DaST: Data-free Substitute Training for Adversarial Attacks

Mingyi Zhou[1,2,*], Jing Wu[1,2,*], Yipeng Liu[1,†], Shuaicheng Liu[1,2], Ce Zhu[1]
University of Electronic Science and Technology of China[1]
Megvii Technology[2]
{zhoumingyi,wujing}@std.uestc.edu.cn,{yipengliu,liushuaicheng,eczhu}@uestc.edu.cn

Abstract

Machine learning models are vulnerable to adversarial examples. For the black-box setting, current substitute attacks need pre-trained models to generate adversarial examples. However, pre-trained models are hard to obtain in real-world tasks. In this paper, we propose a data-free substitute training method (DaST) to obtain substitute models for adversarial black-box attacks without the requirement of any real data. To achieve this, DaST utilizes specially designed generative adversarial networks (GANs) to train the substitute models. In particular, we design a multi-branch architecture and label-control loss for the generative model to deal with the uneven distribution of synthetic samples. The substitute model is then trained by the synthetic samples generated by the generative model, which are labeled by the attacked model subsequently. The experiments demonstrate the substitute models produced by DaST can achieve competitive performance compared with the baseline models which are trained by the same train set with attacked models. Additionally, to evaluate the practicability of the proposed method on the real-world task, we attack an online machine learning model on the Microsoft Azure platform. The remote model misclassifies 98.35% of the adversarial examples crafted by our method. To the best of our knowledge, we are the first to train a substitute model for adversarial attacks without any real data. Our codes are publicly available [1].

1. Introduction

Deep neural networks have been shown vulnerable to examples with imperceptible perturbations [38]. This causes researchers a high interest in studying attacks and defenses for assessing and improving the robustness of networks. Adversarial attack methods can be categorized into two main attacks, white-box attacks that have full access to the attacked model and black-box attacks that have partial information of models.

Black-box attacks are more practical in real world systems compared with white-box attacks. Among these attacks, score-based attacks [8, 19, 20, 16] and decision-based attacks [3, 9, 7] directly attack the attacked model using class probabilities or hard labels returned by the attacked model. These attack methods do not need a pre-trained substitute model, however, as a cost, they need numerous queries for attacked models to generate each attack.

Instead, gradient-based attack methods [14, 22, 35, 30] need knowledge of the architecture and weights of the attacked models. Goodfellow *et al.* [14] showed that adversarial examples have the property of transferability which means that adversarial examples generated for one model through white-box attack methods can also attack other models. Therefore, to carry out attack methods in the black-box setting, they use a substitute model to find the adversarial examples and then attack the machine learning model based on the transferability of these adversarial examples. Compared with current score-based and decision-based attacks, the substitute attacks do not need queries for generating adversarial examples. However, they need a pre-trained model to generate adversarial attacks. Papernot *et al.* [34] developed a method that uses a number of images to imitate the outputs of attacked models to obtain substitute networks. Prediction APIs were also developed to steal the machine models [39]. Orekondy *et al.* [32] proposed a ”knockoff” to steal the function of machine learning models. These methods do not need a pre-trained model but require many real data labeled by the attacked model to train a substitute model. However, the real images are hard to get in some real-world tasks. Therefore, it is important to develop a data-free substitute attack, such that the risks faced by current machine learning models can be assessed more comprehensively.

In this study, we propose a data-free substitute training (DaST) method to train a substitute model for adversarial attacks. We utilize generative adversarial networks (GANs) to create synthetic samples to train the substitute model. The

[*] Equal contribution
[†] Corresponding author
[1] https://github.com/zhoumingyi/DaST

substitute model uses these samples to train, where the labels of the samples are produced by the attacked model. For the performance, the synthetic samples should be equally distributed in the input space. The label of samples should span all categories. However, conventional GANs without real data may generate samples that have extremely uneven distribution and only contain few categories, which means the substitute model cannot learn the classification characteristics of the attacked model comprehensively.

To address this problem, we design a multi-branch architecture and a label-control loss for the generative model to deal with the uneven distribution of synthetic samples. The generative model can produce synthetic samples with random labels given by the attacked model. As such, the substitute model can learn the classification characteristics of the attacked model in this adversarial training and produce adversarial examples which have strong transferability for the attacked model. The main contributions of this study are summarized as follows:

- We are the first to train a substitute model for adversarial attacks without any real data. Attackers can use this method to train a substitute model for adversarial attacks without collecting any real data.
- We evaluate the effectiveness of DaST on both local deep learning models and the online machine learning system, which reveals a fact that the current machine learning model has significant risks to be attacked.
- we evaluate the performance of our method in two attack scenarios, including a probability-only scenario that attacker can access the output probability of the attacked model, and a label-only scenario that attacker only accesses the output label of the attacked model. Our method generates adversarial examples efficiently on both scenarios.

In addition, we use different model architectures for the substitute model to test the influence on attack success rate caused by the model capacity.

The rest of our paper is organized as follows: in section 2, we introduce the related works. The proposed method is described in section 3. We evaluate the performance of DaST in section 4.

2. Related Works

Adversarial Scenes Adversarial attacks are carried out in the white-box setting or the black-box setting. In the white-box setting, the attacker has access to the structure and weights of the attacked models. On the contrary, in the black-box setting, the attacker only has the substitute model (gradient-based attacks) or access the outputs returned by the attacked models (query-based attacks). Black-box attack methods are more practical on real tasks.

