model, thanks to the dynamic substitute structure learning strategy, we can still achieve better attack performance when the competitors are able to choose the best substitute model from a set of different networks. We can also notice that the attack performance of the competitors is highly dependent on the chosen substitute model structure. Such results demonstrate that our DST can automatically

	Method	MNIST					CIFAR-10					CIFAR-100				
		V-16	V-19	R-18	R-34	R-50	V-16	V-19	R-18	R-34	R-50	V-16	V-19	R-18	R-34	R-50
Non-Target	DaST [38]	62.49	59.21	40.36	60.38	49.85	23.48	45.83	34.87	49.36	43.20	17.38	12.42	20.48	27.32	22.56
	MAZE [14]	57.36	69.31	64.20	67.68	55.47	36.88	29.42	38.47	52.99	50.35	13.27	23.49	19.21	24.83	25.55
	DDG+AST [32]	68.28	57.20	63.12	65.03	68.37	36.02	52.30	47.21	50.30	49.46	12.48	17.20	15.93	26.56	22.46
	DST (Ours)	<b>72.49</b>					<b>54.93</b>					<b>31.39</b>				
Target	DaST [38]	53.92	61.48	42.01	67.33	48.47	19.49	39.61	37.02	46.10	47.00	9.32	22.51	14.29	22.48	17.56
	MAZE [14]	54.21	67.50	57.86	67.39	64.23	42.93	35.32	37.48	41.26	36.06	12.41	21.48	18.60	20.21	20.35
	DDG+AST [32]	48.99	63.81	66.91	66.45	55.30	43.05	41.29	40.82	45.25	38.90	13.40	18.66	21.43	19.57	19.95
	DST (Ours)	<b>68.49</b>					<b>47.90</b>					<b>23.01</b>				

Table 3. ASRs comparisons among our DST and competitors using different substitute model structures under Probability-based scenario. The target model are VGG-16 for MNIST, ResNet-18 for CIFAR-10, and ResNet-50 for CIFAR-100. We use PGD as the attack method for all attacks. The competitors can use VGG-16 (V-16), VGG-19 (V-19), ResNet-18 (R-18), ResNet-34 (R-34), and ResNet-50 (R-50) as their substitute models, and the ASRs with blue represent the selected best substitute models according to different attack targets for each competitor. Our DST attack can only use ResNet-34 as the basic substitute model structure under all conditions.

	Method	Probability-based	Label-based
Non-Target	GD-UAP [22]	73.12	68.01
	Cosine-UAP [36]	77.48	82.56
	DaST [38]	92.48	94.21
	MAZE [14]	93.95	93.18
	DDG+AST [32]	93.29	95.83
	DST (Ours)	<b>95.02</b>	<b>96.44</b>
Target	GD-UAP [22]	42.19	31.47
	Cosine-UAP [36]	46.10	52.36
	DaST [38]	52.18	68.85
	MAZE [14]	50.23	59.83
	DDG+AST [32]	53.46	71.48
	DST (Ours)	<b>56.39</b>	<b>75.90</b>

Table 4. Comparing ASRs results among our proposed DST attack and competitors for attacking the online Microsoft Azure example model under both Probability- and Label-based scenarios. For a fair comparison, we use PGD as the attack method and ResNet-34 as the default substitute model for all substitute training.

generate optimal substitute model according to the target, which is crucial for the practical applications.

**Comparisons with state-of-the-art competitors against online Microsoft Azure example model.** The attack performance targeting real-world black-box model is vital to evaluate the adversarial example generation method. Thus, we compare our DST with competitors over the online Microsoft Azure model. As in Tab. 4, our DST outperforms all competitors, which indicates the practical black-box attack capacity of the proposed DST in real-world applications.

### 4.3. Ablation Study

To further explore the efficacy of components in our DST attack, we conduct extensive ablation studies over the following variants: (1) ‘Baseline-I’: using random noise as

	Components	Probability-based		Label-based	
		MNIST	C-100	MNIST	C-100
Non-Target	Baseline-I	30.82	12.40	15.33	7.54
	Baseline-II	48.99	21.36	23.47	14.22
	+ GSIL	59.32	24.81	29.65	20.23
	+ DSSL	<b>70.48</b>	<b>31.39</b>	<b>36.22</b>	<b>25.83</b>
Target	Baseline-I	27.49	10.48	16.72	5.83
	Baseline-II	38.21	18.49	23.95	8.49
	+ GSIL	52.40	21.48	26.58	13.77
	+ DSSL	<b>64.82</b>	<b>23.01</b>	<b>31.94</b>	<b>19.30</b>

