Physical Black-box Adversarial Attacks through Transformations

Wenbo Jiang, *Student Member, IEEE,* Hongwei Li (Corresponding author), *Senior Member, IEEE,* Guowen Xu, *Member, IEEE,* Tianwei Zhang, *Member, IEEE,* and Rongxing Lu, *Fellow, IEEE*

Abstract—Deep learning has shown impressive performance in numerous applications. However, recent studies have found that deep learning models are vulnerable to adversarial attacks, where the attacker adds imperceptible perturbations into benign samples to induce misclassifications. Adversarial attacks in the digital domain focus on constructing imperceptible perturbations. However, they are always less effective in the physical world because the perturbations may be destroyed when captured by the camera. Most physical adversarial attacks require adding invisible adversarial features (e.g., a sticker or a laser) to the target object, which may be noticed by human eyes. In this work, we propose to employ image transformation to generate more natural adversarial samples in the physical world. Concretely, we propose two attack algorithms to satisfy different attack goals: *Efficient-AATR* employs a greedy strategy to generate adversarial samples with fewer queries; *Effective-AATR* employs an adaptive particle swarm optimization algorithm to search for the most effective adversarial samples within the given the number of queries. Extensive experiments demonstrate the superiority of our attacks compared with state-of-the-art adversarial attacks under mainstream defenses.

Index Terms—Physical adversarial attack, Black-box attack, Deep learning.

1 Introduction

Deep neural networks (DNN) have achieved very noticeable success in various domains and are being deployed in an increasing number of real-world applications, including but not limited to image recognition, speech recognition and autonomous vehicles. Nevertheless, recent studies have found that the well-trained models are susceptible to adversarial attacks, where the attacker adds almost imperceptible perturbations into a benign sample in order to make the model misclassify the sample with a high probability. Adversarial attacks pose serious security threats to deep learning models, especially those applied in security-critical scenarios [1–3].

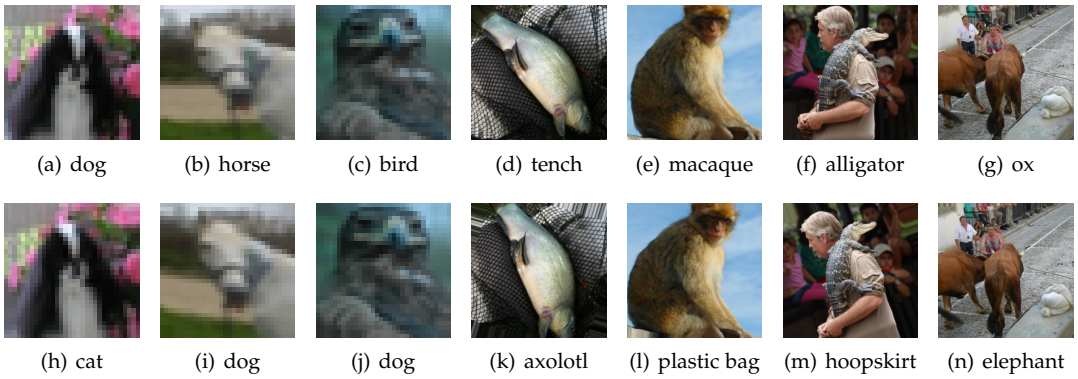
Most adversarial attacks focus on the digital domain [4–18], where adversarial examples are generated by changing image pixels and fed directly to DNN classifiers. However, in physical world scenarios, since the model only receives images from the camera (or other sensors), the perturbations need to be added to the target object physically (or change the target object physically) rather than changing image pixels. Due to the variations caused by the camera, achieving a perturbation-based adversarial attack in physical scenarios always requires much larger perturbations than that in digital scenarios, making the perturbations easily detectable [19]. Some efforts added stickers or patches [20–22] to the target object to generate physical adversarial samples and some works employed natural phenomenon (such as optical phenomenon [23–25] and shadows [26]) to construct physical adversarial samples. However, these attacks require adding adversarial features (e.g., a sticker or a laser) to the original sample, which may be noticed by human eyes.

In this work, we propose to employ more common and stealthy transformations, i.e., translation and rotation, to construct physical adversarial samples in the black-box setting. There is also a work that generates adversarial samples by transformations [27]. It proposed three attack methods: *First-Order Method* employs *PGD* to optimize adversarial transformed samples; *Grid Search* is an exhaustive search method that searches every possible parameter (rotation of angle and translations) in the parameter space until it finds the parameter that induces the model to misclassify the sample; *Worst-of-k* searches for the most effective adversarial sample in k random parameter combinations. However, these methods have limitations: *First-Order Method* employs *PGD* to optimize adversarial samples, which is inapplicable to more practical black-box scenarios. *Grid Search* and *Worst-of-k* are all simple exhaustive search and random search methods, which have shortcomings in effectiveness and efficiency (which will be demonstrated by our experiments in Section 5).

Specifically, we propose two optimization-based physical black-box adversarial attacks through translation and rotation, i.e., *Efficient-AATR* and *Effective-AATR*, which are more efficient than the exhaustive methods proposed in [27]. *Efficient-AATR* is aimed at inducing a misclassification with fewer queries. It employs a greedy strategy to generate adversarial samples and stops when the model misclassifies the sample; *Effective-AATR* is designed to achieve a higher misclassification probability within a given number of queries. It employs an APSO (Adaptive Particle Warm Optimization) algorithm to search for the optimal rotation

- *W. Jiang, H. Li are with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: wenbo_jiang@outlook.com, hongweili@uestc.edu.cn).*
- *G. Xu and T. Zhang are with School of Computer Science and Engineering, Nanyang Technological University, Singapore (e-mail: guowen.xu, tianwei.zhang@ntu.edu.sg).*
- *R. Lu is with the Faculty of Computer Science, University of New Brunswick, Fredericton, NB, Canada E3B 5A3 (e-mail: RLU1@unb.ca).*

This article has been accepted for publication in IEEE Transactions on Big Data. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2022.3227318



(a) dog (b) horse (c) bird (d) tench (e) macaque (f) alligator (g) ox

(h) cat (i) dog (j) dog (k) axolotl (l) plastic bag (m) hoopskirt (n) elephant

Fig. 1: Adversarial samples generated by our proposed attacks (from CIFAR-10 and ImageNet): the first column provides the benign samples and the column presents the adversarial samples generated by our proposed attacks. Their prediction results are all changed after our adversarial transformations.

angle and translation. Compared with traditional adversarial attacks based on additional adversarial perturbations, our attacks are easier to perform in the real world and more robust to defenses. Compared with other physical adversarial attacks, our attacks do not require adding adversarial features to the target object, making the adversarial sample more indistinguishable from the original sample. The generated adversarial samples of our attacks are illustrated in Fig. 1. In summary, the contributions of our work are as follows:

- We explore a way to generate more natural adversarial samples in the physical world, i.e., generating adversarial samples through translation and rotation.
- We propose two optimization-based attack algorithms to search for the optimal translation and rotation to construct adversarial samples.
- We conduct extensive evaluations to show our attacks are more effective and query-efficient than state-of-the-art adversarial attacks under mainstream defense mechanisms.