Adversarial Attacks Gradient-based attacks such as FGSM [14] and BIM [22] have full access to the models, so they usually use a pre-trained substitute model to generate adversarial examples, and then attack the attacked model using the transferability of adversarial examples. FGSM aims to find adversarial examples by directly increasing the loss of the model, BIM is an iterative version of FGSM. Likewise, DeepFool [30] finds adversarial examples that are likely to cross the decision boundary. To find perturbations with minimal ℓ_p norm, Nicholas Carlini and David Wagner [6] introduced a method to craft these perturbations through simultaneously minimizing the perturbations. Similar to this method, Rony *et al*. [36] also constrain the ℓ_2 norm of the perturbations, they decoupled the value and direction of the perturbation. In the black-box setting, these attacks rely on the transferability of adversarial examples. However, Liu *et al*. [25] showed that these examples nearly have no transferability on attacked attacks. Instead, Cheng *et al*. [8] proposed a score-based attack method zeroth order based attack (ZOO) using gradient estimation, and Ilyas *et al*. [20] improve the way of the gradient estimation. Instead of gradient estimation, Guo *et al*. [16] introduced a simple black-box attack (SimBA) which decides the direction of the perturbations based on the changes of output probability. Brendel *et al*. [3] first proposed a decision-based attack. Based on this method, Cheng *et al*. [9] and Cheng *et al*. [7] improved the query efficiency, which is an important metric for black-box attacks.

Adversarial Defenses Several defense methods for increasing the robustness of models have been proposed. Adversarial training [38, 27, 23, 40] modifies the training schemes of the models, they directly train with the adversarial examples. Another method aims to modify the adversarial examples themselves such as random transformation [22, 28, 41]. Buckman *et al*. [4] proposed a nonlinear transformation based on one-hot encoding to inputs of models. Gradient masking methods [40, 10] destroy the gradient information so that they fail the optimization-based attacks. However, these defense methods based on gradient masking have been showed unreliable [1], and models with defenses above are still unsafe against some attacks [5, 17]. Besides, detecting adversarial examples raises the interest of researchers. Some of them detect the examples of whether they are adversarial or clean by an auxiliary network [13, 15, 29], while some find out adversarial examples through their statistical properties [2, 18, 12, 26, 33].

3. Method

In this section, we describe the attack scenario in this study, then introduce the substitute attack and propose a data-free method to train the substitute model.

3.1. Attack Scenario

Label-only scenario Suppose the attacked machine learning model is employed online and attackers can freely probe the output labels of the attacked model. The attackers are hard to obtain any data which is in the input space of the attacked model. We name the proposed DaST on the label-only scenario as DaST-L.

Probability-only scenario The other settings of this scenario are the same as the label-only scenario, but attackers can access the output probability of the attacked model. We name the proposed DaST on the probability-only scenario as DaST-P.

3.2. Adversarial Attack

In this subsection, we introduce the definition of adversarial substitute attacks.

$\mathbf{X}$ denotes samples from the input space of the attacked model T. $\bar{y}$ and y' refers to the real labels and the target labels of the samples $\mathbf{X}$, respectively. $T(y|\mathbf{X}, \theta)$ is the attacked model parameterized by θ. For non-targeted attacks, the objective of the adversarial attack can be formulated as:

$$\min_{\epsilon} \|\epsilon\| \ \text{ subject to } \ \operatorname*{argmax}_{y_i} T(y_i|\overline{\mathbf{X}}{=}\mathbf{X}{+}\epsilon, \theta) \neq \bar{y} \\ \text{and } \ \|\epsilon\| \leq r. \tag{1}$$

For targeted attacks, the objective is:

$$\min_{\epsilon} \|\epsilon\| \ \text{ subject to } \ \operatorname*{argmax}_{y_i} T(y_i|\overline{\mathbf{X}}{=}\mathbf{X}{+}\epsilon, \theta) = y' \\ \text{and } \ \|\epsilon\| \leq r, \tag{2}$$

where the ϵ and r are perturbation of the sample and upper bound of the perturbation, respectively. For attacking the machine learning system which is hard to detect, r is set to a small value in attack methods. $\overline{\mathbf{X}} = \mathbf{X} + \epsilon$ are the adversarial examples which can lead the attacked model T to output a wrong label (non-targeted setting) and a specific wrong label (targeted setting).

For white-box attacks, they can fully access the gradient information of T, then use it to generate adversarial examples to attack the T. For black-box substitute attacks, they train a model $\widehat{T}$ to substitute the attacked model to generate adversarial examples and then transfer the examples to attack the T. The attack success rate of these black-box attacks heavily relies on the transferability of the adversarial examples. Therefore, the key point of developing an efficient substitute attack is to train a substitute model having properties that are as similar as possible to the attacked model. Current attack methods utilize the same training set of the attacked model or collect a lot of other images labeled by the attacked model to train the substitute model. In the next two subsections, we will introduce a method that can train a substitute model without any image. The whole process is shown in Figure 1.

3.3. Adversarial Generator-Classifier Training

In this subsection, we introduce the basic adversarial training method and discuss its limitation.

For training the substitute model without any image, we use a generative model G to produce training data for the substitute model D. The generator randomly samples the noise vector $\mathbf{z}$ from the input space and produces the data $\widehat{\mathbf{X}} = G(\mathbf{z})$. Then, the generated data is used to probe the output $T(\widehat{\mathbf{X}})$ of the attacked model T. The substitute model is trained by the image-output pair $(\widehat{\mathbf{X}}, T(\widehat{\mathbf{X}}))$. As shown in Figure 1, the objective of G is to create new samples to explore the difference between T and D, and the role of D is to imitate the output of T. It is a special two-player game, the attacked model involved in this game is a referee. To simplify the expression but without loss of generality, we utilize the binary classification as a case to analyze (the output probability can be considered as one scalar in binary classification, so does the output label). The value function of the game is presented as:

$$\max_{G} \min_{D} \ \mathcal{V}_{G,D} = d(T(\widehat{\mathbf{X}}), D(\widehat{\mathbf{X}})) \tag{3}$$

where $d(T(\widehat{\mathbf{X}}), D(\widehat{\mathbf{X}}))$ is a metric to measure the output distance between T and D. For label-only attack scenario, this measurement can be formulated as:

$$d(T, D) = \text{CE}(D(\widehat{\mathbf{X}}), T(\widehat{\mathbf{X}})), \tag{4}$$

where $D(\widehat{\mathbf{X}})$ and $T(\widehat{\mathbf{X}})$ in this scenario denote the output labels of the substitute model and those of the attacked model, respectively. $\text{CE}(D(\widehat{\mathbf{X}}), T(\widehat{\mathbf{X}}))$ denotes the cross entropy loss, and the output labels of T are utilized as the label of this loss. The function of cross entropy loss is to constrain the difference between the T and D. For probability-only attack scenario, this measurement is formulated as:

$$d(T, D) = \|D(\widehat{\mathbf{X}}), T(\widehat{\mathbf{X}})\|_F, \tag{5}$$

where $D(\widehat{\mathbf{X}})$ and $T(\widehat{\mathbf{X}})$ in this scenario denote the output probabilities of the substitute model and those of the attacked model, respectively.

Hence the substitute model D replicates the information of attacked model T by this adversarial training. In the training, the loss function of D is set to $\mathcal{L}_D = \mathcal{V}_{G,D}$. In order to maintain the stability of training, the loss function of G is designed as $\mathcal{L}_G = e^{-d(T,D)}$. Therefore, the global optimal substitute network D is obtained if and only if $\forall\ \widehat{\mathbf{X}},\ T(\widehat{\mathbf{X}}) = D(\widehat{\mathbf{X}})$. At this point, $\mathcal{L}_D = 0$ and $\mathcal{L}_G = e^0 = 1$.

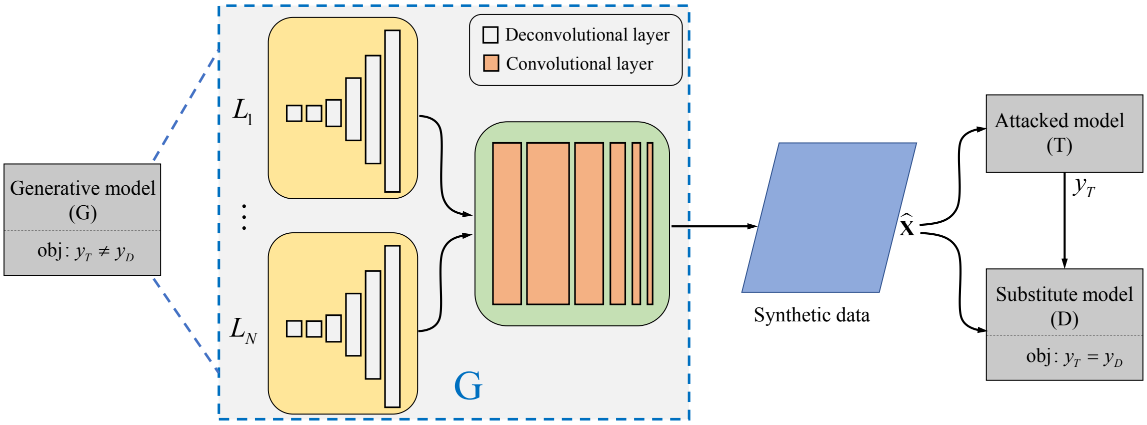

Figure 1. The proposed adversarial data-free imitation. The architecture of G is shown in the blue dotted block. N denotes the number of categories. In the training stage, the objective of G is to generate samples $\widehat{\mathbf{X}} = G(\mathbf{X})$ and let $y_D(\widehat{\mathbf{X}}) \neq y_T(\widehat{\mathbf{X}})$. The objective of D is to guarantee $y_D(\widehat{\mathbf{X}}) = y_T(\widehat{\mathbf{X}})$. In the testing stage, the substitute model D is utilized to generate adversarial examples to attack T.

We suppose that $\forall\ \widehat{\mathbf{X}} = G(\mathbf{z}), \widehat{\mathbf{X}} \in \mathbb{R}$, $\mathbb{R}$ is the input space of T. If the D can achieve $D(\widehat{\mathbf{X}}) = T(\widehat{\mathbf{X}})$, the adversarial attacks carried out by our substitute model will have the same success rate as the white-box attack without the gradient information of T. Therefore, for a well-trained substitute network, adversarial examples generated by D have strong transferability for T.

However, it is impossible to guarantee that $D(\widehat{\mathbf{X}}) = T(\widehat{\mathbf{X}})$ in a limited time. If we do not constrain the output of G, the synthetic training data for T is likely only distributed in a small range of $\mathbb{R}$, thus this training cannot work. For addressing this problem, we design a label-controllable architecture for G, which can control the distribution of synthetic data and speed up the convergence of training.

3.4. Label-controllable Data Generation

In this subsection, we introduce the label-controllable architecture for the generative model G.