Table 5. ASRs results of variants of the proposed DST attack method. The target model is based on AlexNet for MNIST, and ResNet-50 for CIFAR-100. ‘C-100’ refers to the CIFAR-100 dataset. We use PGD as the attack method and ResNet-34 as the substitute model for all experiments.

input to generate training data and applying MSE loss to constrain the outputs’ similarity between the  $\mathcal{S}$  and  $\mathcal{T}$ ; (2) ‘Baseline-II’: using random noise as input to generate training data and applying Kullback-Leibler Divergence to constrain the outputs’ similarity between the  $\mathcal{S}$  and  $\mathcal{T}$ ; (3) ‘+ GSIL’: using the graph-based structural information learning constrain to conduct substitute training; (4) ‘+ DSSL’: based on the ‘+ GSIL’, a dynamic substitute structure learning strategy is applied to adaptively generate suitable substitute model structure, and it is the whole DST attack model.

**The efficacy of the components for attack ability in DST attack.** As in Tab. 5, comparing the results among the variants, we can notice the following observations, (1) ‘Baseline-I’ can only realize basic attack ability against the target model. (2) Comparing the ‘Baseline-II’ with ‘Baseline-I’, it is obvious that changing the MSE loss to Kullback-Leibler Divergence constrain can better restrict the output similarity between the  $\mathcal{S}$  and  $\mathcal{T}$ . (3) Compar-

Dataset		MNIST			CIFAR-10			CIFAR-100	
Target Model	AlexNet	VGG-16	ResNet-18	AlexNet	VGG-16	ResNet-18	VGG-19	ResNet-50	
Non-Target	ResNet-18	59.41	49.82	48.24	35.67	54.28	43.09	12.49	14.23
	+ DSSL (Ours)	68.81(55.6)	73.59(50.0)	60.40(44.4)	50.91(38.9)	57.24(33.3)	55.96(22.2)	30.43(22.2)	27.98(11.1)
	ResNet-34	59.32	64.40	56.28	39.21	50.90	46.29	23.55	24.81
	+ DSSL (Ours)	70.48(70.6)	72.49(73.5)	63.72(55.9)	47.20(61.8)	58.21(55.9)	54.93(67.7)	34.03(55.9)	31.39(38.2)
	ResNet-50	47.80	63.36	50.19	38.42	43.28	50.12	29.65	17.33
	+ DSSL (Ours)	69.55(86.0)	73.42(80.0)	61.82(82.0)	50.37(78.0)	55.61(60.0)	53.57(70.0)	33.50(42.0)	32.40(54.0)
Non-Target	ResNet-101	49.08	58.21	38.17	32.54	54.83	49.22	19.34	23.46
	+ DSSL (Ours)	70.33(92.1)	74.02(87.1)	59.48(93.1)	49.53(88.1)	57.31(88.1)	56.24(79.2)	35.09(67.3)	30.47(77.2)
	ResNet-18	60.35	58.28	59.12	28.44	43.75	49.20	15.91	11.27
	+ DSSL (Ours)	62.94(55.6)	67.21(50.0)	63.38(44.4)	39.41(38.9)	46.82(33.3)	48.03(22.2)	16.23(22.2)	21.10(11.1)
	ResNet-34	52.40	60.22	55.94	32.51	35.94	39.42	14.09	21.48
	+ DSSL (Ours)	64.82(70.6)	68.49(73.5)	65.77(55.9)	38.29(61.8)	44.71(55.9)	47.90(67.7)	20.59(55.9)	23.01(38.2)
Non-Target	ResNet-50	54.28	63.47	60.02	33.00	41.25	35.91	17.32	23.99
	+ DSSL (Ours)	66.91(86.0)	70.36(80.0)	64.24(82.0)	37.58(78.0)	43.92(60.0)	49.50(70.0)	21.42(42.0)	25.84(54.0)
	ResNet-101	43.35	56.17	63.29	30.88	37.25	42.63	21.38	14.56
	+ DSSL (Ours)	64.27(92.1)	66.05(87.1)	64.88(93.1)	40.10(88.1)	41.29(88.1)	48.57(79.2)	22.50(67.3)	23.36(77.2)

Table 6. Comparing ASRs results using probability as the target output among our method and variants with different substitute networks. We use PGD as the default attack for all experiments. The ‘ResNet-18’, ‘ResNet-34’, ‘ResNet-50’, and ‘ResNet-101’ represent the basic substitute model with graph-based structural information learning module, and ‘+DSSL (ours)’ means the above raw model with dynamic substitute structure learning strategy. The numbers in ‘()’ denote skip rate (the percentage (%) of skip blocks in the substitute model).