The remainder of this paper proceeds as follow. Section 2 overviews the preliminaries. Section 3 depicts our adversary model and Section 4 describes the details of our attack methodologies. Experimental evaluation will be carried out in Section 5. Finally, Section 6 concludes the paper.

2 Preliminaries

In this section, we first review previous work on adversarial attacks and defenses. Then, we introduce the technique of Particle Warm Optimization (PSO).

2.1 Adversarial Attacks

2.1.1 Digital Adversarial Attacks

Adversarial attacks in the digital world have been intensively investigated. Early studies of adversarial attacks concentrated on gradient-based attacks in white-box scenarios (such as *Fast Gradient Sign Method (FGSM)* [4] and *Project Gradient Descent (PGD)* [5]). Recent research efforts have concentrated on more realistic black-box adversarial scenarios, where the attacker has no knowledge of the target model. There are mainly two approaches to achieve black-box adversarial attacks: Transfer-based black-box attacks [6–11] reconstruct a substitute model which is similar to the target model and employ the transferability of adversarial samples to attack the target model; Query-based black-box attacks [12–18] optimize adversarial samples based on the corresponding output (label or probability) by querying the target model.

Nevertheless, the effectiveness of transfer-based black-box attacks depends largely on the transferability of the adversarial sample and cannot achieve a high attack success rate. The query-based black-box attacks often require a huge number of queries to achieve a high attack success rate. In addition, the small perturbation in the digital domain is always less effective in the physical world [19].

2.1.2 Physical Adversarial Attacks

Achieving an adversarial attack in the physical domain is more challenging and has received more attention recently. *Kurakin et al.* [19] proposed to generate physical world adversarial samples using iterative *FGSM*, which first generates perturbed digital adversarial samples and later prints digital adversarial samples as physical adversarial samples. However, it requires much larger perturbations than that in digital scenarios, making the perturbations detectable. Several works have proposed to add stickers or patches to images to generate physical adversarial samples [20–22]. For example, *Brown et al.* [20] proposed an adversarial patch method, which generated an adversarial patch and constructed physical adversarial samples by adding this patch to clean samples. Rather than adding visible adversarial patterns to the image, some recent works employed natural phenomenons (such as optical phenomena [23–25] and shadows [26]) to perform physical adversarial attacks. For example, *Sayles et al.* [23] crafted a maliciously modulated light signal and illuminated an image in such light signal. Then, this image will be misclassified by the deep learning model; *Duan et al.* [24] achieved an adversarial attack by producing an adversarial laser beam in front of the target object. *Gnanasambandam et al.* [25] utilized structured illumination to modify the appearance of the target objects and caused misclassification. *Zhong et al.* [26] used natural shadows to construct physical adversarial samples.

However, these attacks require adding adversarial features (e.g., a sticker or a laser) to the original sample, which are invisible to human eyes.

2.2 Defenses against Adversarial Attacks

2.2.1 Detection-based Defenses

This type of defenses aims on defending against adversarial attacks through detecting adversarial samples [28–32]. For example, *Ma et al.* [28] found that *Local Intrinsic Dimension (LID)* [33] of adversarial samples are obviously higher than that of clean samples. Thus, they proposed a detection method which identifies adversarial sample through *LID*. *Ma et al.* [29] proposed an another detection method, which detects adversarial samples through *Neural-network Invariant Checking (NIC)*.

2.2.2 Input Preprocessing

Input preprocessing transforms the input image before feeding it to the network in order to reduce the model sensitivity to adversarial perturbations [34–36]. For instance, *Tian et al.* [37] explored image transformation (such as rotation and shifting) to detect adversarial attacks. *Aydemir et al.* [35] and *Dziugaite et al.* [36] focused on employing image compression to decrease the effectiveness of adversarial attacks.

2.2.3 Model Robustness Enhancement

This type of methods modifies the model to improve robustness against adversarial samples. The most representative technique is adversarial training [38–42], which enhances the robustness by training the model with some adversarial samples (with correct labels). For example, *Jin et al.* [39] enhanced the adversarial training through second-order statistics optimization with respect to the weights.

In addition to adversarial training, several approaches alter the model architecture to enhance robustness [43–45]. For instance, neural architecture search (NAS) [46] is also employed as a method to search for network architectures that are robust to adversarial attacks.

2.3 Particle Swarm Optimization (PSO)

PSO [47] regards the process of finding the optimal solution as the process of birds foraging. Specifically, individuals in the whole swarm search for the optimal solution cooperatively. Each individual in the swarm continuously changes its search direction by learning from its own experience and the experience of the whole swarm. The process of the PSO algorithm can be roughly divided into four steps:

1) **Initialization.** The PSO algorithm first sets the maximum number of iterations, the number of particles in the swarm and the maximum velocity of particles. Then, it randomly initializes the velocity of each particle in the velocity range and randomly initializes the position of each particle in the search space. The velocity v_i and position p_i of the ith particle can be expressed as (in an N-dimensional space):
$$p_i = (p_{i1}, p_{i2}, ..., p_{iN}), v_i = (v_{i1}, v_{i2}, ..., v_{iN}) \quad (1)$$
2) **Evaluation.** The PSO algorithm defines a fitness value to evaluate the goodness of the position of a particle, where the fitness value is determined by the optimization problem. $pbest_i$ represents the best position of the ith particle has experienced and $gbest$ represents the best position of the whole group have experienced.
3) **Iteratively updating.** The position and velocity of each particle are updated according to the formula (2) and (3):
$$\begin{aligned} v_i^{(k+1)} = \omega v_i^{(k)} &+ c_1 r_1 (pbest_i^{(k)} - p_i^{(k)}) \\ &+ c_2 r_2 (gbest^{(k)} - p_i^{(k)}) \end{aligned} \quad (2)$$
$$p_i^{(k+1)} = p_i^{(k)} + v_i^{(k+1)} \quad (3)$$
where $v_i^{(k)}$ and $p_i^{(k)}$ represent the velocity and position of the ith particle in the kth iteration. $pbest_i^{(k)}$ represents the $pbest$ of the ith particle in the kth iteration. $gbest^{(k)}$ represents the $gbest$ in the kth iteration. r_1 and r_2 are two random numbers in $(0, 1)$, c_1 and c_2 are the acceleration factors, ω is the inertia weight. After that, $pbest$ and $gbest$ are also updated. The update process will be repeated until the termination condition is reached.
4) **Output.** $gbest$ at the current iteration is outputted as the optimal solution.