To obtain equally distributed synthetic data to train the substitute model D, we consider developing a method that can control the distribution of $\widehat{\mathbf{X}}$. For training a replication of T, the synthetic data is used to probe the information of the attacked model. The label of samples, which is produced by the attacked model, should span all categories. Therefore, as shown in the blue dotted box of Figure 1, we design a generative network which contains N upsampling deconvolutional components, N is the number of categories. All upsampling components share a post-processing convolutional network. The model G randomly samples the noise vector $\mathbf{z}$ from the input space and variable label value n. The $\mathbf{z}$ is then entered into the n-th upsampling deconvolutional network and the shared convolutional network to produce the data $\widehat{\mathbf{X}} = G(\mathbf{z}, n)$. The additional label-control loss for generative model G is formulated as:

$$\mathcal{L}_C = \text{CE}(T(G(\mathbf{z}, n)), n). \tag{6}$$

The above method generates data with random labels, which are produced by T. However, the backpropagation of this label-control loss needs the gradient information of the attacked model T, it violates the rules of black-box attacks. We need to train a label-controllable generative model without the gradient information of T. For the imitation process, it can be approximated as the following objective function:

$$\min_{D} \quad d(T(\widehat{\mathbf{X}}), D(\widehat{\mathbf{X}})). \tag{7}$$

In the training progresses, the outputs of D will gradually approach the outputs of T under the same inputs. Therefore, we use D to replace the T in Eq. (6), which is formulated as:

$$\mathcal{L}_C = \text{CE}(D(G(\mathbf{z}, n)), n). \tag{8}$$

The training of substitute D can avoid accessing the information of T. Then we update the loss of G as:

$$\mathcal{L}_G = e^{-d(T,D)} + \alpha \mathcal{L}_C, \tag{9}$$

where α controls the weight of label-control loss (we set it to 0.2 in our experiments).

In the training stage, as the imitation ability of D increases, the diversity of synthetic samples which is labeled by the T will enhance. Therefore, the D can learn the information of the attacked model T, which can improve the transferability of adversarial examples generated by D. We

Algorithm 1 Mini-batch stochastic gradient descent training of the proposed method DaST.

```
# $acc$ denotes the accuracy of D. $att$ denotes the attack
success rate for the attacks generated by D.
While iteration $< \delta$ or $acc, att$ do not increace
    Generate $m$ examples$\{\widehat{\mathbf{X}}^{(1)}, \ldots, \widehat{\mathbf{X}}^{(m)}\}$by $G$.
    Update the substitute model :
        $\mathcal{L}_D = d(T(\widehat{\mathbf{X}}), D(\widehat{\mathbf{X}}))$.
    Update the generative model :
        $\mathcal{L}_G = e^{-d(T,D)} + \alpha\mathcal{L}_C$.
end for
```

name this method as data-free substitute training (DaST), which is shown in Algorithm 1.

Like the current substitute attack methods, the substitute model trained by our method is utilized to generate adversarial examples to attack T.

4. Experiments

4.1. Experiment Setting

In this subsection, we introduce our experiment settings, including the datasets, model architectures, attack methods, and evaluation criteria.

Dataset: we evaluate our proposed method on MNIST [24] and CIFAR-10 [21]. The test sets of these two datasets have 10k images, respectively.

Scenario: we evaluate our method in both label-only attack and probability-only scenario. The DaST-L and DaST-P denote the DaST in the label-only scenario and DaST in the probability-only scenario, respectively. Attackers in the scenarios of this study can freely access the output of the attacked model. Therefore, we obtain the substitute model trained by DaST when the algorithm convergence.

Model architecture and attack method: the substitute network has no prior knowledge of the attacked model, which means it does not load any pre-trained model in experiments. For the experiments on MNIST, we design 3 different network architectures including a small network (3 convolutional layers), a medium network (4 convolutional layers) and a large network (5 convolutional layers) for evaluating the performance of our DaST with models having different capacity. We utilize the pre-trained medium network and VGG-16 [37] as the attacked model on MNIST and CIFAR-10, respectively. In addition, we use different architectures for the substitute model and attacked model to evaluate the impact of model structure on our method in CIFAR-10 experiments. In order to compare the substitute model produced by DaST with the pre-trained models, we utilize 4 attack methods to generate adversarial examples, which include FGSM [14], BIM [22], projected gradient descent (PGD) [27], C&W [6]. For testing, we use AdverTorch library [11] to generate adversarial examples. For evaluating performances of the proposed method in real-world tasks, we apply our attack to the online MNIST model of Microsoft Azure. The training tricks and machine learning methods utilized by this online model cannot be accessed.

Table 1. Performance of the proposed DaST on MNIST. "Pre-trained", "DaST-L" and "DaST-P": the attack success rate (%) of adversarial examples generated by the pre-trained large network and DaST-L and DaST-P, respectively. ( ) denotes the average L_F perturbation distance per image.

Attack	**Non-targeted**		
	Pre-trained	DaST-P	DaST-L
FGSM [14]	59.72 (5.40)	**69.76** (5.41)	35.74 (5.40)
BIM [22]	85.70 (4.80)	**96.36** (4.81)	64.61 (4.82)
PGD [27]	37.93 (3.98)	**53.99** (3.99)	23.22 (3.98)
C&W [6]	23.34 (2.91)	**27.35** (2.74)	18.16 (2.75)
Attack	**Targeted**		
	Pre-trained	DaST-P	DaST-L
FGSM [14]	12.10 (5.46)	**20.45** (4.49)	13.10 (5.46)
BIM [22]	37.83 (4.90)	**57.22** (4.87)	29.18 (4.87)
PGD [27]	28.95 (4.60)	**47.57** (4.63)	19.25 (4.63)
C&W [6]	10.32 (2.57)	**23.80** (2.99)	12.31 (2.98)

Evaluation criteria: for evaluating the performance of our DaST, we set the attack success rates of adversarial examples generated by other pre-trained networks as the baseline. The goals of non-targeted attacks and targeted attacks are to lead the attacked model to output wrong labels and specific wrong labels, respectively. In the non-targeted attack scenario, we only generate adversarial examples on the images classified correctly by the attacked model. In targeted attacks, we only generate adversarial examples on the images which are not classified to the specific wrong labels. The success rates of adversarial attack are calculated by n/m, where n and m are the number of adversarial examples which can fool the attacked model and the total number of adversarial examples, respectively.