ing the results between the ‘Baseline-II’ and ‘+ GSIL’, the effectiveness of the structural information learning strategy is proved. Specially, the distance of graph edges represent the outputs’ structural relationship, thus, learning such information can greatly encourage  $\mathcal{S}$  to learn more detailed knowledge from the  $\mathcal{T}$ . (4) With the ‘+DSSL’ module, the ASRs have been significantly improved. Such results illustrate that the static substitute structure does not benefit to various attack targets, and self-optimized substitute structure is more reasonable for unknown diverse attack targets.

### The efficacy of dynamic substitute structure learning strategy for different tasks and various target models.

To better verify the effectiveness of the proposed dynamic substitute structure learning strategy, we choose several basic deep learning networks, *i.e.*, ResNet-18, ResNet-34, ResNet-50, and ResNet-101, as the substitute models. For a fair comparison, compared to ‘+ DSSL (Ours)’ the basic substitute model also uses the graph-based structural information learning strategy. As shown in Tab. 6, we can draw the following conclusions, (1) By applying the dynamic substitute structure learning strategy, our method can achieve much better attack results over all tasks and target models, such as comparing the ‘ResNet-34’ with the next lower raw ‘+ DSSL (Ours)’. It is reasonable for different tasks and target models to have various corresponding structures as the best substitute models. Thus, our DSSL component adaptively generating more reasonable substitute structure can indeed promote the attack performance. (2) For the same target model over the same dataset, the ASRs of ‘+ DSSL (Ours)’ have little differences among various basic

substitute model architectures. For example, to attack the AlexNet model trained on MNIST, our ASRs are around 69% with small fluctuations based on four basic models under non-target attack setting. However, without applying the dynamic substitute structure learning strategy, the ASRs change dramatically from 47% to 59% according to the different substitute models. These results reinforce that our method can automatically produce optimal substitute structures over different basic models. (3) We also report the skip rate in the experiments, it is obvious that the number of skipped blocks is more when the basic substitute structure is complex with deeper structure, *e.g.*, the skip rate is usually larger for utilizing ResNet-101 as basic substitute model compared to the ResNet-18. And the skip rate is usually smaller when targeting more difficult task, *e.g.*, CIFAR-100 dataset, due to the substitute model keeping more blocks to learn more complex feature representations.

### The efficacy of each proposed component for reducing the number of query times during the substitute training.

Considering that in real-world applications, many online platforms defend themselves by detecting the number of query times for a single IP address, thus, more and more attack methods pay more attention to decreasing the number of query times. As illustrated in Tab. 7, we compare the number of query times during both the model learning and inference stage with competitors. (1) With a similar perturbation distance, our DST attack method not only achieves the comparable ASRs against others, but also needs the least query times during training and zero query times during attacking. As for the compared score-based and decision-

Method	ASRs	Distance	Train-Q	Test-Q
GLS [23]	53.28	3.68	-	311
Boundary [1]	99.32	3.94	-	702
DaST [38]	92.48	3.79	20M	-
MAZE [14]	93.95	3.90	30M	-
DDG+AST [32]	93.29	3.88	15M	-
Baseline-I	55.92	3.82	30M	-
Baseline-II	70.28	3.91	30M	-
+ GSIL	83.24	3.76	13M	-
+ DSSL	95.02	3.80	9M	-

Table 7. Comparison results among our DST method, variants of DST, and several competitors, targeting the Microsoft Azure example model and using BIM as the default non-target attack method. The ‘ASRs’ refers to the attack success rate, ‘Distance’ means the average perturbation distance per image, ‘Train-Q’ is the number of queries in the model learning stage, and ‘Test-Q’ denotes the query times during the evaluation.

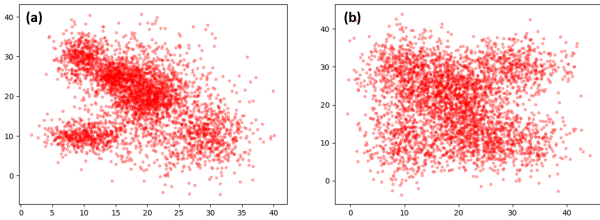


Figure 3. Visualization of 3,000 generated data distribution using t-SNE [20] on CIFAR-10. (a) Data generated by the  $\mathcal{G}$  of DST without applying graph-based structural information learning strategy. (b) Data generated by the  $\mathcal{G}$  of our DST.

based attacks, they need to query the target model to generate each attack in the evaluation stage, and feed the target model with the same original data numerous times which can be tracked easily by the defender. (2) With the structural information learning strategy, the number of training query times drops significantly, which states that the graph-based relationship among the outputs can help  $\mathcal{S}$  learn knowledge much quicker from the target one. (3) With the proposed dynamic substitute structure learning strategy, the number of training query times of our attack method declines to some extent, which explain that  $\mathcal{S}$  with optimal structure not only learn better but also learn faster from the  $\mathcal{T}$ .