Due to the gradient-free feature of the PSO algorithm, it is suitable to be used in black-box scenarios. In this work, the PSO algorithm is employed in *Effective-AATR* to search for the most effective adversarial samples within the given the number of queries.

3 Adversary Model

3.1 Adversary's Knowledge and Capability

We consider the probability-based black-box attack in this work, where the attacker knows nothing of the target model, but he is able to query the model with samples and obtain the classification probabilities of these query samples. The adversary is capable of generating adversarial samples through rotating and translating the benign samples, but the size of translation and rotation angle is limited (see Section 5 for more details).

3.2 Adversary's Goal

We consider two different attack goals in this work:

- **Query-efficiency goal** aims on achieving an adversarial attack (i.e., inducing a misclassification) with as few queries as possible, where *Efficient-AATR* is developed to achieve this goal.
- **Effectiveness goal** aims on achieving a higher misclassification probability within a given query budget, where *Effective-AATR* is proposed to achieve this goal.

4 Attack Methodologies

4.1 Overview

Recent studies [27, 48] have discovered that small transformations (e.g., rotation and translation) of the input image can greatly affect the output of the CNN[1]. This vulnerability gives the adversary an opportunity to generate physical adversarial samples through image transformations instead of adding perturbations.

1. The vulnerability may be attributed to two reasons: the ignorance of the Shannon-Nyquist sampling theorem [49, 50] and the photographer's biases on the datasets [51].

We propose two new physical adversarial attacks to satisfy the two goals described in Section 3. For an adversary whose goal is to achieve the attack with fewer queries, we propose *Efficient-AATR* to achieve the goal; For an adversary whose goal is to achieve a higher misclassification probability within a given number of queries, we propose *Effective-AATR* to achieve the goal. The details of the two attacks are described below.

4.2 Efficient-AATR: Efficient Adversarial Attack through Translation and Rotation

We first propose *Efficient-AATR* to achieve the query-efficiency goal. The process of *Efficient-AATR* is presented in **Algorithm 1**.

Algorithm 1 The process of *Efficient-AATR*

Input: the benign sample and its label (x, y); the target model f; the step size s (includes the step size of rotation s_1 and the step size of translation s_2); the maximum number of iterations T_1

Output: the adversarial sample x_{adv}

1: Initialize the iteration counter: $t \leftarrow 0$
2: **while** $f(x) = y$ and $t < T_1$ **do**
3: **for** $j = 1$ to 6 **do**
4: $x_j \leftarrow x + s \cdot d_j$ /*where d_j represents the direction vector of the 6 update directions*/
5: Calculate $pro(x_j)$ /*where $pro(x_j)$ represents the probability of x_j being classified correctly*/
6: **end for**
7: $x \leftarrow x_{min}$ /*where x_{min} denotes the sample with the minimum probability of correct classification in x_j*/
8: $t \leftarrow t + 1$
9: **end while**
10: $x_{adv} \leftarrow x$
11: **return** x_{adv}

To be more specific, we denote the update step of the angle of rotation as s_1 and denote the update step of horizontal (or vertical) translation as s_2. Then, the total update step for each iteration can be $(\pm s_1, \pm s_2, \pm s_2)$, i.e., there are 6 possible directions to update the adversarial sample. As presented in steps 3-8 of **Algorithm 1**, during each iteration, *Efficient-AATR* generates 6 samples along the 6 update directions and calculates the probabilities of them being classified correctly. The sample with the minimum probability of correct classification will be chosen as the desired updated sample[2]. The update process is carried out repeatedly until the generated adversarial sample finally crosses the decision boundary (i.e., the generated adversarial sample is misclassified by the target model.) or the algorithm reaches the maximum number of iterations T_1.

Efficient-AATR utilizes a greedy strategy to generate adversarial samples and stops when the generated sample causes misclassification. This allows it to achieve the attack with fewer queries and smaller transformations.

2. As a special case, if all the updated samples have higher probabilities than the current sample, the current samples will be updated with some random transformations to prevent the update process from sticking in an infinite loop.

4.3 Effective-AATR: Effective Adversarial Attack through Translation and Rotation

4.3.1 The process of Effective-AATR

Different from *Efficient-AATR* that stops when the generated sample is misclassified by the model, we further propose *Effective-AATR*, which aims at generating more effective adversarial samples within a given query budget. Specifically, *Effective-AATR* utilizes an adaptive particle swarm optimization (APSO) algorithm to search for better rotation angle and translation that make the target model misclassify the adversarial sample with higher confidence. The workflow of *Effective-AATR* is illustrated in Fig. 2.

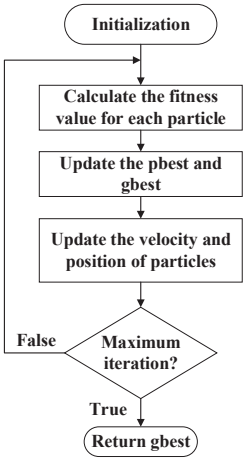


Fig. 2: The workflow of *Effective-AATR*

Concretely, the initialization of *Effective-AATR* (see **Algorithm 2**) randomly initializes numerous particles, including the initialization of position and velocity. The position of each particle p_i $(i = 1, ..., M)$ represents the transformation applied to the benign sample, which is three-dimensional (including the angle of rotation, horizontal translation and vertical translation). Besides, $pbest$ (denoted as p_i^*) and $gbest$ (denoted as p_{gb}) are also initialized through measuring fitness values, where the fitness value is defined as the probability of x with the transformation p_i being classified correctly. Then, *Effective-AATR* performs the search process in **Algorithm 3** to update the position and velocity of each particle iteratively according to the Eq. (2) and (3). After T_2 rounds of iteration, the final p_{gb} is obtained as the optimal transformation applied to the benign sample. The benign sample with the transformation p_{gb} is the optimal adversarial sample.