4.2. Experiments on MNIST

In this subsection, we employ the proposed DaST to train a substitute model for adversarial attacks on the MNIST dataset and evaluate the performance in terms of attack success rate in label-only and probability-only scenarios.

First, we conduct experiments to evaluate the performance in probability-only and label-only attack scenarios. We use the medium network as the attacked model on MNIST and the large network as the substitute model of DaST. We train a pre-trained large network on the same train set of the attacked model. We utilize the attack success

Table 2. Performances of the proposed DaST with three different substitute architectures on MNIST. "Small", "Medium" "Large": the attack success rates (%) of adversarial examples generated by DaST with small, medium and large substitute networks, respectively. ( ) denotes the average L_F perturbation distance per image.

Attack	Non-targeted		
	Small	Medium	Large
FGSM [14]	62.61 (4.38)	56.21 (4.45)	**69.76** (5.41)
BIM [22]	94.86 (4.85)	92.47 (4.84)	**96.36** (4.81)
PGD [27]	45.31 (3.99)	43.62 (3.99)	**53.99** (3.99)
C&W [6]	**30.61** (2.89)	24.34 (2.75)	23.80 (2.99)
Attack	Targeted		
	Small	Medium	Large
FGSM [14]	19.92 (4.43))	20.45 (4.49)	**23.93** (5.45)
BIM [22]	56.73 (4.89)	53.50 (4.84)	**57.22** (4.87)
PGD [27]	39.42 (4.64)	40.76 (4.60)	**47.57** (4.63)
C&W [6]	**24.86** (3.09)	16.25 (3.13)	23.80 (2.99)

Table 3. Performance of the proposed DaST on CIFAR-10. "Pre-trained", "DaST-P" "DaST-L": the attack success rates (%) of adversarial examples generated by the pre-trained large network, DaST-P and DaST-L, respectively. ( ) denotes the average L_F perturbation distance per image.

Attack	Non-targeted		
	Pre-trained	DaST-P	DaST-L
FGSM [14]	39.10 (1.54)	**39.63** (1.54)	22.65 (1.54)
BIM [22]	59.18 (1.01)	**59.71** (1.18)	28.42 (1.19)
PGD [27]	**35.40** (1.02)	29.10 (1.10)	17.80 (1.10)
C&W [6]	9.76 (0.77)	**13.52** (0.74)	10.34 (0.74)
Attack	Targeted		
	Pre-trained	DaST-P	DaST-L
FGSM [14]	**9.62** (1.54)	6.69 (1.54)	7.32 (1.54)
BIM [22]	17.43 (1.00)	**20.22** (1.18)	15.26 (1.16)
PGD [27]	10.46 (1.05)	**14.09** (1.12)	8.32 (1.10)
C&W [6]	23.15 (2.05)	**26.53** (1.98)	19.78 (2.04)

rate of adversarial examples generated by the pre-trained model as the baseline. The performances of our DaST are shown in Table 1. The substitute model trained by DaST-P and DaST-L achieve 97.82% and 83.95% of accuracy on the test set, respectively. The attack success rates of the substitute model produced by our DaST are higher than those of the pre-trained model on non-targeted (10.04%, 10.66%, 16.06%, and 4.01% higher on FGSM, BIM, PGD, and C&W, respectively) and targeted attacks (11.83%, 19.39, 18.62, 13.48% higher on FGSM, BIM, PGD, and C&W, respectively). It shows that the substitute model generated by DaST-P outperform the models trained by the same train set (60000 images) with the attacked model. Even he substitute models trained by DaST-L perform better than baseline models on FGSM and C&W attacks (targeted).

Then we evaluate the performances of our DaST with different substitute architectures in the probability-only scenario. We also use the medium network as the attacked model on MNIST and apply our DaST using three different substitute architectures, which include the large, medium and small networks. The attack success rates of these three substitute architectures are shown in Table 2. The large substitute model achieves the best results on FGSM, BIM, PGD attacks compared with other models. The small substitute model obtains the best results on C&W attacks compared with other models. It shows that both architectures for the substitute model obtain good results on adversarial attacks. In general, the substitute models with more complex structure can obtain better performance for adversarial attacks.

4.3. Experiments on CIFAR-10

In this subsection, we employ the proposed DaST to train a substitute model for adversarial attacks on the CIFAR-10 dataset, and evaluate the performance in terms of attack success rate in label-only and probability-only scenarios.

We conduct experiments to evaluate the performance in probability-only and label-only attack scenarios and use the VGG-16 network as the attacked model. We train a pre-trained ResNet-50 network on the same train set of the attacked model. The performances of our DaST are shown in Table 3. The substitute model trained by DaST-P and DaST-L achieve 25.15% and 20.35% of accuracy on the test set, respectively. Our DaST also achieves competitive performance with the pre-trained model. In most cases of the probability-only scenario (FGSM, BIM, C&W for non-targeted attack, BIM, PGD, C&W for targeted attacks), the substitute models generated by DaST-P outperform baseline models. The substitute models trained by DaST-L perform better than baseline models on C&W attacks (non-targeted).

We also evaluate the performances of our DaST with different substitute architectures in the probability-only scenario. The VGG-16 network is used as the attacked model. We apply our DaST using 3 different substitute architectures, which include the VGG-13, ResNet-18, and ResNet-50. The attack success rates of these three substitute architectures are shown in Table 4. It demonstrates that both architectures for the substitute model obtain good results on adversarial attacks. In most cases (BIM, PGD, C&W for non-targeted attack, FGSM, BIM, PGD, C&W for targeted attacks), the VGG-13 outperforms other models in terms of the adversarial attack. The ResNet-50 obtains the best results on FGSM attacks (targeted). Different from experiments on MNIST, the simple model achieves the best results on CIFAR-10. We visualize the adversarial examples generated by DaST-P and DaST-L in Figure 2 and 3, respectively. The attack perturbations for these two scenarios are small.