**The efficacy of graph-based structural information learning strategy for the quality of generated training data.** In terms of feature, we visualize the feature distribution of synthesized data extracted by the target model in Fig. 3. Such qualitative results show that, with applying the proposed graph-based structural information learning strategy, the generated data is more evenly distributed, which is friendly for substitute model training.

**The visualizations of dynamic substitute model structures according to different target models and tasks.**

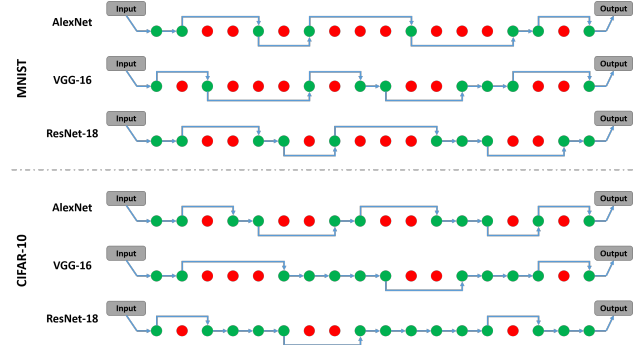


Figure 4. The visualizations of the dynamic substitute model structures according to different target models (*i.e.*, AlexNet, VGG-16, and ResNet-18) and tasks (*i.e.*, MNIST and CIFAR-10). The basic network of substitute models is ResNet-18. The green dot means the keeping block, the red dot represents the skipped block.

As shown in Fig. 4, for the same basic substitute model architecture, the optimal substitute models, generated by our dynamic substitute structure learning strategy, are different according to various targets. Considering that the MNIST dataset is relatively simple compared to CIFAR-10, the number of skipped blocks is more and keeping blocks is fewer for attacking the MNIST model. Meanwhile, the AlexNet expresses relatively simple representations, thus, the corresponding optimal substitute models keep fewer blocks to learn knowledge from AlexNet.

**Social impact and limitations.** The task of adversarial attack is predominantly utilized as a tool for the verification and validation the robustness of the state-of-the-art deep models. Our DST method shall not be utilized to attack existing recognition systems. On the other hand, our DST still relies on white-box attack methods, we will focus on attacking directly and study the defense algorithm in future.

## 5. Conclusion

To tackle the data-free black-box attack task, we propose a novel dynamic substitute training attack method (DST). DST generate the optimal substitute model structure according to the different targets by the proposed dynamic substitute structure learning strategy, and encourage the substitute model to learn implicit information from the target one via graph-based structure information learning constrain. The experiments show that DST can achieve the best attack performance compared with existing methods.

## 6. Acknowledgement

This work was supported in part by NSFC Project (62176061), Shanghai Municipal Science and Technology Major Project (2018SHZDZX01), and Shanghai Research and Innovation Functional Program (17DZ2260900).