4.3.2 Adaptive Inertia Weight Strategy

The inertia weight ω in the *Effective-AATR* is critical to the convergence of the algorithm. When the search space dimension is large and the optimization problem is complicated, the algorithm often converges too early and falls into the local optimum. To mitigate this problem, we utilize an adaptive inertia weight strategy [52] in *Effective-AATR*, i.e., assigning different ω to different particles based on the fitness value of the particle:

$$\omega_i = \omega_{min} + (\omega_{max} - \omega_{min}) \times \frac{Rank_i}{M} \tag{4}$$

where $Rank_i$ refers to the ranking of the position of ith particle. Essentially, this method assigns smaller ω to

Algorithm 2 The initialization of *Effective-AATR*

Input: the number of particles in the whole swarm M; the benign sample and its label (x, y)

1: **for** each particle $i = 1$ to M **do**
2: Randomly initialize the position of the particle p_i
3: Randomly initialize the velocity of the particle v_i
4: Initialize $pbest$: $p_i^* \leftarrow p_i$
5: **end for**
6: Calculate $pro(x, p_i)$ /*where $pro(x, p_i)$ represents the probability of the sample (x with transformation p_i) being classified correctly*/
7: Initialize $gbest$: $p_{gb} \leftarrow pro(x, p_{min})$ /*where $pro(x, p_{min})$ is the minimum value from $pro(x, p_1)$ to $pro(x, p_M)$*/

Algorithm 3 The search process of *Effective-AATR*

Input: the acceleration factors c_1, c_2; random numbers r_1, r_2; the inertia weight ω; the number of iteration T_2; the number of particles in the swarm M

Output: the optimal adversarial sample x_{adv}

1: Initialize p_i employing the initialization algorithm of *Effective-AATR*
2: **for** $t = 1$ to T_2 **do**
3: **for** each particle $i = 1$ to M **do**
4: $v_i \leftarrow \omega v_i + c_1 r_1 (p_i^* - p_i) + c_2 r_2 (p_{gb} - p_i)$
5: $p_i \leftarrow p_i + v_i$
6: **if** $pro(x, p_i) < pro(x, p_i^*)$ **then**
7: $p_i^* \leftarrow p_i$
8: **end if**
9: **if** $pro(x, p_i) < pro(x, p_{gb})$ **then**
10: $p_{gb} \leftarrow p_i$
11: **end if**
12: **end for**
13: **end for**
14: $x_{adv} \leftarrow x$ with the adversarial transformation p_{gb}.
15: **return** x_{adv}

the particle with a high fitness value, which is helpful to perform an accurate local search of the current search area. Besides, it assigns larger ω to the particle with a low fitness value, which is helpful to get rid of the local minimum and facilitates the global search.

4.3.3 The Convergence of the Effective-AATR

In order to facilitate the analysis of the convergence of the *Effective-AATR*, we first simplify the algorithm to an one-dimensional, single particle setting. After that, we denote $c_1 r_1$, $c_2 r_2$, $c_1 r_1 pbest^{(k)} + c_2 r_2 gbest^{(k)}$ as φ_1, φ_2, φ_{pg} and fix them as constants. The update process of the adversarial transformation (denoted as particle p) in the *Effective-AATR* is simplified as:

$$v^{(k+1)} = \omega v^{(k)} - (\varphi_1 + \varphi_2) p^{(k)} + \varphi_{pg} \tag{5}$$

$$p^{(k+1)} = (1 - \varphi_1 - \varphi_2) p^{(k)} + \omega v^{(k)} + \varphi_{pg} \tag{6}$$

Then, Eq. (7) can be obtained by eliminating the velocity-related terms in Eq. (5) and Eq. (6):

$$p^{(k+1)} = (1 + \omega - \varphi_1 - \varphi_2) p^{(k)} - \omega p^{(k-1)} + \varphi_{pg} \tag{7}$$

and matrix form of Eq. (7) is:

$$\begin{bmatrix} p^{(k+1)} \\ p^{(k)} \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} p^{(k)} \\ p^{(k-1)} \\ 1 \end{bmatrix} \tag{8}$$

where

$$\mathbf{A} = \begin{bmatrix} 1 + \omega - \varphi_1 - \varphi_2 & -\omega & \varphi_{pg} \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{9}$$

the characteristic equation of matrix $\mathbf{A}$ is:

$$(1 - \lambda)(\lambda^2 - (\omega + 1 - \varphi_1 - \varphi_2)\lambda + \omega) = 0 \tag{10}$$

and the three roots of Eq. (10) can be solved:

$$e_1 = 1 \tag{11}$$

$$e_{2,3} = \frac{\omega + 1 - \varphi_1 - \varphi_2 \pm \sqrt{(\omega + 1 - \varphi_1 - \varphi_2)^2 - 4\omega}}{2} \tag{12}$$

Hence, $p^{(k)}$ and $v^{(k)}$ can be denoted as:

$$p^{(k)} = m_1 + m_2 e_2^k + m_3 e_3^k, v^{(k)} = n_1 e_2^k + n_2 e_3^k \tag{13}$$

where $m_{1,2,3}$ and $n_{1,2}$ are constants. The limit of $p^{(k)}$ and $v^{(k)}$ can be calculated:

$$\lim_{k \to \infty} p^{(k)} = m_1 + m_2 \lim_{k \to \infty} e_2^k + m_3 \lim_{k \to \infty} e_3^k \tag{14}$$

$$\lim_{k \to \infty} v^{(k)} = n_1 \lim_{k \to \infty} e_2^k + n_2 \lim_{k \to \infty} e_3^k \tag{15}$$

- when $\| e_2 \| > 1$ or $\| e_3 \| > 1$, $\lim_{k \to \infty} p^{(k)}$ and $\lim_{k \to \infty} v^{(k)}$ do not exist, the trajectory and velocity of the particle are divergent.
- when $\| e_2 \| < 1$ and $\| e_3 \| < 1$, $\lim_{k \to \infty} p^{(k)} = m_1$ and $\lim_{k \to \infty} v^{(k)} = 0$, the trajectory and velocity of the particle are convergent.
- when $max(\| e_2 \|, \| e_3 \|) = 1$, $\lim_{k \to \infty} p^{(k)} = m_1 + m_2 + m_3$ or $m_1 + m_2$ or $m_1 + m_3$, $\lim_{k \to \infty} v^{(k)} = n_1$ or n_2 or $n_1 + n_2$, the trajectory and velocity of the particle are convergent.

In conclusion, when the parameters (φ_1, φ_2 and ω) are set to meet the condition of $max(\| e_2 \|, \| e_3 \|) \leq 1$, the update process of the adversarial transformation is convergent.

Remark. we consider translation and rotation as the method to construct physical adversarial samples in this work. Other transformation methods (such as zooming and scaling) may also be capable of producing adversarial samples, we did not consider these methods because translation and rotation are the most common transformations in the physical world and they are already able to achieve the attack goal of the adversary.

5 Experimental Evaluation

We first evaluate the attack performance of our attacks with different hyperparameters. After that, we compare the attack performance of our attacks with state-of-the-art black-box adversarial attacks under mainstream defense methods.