Table 4. Performances of the proposed DaST with three different substitute architectures on CIFAR-10. "VGG-13", "ResNet-18" "ResNet-50": the attack success rates (the high is better) of adversarial examples generated by DaST with VGG-13, ResNet-18 and ResNet-50 substitute models, respectively. The numbers in ( ) denote the average L_F perturbation distance per image.

Attack	**Non-targeted** (%)		
	VGG-13	ResNet-18	ResNet-50
FGSM [14]	6.87 (1.54)	17.97 (1.54)	**39.63** (1.54)
BIM [22]	**93.13** (1.18)	31.70 (1.54)	59.71 (1.18)
PGD [27]	**56.14** (1.08)	10.04 (1.11)	29.10 (1.10)
C&W [6]	**56.80** (1.64)	11.54 (1.64)	13.52 (0.74)
Attack	**Targeted** (%)		
	VGG-13	ResNet-18	ResNet-50
FGSM [14]	**18.27** (1.54)	2.07 (1.54)	6.69 (1.54)
BIM [22]	**62.23** (1.24)	8.00 (1.52)	20.22 (1.18)
PGD [27]	**41.48** (1.17)	3.72 (1.26)	14.09 (1.12)
C&W [6]	**33.65** (2.42)	7.31 (1.46)	26.53 (1.98)

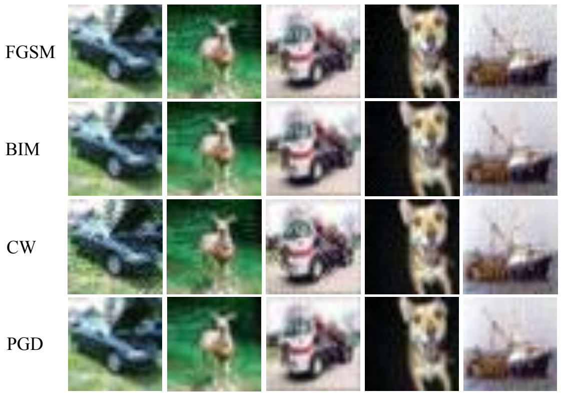

Figure 2. Visualization of the adversarial examples generated by DaST-L on CIFAR-10. We generate 5 samples for each attack.

Table 5. Performance of the proposed DaST for attacking Microsoft Azure example model. "Pre-trained", "DaST-P" "DaST-L": the attack success rate (the high is better) of adversarial examples generated by the pre-trained large network, DaST in probability-only scenario and DaST in label-only scenario, respectively. The numbers in ( ) denote the average L_F perturbation distance per image. Because it is hard to generate adversarial examples for all methods on C&W [6], we omit this attack method.

Attack	**Non-targeted** (%)		
	Pre-trained	DaST-P	DaST-L
FGSM [14]	77.96 (5.41)	96.83 (5.25)	**98.21** (5.36)
BIM [22]	66.25 (4.81)	96.42 (4.79)	**98.35** (4.72)
PGD [27]	59.23 (3.99)	90.63 (3.88)	**96.97** (3.96)
Attack	**Targeted** (%)		
	Pre-trained	DaST-P	DaST-L
FGSM [14]	13.52 (5.46)	32.00 (5.21)	**43.99** (5.37)
BIM [22]	19.31 (4.88)	50.21 (4.90)	**71.15** (4.56)
PGD [27]	19.31 (4.60)	45.66 (4.46)	**65.91** (4.32)

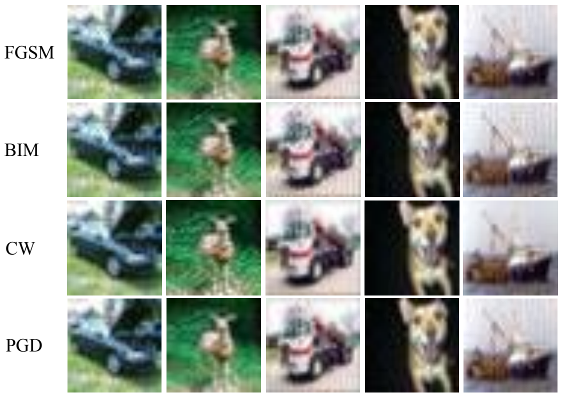

Figure 3. Visualization of the adversarial examples generated by DaST-P on CIFAR-10. We generate 5 samples for each attack.

4.4. Experiments on Microsoft Azure

In this subsection, we conduct experiments for attacking the online model on Microsoft Azure in two scenarios.

We use the example MNIST model of the machine learning tutorial on Azure as the attacked model and employ it as a web service. We do not know the machine learning method and architecture of this model. The only information we can obtain is the outputs of this model. We apply the probability-based DaST and label-based DaST attacks to this model to evaluate the performance of the proposed method in real-world applications. The substitute model in this experiment has 5 convolutional layers. The substitute model trained by DaST-P and DaST-L achieve 79.35% and 90.75% of accuracy on the MNIST test set, respectively. The performance on adversarial attacks of the proposed method is shown in Table 5.