## References

- [1] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint*, 2017. 8
- [2] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, 2017. 1, 2, 4
- [3] 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 *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 15–26, 2017. 1, 2
- [4] Weiyu Cui, Xiaorui Li, Jiawei Huang, Wenyi Wang, Shuai Wang, and Jianwen Chen. Substitute model generation for black-box adversarial attack based on knowledge distillation. In *2020 IEEE International Conference on Image Processing (ICIP)*, pages 648–652. IEEE, 2020. 1, 2
- [5] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *CVPR*, 2018. 1, 2
- [6] Yinpeng Dong, Hang Su, Baoyuan Wu, Zhifeng Li, Wei Liu, Tong Zhang, and Jun Zhu. Efficient decision-based black-box adversarial attacks on face recognition. In *CVPR*, 2019. 1
- [7] Xianfeng Gao, Yu-an Tan, Hongwei Jiang, Quanxin Zhang, and Xiaohui Kuang. Boosting targeted black-box attacks via ensemble substitute training and linear augmentation. *Applied Sciences*, 9(11):2286, 2019. 1, 2
- [8] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint*, 2014. 1, 2, 4
- [9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 1, 2, 4
- [10] Zhaoxin Huan, Yulong Wang, Xiaolu Zhang, Lin Shang, Chilin Fu, and Jun Zhou. Data-free adversarial perturbations for practical black-box attack. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 127–138. Springer, 2020. 2
- [11] Qian Huang, Isay Katsman, Horace He, Zeqi Gu, Serge Belongie, and Ser-Nam Lim. Enhancing adversarial example transferability with an intermediate level attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4733–4742, 2019. 2
- [12] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. *arXiv preprint*, 2018. 1
- [13] Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. *arXiv preprint*, 2018. 1
- [14] Sanjay Kariyappa, Atul Prakash, and Moinuddin K Qureshi. Maze: Data-free model stealing attack using zeroth-order gradient estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13814–13823, 2021. 2, 3, 4, 5, 6, 8
- [15] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. *psu.edu*, 2009. 4
- [16] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 1, 2, 4
- [17] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *arXiv preprint arXiv:1607.02533*, 2016. 1, 2, 4
- [18] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 4
- [19] Maosen Li, Cheng Deng, Tengjiao Li, Junchi Yan, Xinbo Gao, and Heng Huang. Towards transferable targeted attack. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 641–649, 2020. 2
- [20] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008. 8
- [21] Aleksander Madry, Aleksandar Makelov, and Ludwig Schmidt. Towards deep learning models resistant to adversarial attacks. *arXiv preprint*, 2017. 1, 2, 4
- [22] Konda Reddy Mopuri, Aditya Ganeshan, and R Venkatesh Babu. Generalizable data-free objective for crafting universal adversarial perturbations. *IEEE transactions on pattern analysis and machine intelligence*, 41(10):2452–2465, 2018. 2, 4, 5, 6
- [23] Nina Narodytska and Shiva Prasad Kasiviswanathan. Simple black-box adversarial perturbations for deep networks. *arXiv preprint arXiv:1612.06299*, 2016. 8
- [24] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z. Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, 2016. 1, 2
- [25] Xuelin Qian, Huazhu Fu, Weiya Shi, Tao Chen, Yanwei Fu, Fei Shan, and Xiangyang Xue. M<sup>3</sup> lung-sys: A deep learning system for multi-class lung pneumonia screening from ct imaging. *IEEE journal of biomedical and health informatics*, 24(12):3539–3550, 2020. 2
- [26] Xuelin Qian, Yanwei Fu, Tao Xiang, Yu-Gang Jiang, and Xiangyang Xue. Leader-based multi-scale attention deep architecture for person re-identification. *IEEE transactions on pattern analysis and machine intelligence*, 42(2):371–385, 2019. 2
- [27] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 4
- [28] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 4

- [29] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. [1](#), [2](#), [4](#)
- [30] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint*, 2013. [1](#), [2](#)
- [31] Wenxuan Wang, Yanwei Fu, Xuelin Qian, Yu-Gang Jiang, Qi Tian, and Xiangyang Xue. Fm2u-net: Face morphological multi-branch network for makeup-invariant face verification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5730–5740, 2020. [2](#)
- [32] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4761–4770, 2021. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [8](#)
- [33] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. [4](#)
- [34] Jiancheng Yang, Yangzhou Jiang, Xiaoyang Huang, Bingbing Ni, and Chenglong Zhao. Learning black-box attackers with transferable priors and query feedback. *Advances in Neural Information Processing Systems*, 33, 2020. [2](#)
- [35] Bangjie Yin, Wenxuan Wang, Taiping Yao, Junfeng Guo, Zelun Kong, Shouhong Ding, Jilin Li, and Cong Liu. Advmakeup: A new imperceptible and transferable attack on face recognition. *International Joint Conference on Artificial Intelligence (IJCAI)*, 2021. [1](#)
- [36] Chaoning Zhang, Philipp Benz, Adil Karjauv, and In So Kweon. Data-free universal adversarial perturbation and black-box attack. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7868–7877, 2021. [2](#), [4](#), [5](#), [6](#)
- [37] Xiangyu Zhang, Xinyu Zhou, Mengxiao Lin, and Jian Sun. Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6848–6856, 2018. [4](#)
- [38] Mingyi Zhou, Jing Wu, Yipeng Liu, Shuaicheng Liu, and Ce Zhu. Dast: Data-free substitute training for adversarial attacks. In *CVPR*, 2020. [1](#), [2](#), [3](#), [4](#), [5](#), [6](#), [8](#)