TABLE 1: Average transformation of *Efficient-AATR*

Step size / Dataset	$s_1 = 2$ $s_2 = 0.2$	$s_1 = 4$ $s_2 = 0.4$	$s_1 = 6$ $s_2 = 0.6$	$s_1 = 8$ $s_2 = 0.8$	$s_1 = 10$ $s_2 = 1.0$
MNIST	$(9.97°, 1.03, 1.11)$	$(11.39°, 1.16, 1.35)$	$(12.10°, 1.57, 1.66)$	$(12.79°, 1.88, 1.87)$	$(13.46°, 2.01, 1.94)$
F-MNIST	$(11.57°, 0.37, 0.35)$	$(12.11°, 0.32, 0.47)$	$(14.43°, 0.48, 0.49)$	$(17.07°, 0.41, 0.50)$	$(18.44°, 0.38, 0.49)$
CIFAR-10	$(8.67°, 0.31, 0.33)$	$(13.58°, 0.32, 0.59)$	$(19.01°, 0.62, 0.67)$	$(23.32°, 0.71, 0.77)$	$(25.31°, 0.88, 0.75)$
ImageNet	$(5.05°, 0.61, 0.48)$	$(10.65°, 0.58, 0.91)$	$(15.55°, 0.81, 1.06)$	$(18.70°, 0.78, 1.21)$	$(21.10°, 1.51, 1.45)$

5.1 Experimental Setup

Datasets and target models. We use the LeNet-5, AlexNet, ResNet18 and ResNet50 as the target models on MNIST, Fashion-MNIST (F-MNIST), CIFAR-10 and ImageNet.

Hyperparameter settings. Each model is trained with 50 epochs[3]. The learning rate is set to 0.01 and the batch size is set to 128. The maximum value of rotation angle and translation ε_1 and ε_2 are limited to $(-30°, +30°)$ and $(-3, +3)$ pixel, respectively. For each attack, we randomly draw 1,000 samples from the test dataset and construct adversarial samples for them.

All experiments are implemented in Python and run on a 16-core Intel(R) Xeon(R) CPU E5-2620v4 @ 2.10GHz 16G machine.

5.2 Attack Performance Evaluation and Hyperparameters Analysis

5.2.1 Attack Performance of Efficient-AATR with Different Hyperparameters

Efficient-AATR aims on generating adversarial samples with fewer queries and smaller transformations. Similar to the state-of-the-art black-box adversarial attacks, we consider the following three metrics to evaluate *Efficient-AATR*:

- **Attack success rate (ASR):** the proportion of adversarial samples that lead to the misclassifications of the model to the total number of adversarial samples.
- **Average queries:** the average number of queries required to generate an adversarial sample.
- **Average transformation:** the average transformation required to construct an adversarial sample.

Specifically, we choose s_1 uniformly from 1 to 10 and s_2 uniformly from 0.1 to 1, T_1 is set according to Eq. (16). Then we perform *Efficient-AATR* with different combinations of these hyperparameters.

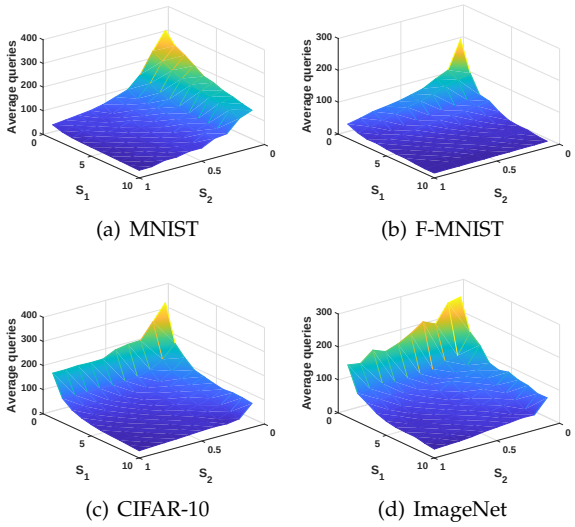
$$T_1 = \frac{\varepsilon_1}{s_1} + 2\frac{\varepsilon_2}{s_2} \tag{16}$$

The attack performance of *Efficient-AATR* with different hyperparameters is shown in Fig. 3 and 4. It indicates that the number of queries required decreases and the attack success rate increases with lager step sizes (s_1 and s_2). However, lager step sizes also cause larger average transformations as presented in TABLE 1, which decrease the stealthiness of the attack. There is a trade-off between the effectiveness and stealthiness of the attack.

5.2.2 Attack Performance of Effective-AATR with Different Hyperparameters

The goal of *Effective-AATR* is to generate more effective adversarial samples within a given query budget. Thus,

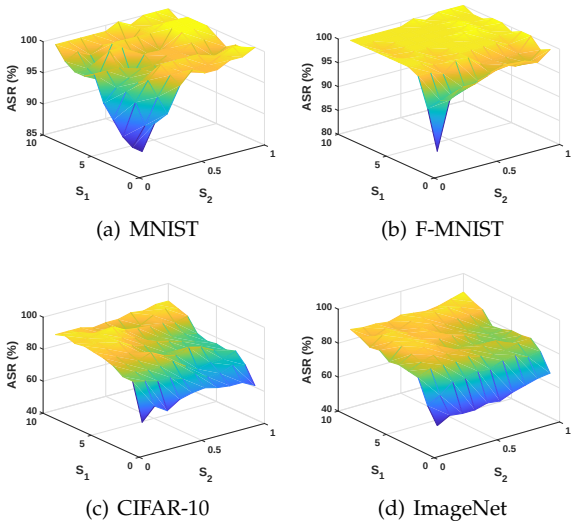
3. We adopt the pre-trained ResNet50 model from Pytorch.



(a) MNIST (b) F-MNIST

(c) CIFAR-10 (d) ImageNet

Fig. 3: Average required query of *Efficient-AATR* with different combinations of hyperparameters



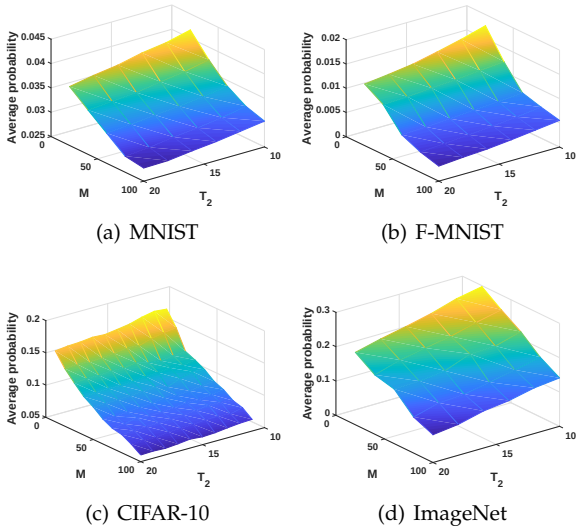
(a) MNIST (b) F-MNIST

(c) CIFAR-10 (d) ImageNet

Fig. 4: Attack success rate (ASR) of *Efficient-AATR* with different combinations of hyperparameters

the average probability that the adversarial sample being classified correctly is used as the metric to evaluate the effectiveness of the attack. Concretely, we choose M uniformly from 10 to 100 and T_2 uniformly from 10 to 20, and perform *Effective-AATR* with different combinations of hyperparameters.