The performance of DaST-L is better than its of DaST-P on this online model. Because the attacked Azure model is too simple, the accuracy on MNIST is only 91.93%. Figure 6 shows the training of DaST-P, which can access more information of attacked model than DaST-L, suffers over-fitting. DaST-L substitutes achieve a very high attack success rate on FGSM (98.21%), BIM (98.35%), PGD (96.97%) attacks. Moreover, our DaST method achieves a high attack success rate even on targeted attacks. Compared with the models trained by the MNIST train set, substitute models trained by DaST perform much better on label-only

Table 6. Comparison of DaST and other attacks. ”ASR”: attack success rate. ”Query”: the number of queries in the evaluation stage. ”Boundary”: Decision-Based Attacks [3]. ”GLS”: a score-based black-box attack based on greedy local search [31]. ”-” denotes our DaST does not need query in the evaluation stage. The DaST in this experiment generate attacks with BIM.

Attack	ASR	Distance	Query
DaST-P	96.83%	4.79	-
GLS [31]	40.51%	4.27	297.07
DaST-L	98.35%	4.72	-
Boundary [3]	100%	4.69	670.54

(20.25%, 32.10%, 37.74% higher on non-targeted FGSM, BIM, PGD attacks, respectively. 30.47%, 51.84%, 46.60% higher on targeted FGSM, BIM, PGD attacks, respectively) and probability-only scenarios. It presents that our approach is better at attacking actual online models, even the proposed method does not need any real data. Because DaST does not need any query in the evaluation stage but needs queries in the training stage, our DaST requires different information than score-based attacks and decision-based attacks (they need queries in the evaluation stage). We show the number of queries for score-based and decision-based attacks, which have similar perturbation distance with DaST in non-targeted attacks. The results are shown in Table 6. Our DaST is trained by 20,000,000 queries for the attacked model in the training stage. Compared with decision-based and score-based attacks, the input each time the DaST accesses the attacked model is different in the training stage (current query-based attacks need to use one original data to access the attacked model numerous times to generate each attack). So the queries of DaST are harder to be tracked than other attacks.

Visualization: we visualize the synthetic samples generated by the generative model in DaST on Azure experiments, which is shown in Figure 4. We also visualize the adversarial examples generated by DaST-P and DaST-L in Figure 5. The attack perturbations of DaST are small.

Training convergence: We show the curve of attack success rate of BIM attacks generated by DaST in the training stage of Azure experiments, which is shown in Figure 6. The attack success rates for DaST-L and DaST-P converge after 20,000,000 and 2,000,000 queries, respectively.

5. Conclusion

We have presented a data-free method DaST to train substitute models for adversarial attacks. DaST reduces the prerequisites of adversarial substitute attacks by utilizing GANs to generate synthetic samples. This is the first method that can train substitute models without the requirement of any real data. The experiments showed the effectiveness of our method. It presented that machine learning systems have significant risks, attackers can train substitute models even when the real input data is hard to collect.

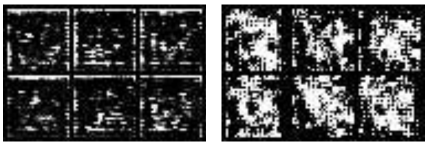

Figure 4. Visualization of the synthetic samples generated by the generator in the training of DaST. Left: samples generated by the DaST-L. Right: samples generated by the DaST-P.

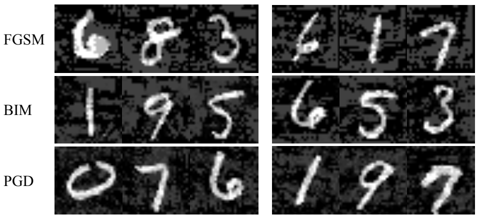

Figure 5. Visualization of the adversarial examples generated by DaST for attacking the Azure model. Left: examples generated by DaST-P. Right: examples generated by DaST-L.

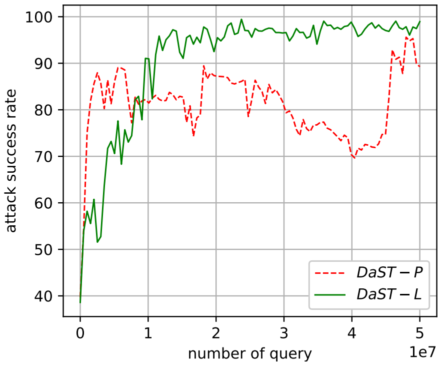

Figure 6. Attack success rate of BIM attacks generated by DaST in training stage of Azure experiments.

The proposed DaST cannot generate adversarial examples alone, it should be used with other gradient-based attack methods. In future work, we will design a new substitute training method, which can generate attacks directly. Furthermore, we will explore the defense for DaST.

6. Acknowledgment

This research is supported by National Natural Science Foundation of China (NSFC, No. 61602091, No. 61571102, No.61872067) and Sichuan Science and Technology Program (No. 2019YFH0008, No.2018JY0035, No.2019YFH0016).

References

[1] Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proceedings of the 35th International Conference on Machine Learning, ICML 2018*, July 2018. 2

[2] Arjun Nitin Bhagoji, Daniel Cullina, and Prateek Mittal. Dimensionality reduction as a defense against evasion attacks on machine learning classifiers. *arXiv preprint arXiv:1704.02654*, 2017. 2

[3] Wieland Brendel, Jonas Rauber, and Matthias Bethge. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. *arXiv preprint arXiv:1712.04248*, 2017. 1, 2, 8

[4] Jacob Buckman, Aurko Roy, Colin Raffel, and Ian Goodfellow. Thermometer encoding: One hot way to resist adversarial examples. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[5] Nicholas Carlini and David Wagner. Adversarial examples are not easily detected: Bypassing ten detection methods. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 3–14. ACM, 2017. 2

[6] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. IEEE, 2017. 2, 5, 6, 7

[7] Jianbo Chen, Michael I. Jordan, and Martin J. Wainwright. Hopskipjumpattack: A query-efficient decision-based attack, 2019. 1, 2