The results in Fig. 5 indicate that *Effective-AATR* performs well on the four models by reducing the average



(a) MNIST (b) F-MNIST

(c) CIFAR-10 (d) ImageNet

Fig. 5: Attack performance of *Effective-AATR* with different combinations of hyperparameters

probabilities to almost zero[4]. Besides, there is a trade-off between the attack effect and the number of queries required: the increase of the number of queries required Q (where Q equals to $M \times T_2$) is conducive to improving the effectiveness of *Effective-AATR*.

In order to balance the trade-off between the effectiveness and stealthiness of the attack, in the following experiments, the hyperparameters of *Efficient-AATR* and *Effective-AATR* are set as follows: for *Efficient-AATR*, we set the step size of rotation s_1 to $5°$, the step size of translation s_2 (horizontal and vertical) to 0.5 pixel, the maximum number of iterations T_1 is set according to Eq. (16). For *Effective-AATR*, c_1 and c_2 are set to 2 and ω is set according to Eq. (4). T_2 is fixed to 10 and M is set to Q/T_2.

5.3 Attack Performance under Defense

In this subsection, we compare the attack performance of our proposed attacks with the state-of-the-art black-box adversarial attacks under mainstream defense methods.

5.3.1 Baseline Attacks

- *Meta Attack* [16] is a representative perturbation-based black-box adversarial attack. It employed meta-learning to train a meta attacker, which is used to estimate the gradient of the victim model.
- *Grid Search* [27] and *Worst-of-k* [27] are also methods to search for rotation and translation to construct physical adversarial samples. *Grid Search* exhaustively searches every possible parameter (rotation of angle and translations) in the parameter space until it finds the parameter that induces the model to misclassify the sample; *Worst-of-k* searches for the most effective adversarial sample in k random parameter combinations.

4. The average probability that the benign samples are classified correctly for MNIST, F-MNIST, CIFAR-10 and ImageNet are 0.9896, 0.9997, 0.8781 and 0.7767, respectively.

5.3.2 Attack Performance on Robust Models

Adversarial training is the most commonly used method to enhance the robustness of the model against adversarial attacks, which includes some adversarial samples into the training dataset and trains the model with these samples. In the context of our transformation-based adversarial attacks, an intuitive way to improve the robustness of the model is data augmentation, which augments the training process with some randomly transformed data.

Thus, for transformation-based adversarial attacks (including *Efficient-AATR*, *Effective-AATR*, *Worst-of-k* and *Grid Search*), we implement a data augmentation strategy during the training phase to obtain robust models. Each sample in the training dataset is augmented with a random transformed sample, where the maximum value of translation and rotation angle are the same as the settings mentioned in Section 5.1. For perturbation-based adversarial attack (*Meta Attack*), we employ a *PGD* adversarial training strategy during the training phase to obtain robust models. The *PGD* step is fixed to 10, the maximum perturbation is set to 0.3 for MNIST, 0.2 for F-MNIST, 0.03 for CIFAR-10 and ImageNet. After that, we evaluate the attack performance on these robust models.

For the attacker from a query-efficiency perspective, we evaluate the attack performance of *Efficient-AATR*, *Grid Search* and *Meta Attack* on robust models. Specifically, in order to unify the step size of *Grid Search* with *Efficient-AATR*, we consider 12 values for translations (horizontal and vertical) and 12 values for rotations, equally spaced by $5°$ and 0.5 pixel. The hyperparameters of *Meta Attack* are set as default in [16].

As presented in TABLE 2, compared with *Meta Attack* and *Grid Search*, *Efficient-AATR* achieves the same (or higher) ASR as the baseline attack with much fewer queries. It demonstrates that *Efficient-AATR* shows a significant improvement in the query-efficiency over the baseline attack. This is mainly because the greedy strategy used in *Efficient-AATR* makes it achieve the attack with fewer queries. Besides, the result also demonstrates that data augmentation is less effective in defending against *Efficient-AATR*.

For the attacker from an attack effectiveness perspective, because most perturbation-based adversarial attacks are considered from a query-efficiency perspective, we only compare *Effective-AATR* with *Worst-of-k*. The value of k in *Worst-of-k* is fixed to Q to ensure that the number of queries of *Worst-of-k* is the same as *Effective-AATR*.

As can be seen from Fig. 6, both of the two attacks can reduce the probability a lot: the probabilities of the four models decrease to below 0.20, below 0.15, below 0.30 and below 0.10 after 1000 queries. In comparison, within the same query budget, adversarial samples generated by *Effective-AATR* always have lower probabilities of correct classification than that generated by *Worst-of-k* . It is mainly because the PSO method used in *Effective-AATR* is more effective than the random search method used in *Worst-of-k*.

5.3.3 Evaluation on Detection-based Defense

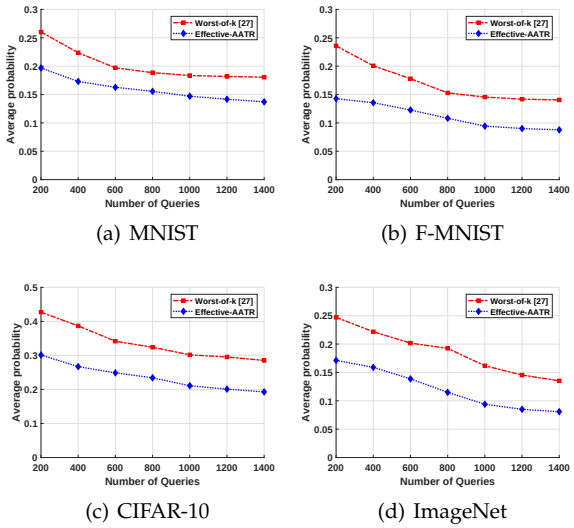
For detection-based defenses, we evaluate our attacks against the detection of Local Intrinsic Dimensionality (LID) [28], which is one of the most representative of detection-based defenses against adversarial attacks. The intuition

This article has been accepted for publication in IEEE Transactions on Big Data. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2022.3227318