[8] 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. ACM, 2017. 1, 2

[9] Minhao Cheng, Thong Le, Pin-Yu Chen, Jinfeng Yi, Huan Zhang, and Cho-Jui Hsieh. Query-efficient hard-label black-box attack: An optimization-based approach. *arXiv preprint arXiv:1807.04457*, 2018. 1, 2

[10] Guneet S. Dhillon, Kamyar Azizzadenesheli, Jeremy D. Bernstein, Jean Kossaifi, Aran Khanna, Zachary C. Lipton, and Animashree Anandkumar. Stochastic activation pruning for robust adversarial defense. In *International Conference on Learning Representations*, 2018. 2

[11] Gavin Weiguang Ding, Luyu Wang, and Xiaomeng Jin. AdverTorch v0.1: An adversarial robustness toolbox based on pytorch. *arXiv preprint arXiv:1902.07623*, 2019. 5

[12] Reuben Feinman, Ryan R Curtin, Saurabh Shintre, and Andrew B Gardner. Detecting adversarial samples from artifacts. *arXiv preprint arXiv:1703.00410*, 2017. 2

[13] Zhitao Gong, Wenlu Wang, and Wei-Shinn Ku. Adversarial and clean data are not twins. *arXiv preprint arXiv:1704.04960*, 2017. 2

[14] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *International Conference on Learning Representations (ICLR)*, 2015. 1, 2, 5, 6, 7

[15] Kathrin Grosse, Praveen Manoharan, Nicolas Papernot, Michael Backes, and Patrick McDaniel. On the (statistical) detection of adversarial examples. *arXiv preprint arXiv:1702.06280*, 2017. 2

[16] Chuan Guo, Jacob Gardner, Yurong You, Andrew Gordon Wilson, and Kilian Weinberger. Simple black-box adversarial attacks. In *International Conference on Machine Learning*, pages 2484–2493, 2019. 1, 2

[17] Warren He, James Wei, Xinyun Chen, Nicholas Carlini, and Dawn Song. Adversarial example defense: Ensembles of weak defenses are not strong. In *11th Workshop on Offensive Technologies (WOOT 2017)*, 2017. 2

[18] Dan Hendrycks and Kevin Gimpel. Early methods for detecting adversarial images. *International Conference on Learning Representations (ICLR)*, 2017. 2

[19] Andrew Ilyas, Logan Engstrom, Anish Athalye, and Jessy Lin. Black-box adversarial attacks with limited queries and information. In *International Conference on Machine Learning*, pages 2142–2151, 2018. 1

[20] Andrew Ilyas, Logan Engstrom, and Aleksander Madry. Prior convictions: Black-box adversarial attacks with bandits and priors. *arXiv preprint arXiv:1807.07978*, 2018. 1, 2

[21] Alex Krizhevsky and Geoffrey Hinton. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009. 5

[22] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial examples in the physical world. *International Conference on Learning Representations (ICLR)*, 2017. 1, 2, 5, 6, 7

[23] Alexey Kurakin, Ian Goodfellow, and Samy Bengio. Adversarial machine learning at scale. *International Conference on Learning Representations (ICLR)*, 2017. 2

[24] Yann LeCun, Léon Bottou, Yoshua Bengio, Patrick Haffner, et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 5

[25] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *International Conference on Learning Representations (ICLR)*, 2017. 2

[26] Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Michael E. Houle, Dawn Song, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[27] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations(ICLR)*, 2018. 2, 5, 6, 7

[28] Dongyu Meng and Hao Chen. Magnet: a two-pronged defense against adversarial examples. In *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 135–147. ACM, 2017. 2

[29] Jan Hendrik Metzen, Tim Genewein, Volker Fischer, and Bastian Bischoff. On detecting adversarial perturbations. *International Conference on Learning Representations (ICLR)*, 2017. 2

[30] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016. 1, 2

[31] Nina Narodytska and Shiva Prasad Kasiviswanathan. Simple black-box adversarial perturbations for deep networks. *arXiv preprint arXiv:1612.06299*, 2016. 8

[32] Tribhuvanesh Orekondy, Bernt Schiele, and Mario Fritz. Knockoff nets: Stealing functionality of black-box models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 1

[33] Nicolas Papernot and Patrick McDaniel. Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning. *arXiv preprint arXiv:1803.04765*, 2018. 2

[34] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519. ACM, 2017. 1

[35] 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)*, pages 372–387. IEEE, 2016. 1

[36] Jérôme Rony, Luiz G Hafemann, Luis S Oliveira, Ismail Ben Ayed, Robert Sabourin, and Eric Granger. Decoupling direction and norm for efficient gradient-based l2 adversarial attacks and defenses. *arXiv preprint arXiv:1811.09600*, 2018. 2

[37] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations (ICLR)*, 2015. 5

[38] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *International Conference on Learning Representations (ICLR)*, 2014. 1, 2

[39] Florian Tramèr, Fan Zhang, Ari Juels, Michael K Reiter, and Thomas Ristenpart. Stealing machine learning models via prediction apis. In *25th {USENIX} Security Symposium ({USENIX} Security 16)*, pages 601–618, 2016. 1

[40] Florian Tramèr, Alexey Kurakin, Nicolas Papernot, Ian Goodfellow, Dan Boneh, and Patrick McDaniel. Ensemble adversarial training: Attacks and defenses. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[41] Cihang Xie, Jianyu Wang, Zhishuai Zhang, Zhou Ren, and Alan Yuille. Mitigating adversarial effects through randomization. In *International Conference on Learning Representations (ICLR)*, 2018. 2