TABLE 2: Attack performance of *Efficient-AATR* on robust models

Model and dataset	Attack method	ASR (%)	Avg. queries	Avg. transformation[1]	Avg. L_2 distance[2]
LeNet-5 (MNIST)	*Efficient-AATR*	81.43	**92.41**	$(16.57°, 2.59, 2.36)$	-
	Grid Search [27]	89.17	1011.40	$(18.45°, 2.18, 2.65)$	-
	Meta Attack [16]	68.15	4492.44	-	3.37
AlexNet (F-MNIST)	*Efficient-AATR*	89.30	**41.27**	$(19.90°, 1.24, 0.62)$	-
	Grid Search [27]	93.15	407.54	$(21.61°, 1.13, 1.27)$	-
	Meta Attack [16]	81.01	2853.87	-	2.07
ResNet18 (CIFAR-10)	*Efficient-AATR*	78.08	**75.90**	$(17.12°, 1.45, 1.29)$	-
	Grid Search [27]	80.19	867.18	$(19.63°, 1.31, 0.98)$	-
	Meta Attack [16]	83.76	1923.65	-	1.23
ResNet50 (ImageNet)	*Efficient-AATR*	86.01	**67.78**	$(12.76°, 0.83, 0.75)$	-
	Grid Search [27]	87.01	791.67	$(18.89°, 1.65, 1.35)$	-
	Meta Attack [16]	84.33	2149.50	-	2.07

[1] The three dimensions of the average transformation represent the absolute value of the rotation, horizontal translation and vertical translation, respectively.
[2] The average L_2-norm distance between the adversarial sample and the benign sample. Since the adversarial samples of *Efficient-AATR* and *Grid Search* [27] are generated through rotation and translation, their Avg. L_2 is not considered.



(a) MNIST (b) F-MNIST

(c) CIFAR-10 (d) ImageNet

Fig. 6: Attack performance of *Effective-AATR* on robust models

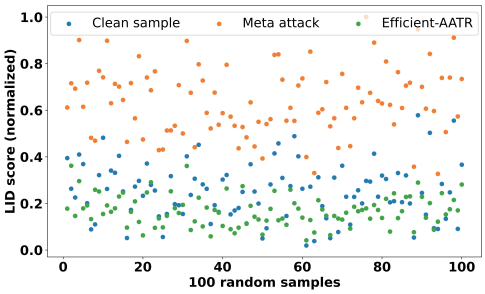


Fig. 7: LID scores of 100 normal samples, adversarial samples generated by *Meta Attack* and adversarial samples generated by *Efficient-AATR*.

of LID is that the LID features of adversarial samples are always different from normal samples. Therefore, clean and adversarial samples can be distinguished by calculating and analyzing the distribution of their LID features.

Without loss of generality, we select the ResNet50 model on ImageNet as the victim model and compute the LID features of clean samples, adversarial samples generated by *Meta Attack* and adversarial samples generated by *Efficient-AATR* (the evaluation on *Effective-AATR* and other datasets give the same conclusion). The distributions of their LID features are illustrated in Fig. 7. It indicates that LID is effective in distinguishing perturbation-based adversarial samples from clean samples. The LID features of these adversarial samples are significantly different from that of clean samples. However, the LID features of adversarial samples generated by our attack are extremely similar to that of clean samples, which enable our adversarial samples to bypass the detection. It is mainly because our adversarial samples are generated in a more natural way through transformations instead of adding perturbations, and we believe our attacks can evade other detection techniques as well.

5.3.4 Robustness against Image Compression

Image compression [34–36] is one of the most commonly used input preprocessing defenses against adversarial attacks. In this work, we adopt the image compression defense method used in [34] to evaluate the robustness of our attacks. Specifically, we select 1000 adversarial samples generated and perform image compression on them to calculate the proportion of samples remaining adversarial.

As illustrated in Fig. 8, a larger proportion of adversarial samples generated by *Meta Attack* turn into benign samples with the decrease of compression ratio. However, as for *Efficient-AATR* and *Grid Search*, the effectiveness of the attack is slightly decreased, most of the adversarial samples still remain adversarial. Moreover, in terms of *Effective-AATR* and *Worst-of-k*, almost all adversarial samples generated by these two attacks still remain adversarial after image compression. It demonstrates that image compression is effective in defending against perturbation-based adversarial attacks such as *Meta Attack*, but it fails to defeat transformation-based adversarial attack such as our attacks and attacks proposed in [27].

5.3.5 Robustness against Image Transformation

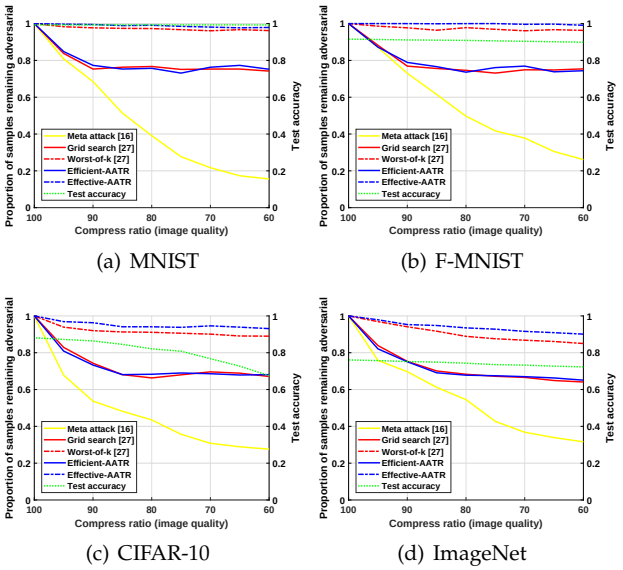
Tian et al. [37] proposed to pre-process the input images through random image transformations before feeding them to the model. It can reduce the model sensitivity to adversarial perturbations and make the attack ineffective. We follow the work [37] and evaluate the effectiveness of our attacks

This article has been accepted for publication in IEEE Transactions on Big Data. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TBDATA.2022.3227318

TABLE 3: ASR (%) under the defense of random transformation

Attack method \ Transformation range		$tr_1 \in (-4, 4)$ $tr_2 \in (-0.4, 0.4)$	$tr_1 \in (-8, 8)$ $tr_2 \in (-0.8, 0.8)$	$tr_1 \in (-12, 12)$ $tr_2 \in (-1.2, 1.2)$	$tr_1 \in (-16, 16)$ $tr_2 \in (-1.6, 1.6)$	$tr_1 \in (-20, 20)$ $tr_2 \in (-2, 2)$
MNIST	*Efficient-AATR*	75.6	63.5	61.7	53.9	57.1
	Grid Search [27]	30.3	41.4	29.8	42.1	34.3
	Effective-AATR	**99.1**	**89.8**	**83.7**	**84.9**	**91.3**
	Worst-of-k [27]	85.7	80.1	73.8	78.2	69.7
	Meta Attack [16]	38.7	31.2	30.1	34.0	38.1
F-MNIST	*Efficient-AATR*	67.9	42.6	50.9	52.2	51.9
	Grid Search [27]	26.7	38.1	40.2	44.8	42.7
	Effective-AATR	**98.9**	**89.5**	**85.1**	**87.0**	**92.4**
	Worst-of-k [27]	88.7	75.0	72.1	77.9	75.1
	Meta Attack [16]	41.6	37.7	38.2	35.1	37.9
CIFAR-10	*Efficient-AATR*	53.5	37.7	42.2	56.7	57.0
	Grid Search [27]	21.5	18.4	22.7	25.4	28.1
	Effective-AATR	**85.1**	**79.7**	**76.1**	**75.3**	**79.1**
	Worst-of-k [27]	79.8	74.1	69.0	72.1	69.7
	Meta Attack [16]	29.1	26.3	37.0	35.7	40.2
ImageNet	*Efficient-AATR*	60.2	59.7	55.3	51.1	53.5
	Grid Search [27]	31.1	28.6	27.3	25.4	27.1
	Effective-AATR	**87.1**	**81.0**	**78.5**	**76.6**	**77.4**
	Worst-of-k [27]	80.2	77.5	76.1	73.0	74.8
	Meta Attack [16]	45.1	39.3	34.4	38.9	40.3

* tr_1 and tr_2 denotes the range of the random rotation and random translation, respectively.



(a) MNIST (b) F-MNIST

(c) CIFAR-10 (d) ImageNet

Fig. 8: Robustness evaluation against image compression

under random image transformations. Specifically, we select 1000 adversarial samples generated from the five attacks (including *Efficient-AATR*, *Grid Search*, *Effective-AATR*, *Meta Attack* and *Worst-of-k*) and perform random transformations on them before feeding them to the models.

As presented in TABLE 3, random transformation is effective in defending against *Grid Search* and *Meta Attack*. The attack success rates of these two attacks drop significantly. Besides, random transformation also reduces the attack success rates of *Efficient-AATR* and *Worst-of-k* to a certain extent. However, *Effective-AATR* is more robust against the defense of random transformation, the attack success rates of *Effective-AATR* on the four datasets still remain high (83.7%, 85.1%, 76.1% and 78.5%) after random transformation.

From the results we have obtained, it can be concluded that our attacks are more robust than baseline attacks against mainstream defenses including detection of Local Intrinsic Dimensionality (LID), model robustness enhancement and input preprocessing defenses. These defenses are far from a solution to defending against our transformation-based attacks, more effective countermeasures still require further research.

6 Conclusions

In this paper, we explore a natural way to generate adversarial samples in the physical world, i.e., generating adversarial samples through image transformations. Specifically, we propose two attack algorithms to satisfy the different goals of the adversary: for the attacker from a query-efficiency perspective, we propose *Efficient-AATR*. It employs a greedy strategy to update adversarial samples in order to generate them with fewer queries and smaller transformations; For the attacker who aims on achieving the most effective attack within a given query budget, we propose *Effective-AATR*. It utilizes an adaptive particle swarm optimization algorithm (APSO) to generate more effective adversarial samples. Finally, we conduct experiments to demonstrate the superiority of our attacks compared with state-of-the-art adversarial attacks under mainstream defenses.

Acknowledgment

This work is supported by the National Natural Science Foundation of China under Grants 62020106013, 61972454, 61802051, 61772121, and 61728102, Sichuan Science and Technology Program under Grants 2020JDTD0007 and 2020YFG0298, the Fundamental Research Funds for Chinese Central Universities under Grant ZYGX2020ZB027, Singapore Ministry of Education (MOE) AcRF Tier 2 MOE-T2EP20121-0006 and AcRF Tier 1 RS02/19.

References

[1] G. Xu, H. Li, H. Ren, K. Yang, and R. H. Deng, "Data security issues in deep learning: attacks, countermeasures, and opportunities," *IEEE Communications Magazine*, vol. 57, no. 11, pp. 116–122, 2019.

[2] G. Xu, H. Li, S. Liu, K. Yang, and X. Lin, "Verifynet: Secure and verifiable federated learning," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 911–926, 2019.

[3] G. Xu, H. Li, Y. Zhang, S. Xu, J. Ning, and R. Deng, "Privacy-preserving federated deep learning with irregular users," *IEEE Transactions on Dependable and Secure Computing*, vol. 19, no. 2, pp. 1364–1381, 2022.

[4] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[5] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *Proceedings of ICLR*, 2018.

[6] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proceedings of Asia CCS*, 2017, pp. 506–519.

[7] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," in *Proceedings of ICLR*, 2016.

[8] Y. Shi, Y. Han, Q. Hu, Y. Yang, and Q. Tian, "Query-efficient black-box adversarial attack with customized iteration and sampling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.

[9] Z. Yuan, J. Zhang, Y. Jia, C. Tan, T. Xue, and S. Shan, "Meta gradient adversarial attack," in *Proceedings of ICCV*, 2021, pp. 7748–7757.

[10] C. Ma, L. Chen, and J.-H. Yong, "Simulating unknown target models for query-efficient black-box attacks," in *Proceedings of CVPR*, 2021, pp. 11 835–11 844.

[11] J. Zou, Z. Pan, J. Qiu, Y. Duan, X. Liu, and Y. Pan, "Making adversarial examples more transferable and indistinguishable," *Proceedings of AAAI*, 2022.

[12] W. Brendel, J. Rauber, and M. Bethge, "Decision-based adversarial attacks: Reliable attacks against black-box machine learning models," in *Proceedings of ICLR*, 2018.

[13] W. Chen, Z. Zhang, X. Hu, and B. Wu, "Boosting decision-based black-box adversarial attacks with random sign flip," in *Proceedings of ECCV*, 2020, pp. 276–293.

[14] Y. Dong, H. Su, B. Wu, Z. Li, W. Liu, T. Zhang, and J. Zhu, "Efficient decision-based black-box adversarial attacks on face recognition," in *Proceedings of CVPR*, 2019, pp. 7714–7722.

[15] J. Chen, M. I. Jordan, and M. J. Wainwright, "Hopskipjumpattack: A query-efficient decision-based attack," in *Proceedings of S&P*, 2020, pp. 1277–1294.

[16] J. Du, H. Zhang, J. T. Zhou, Y. Yang, and J. Feng, "Query-efficient meta attack to deep neural networks," in *Proceedings of ICLR*, 2020.

[17] X.-C. Li, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Decision-based adversarial attack with frequency mixup," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1038–1052, 2022.

[18] T. Maho, T. Furon, and E. Le Merrer, "Surfree: a fast surrogate-free black-box attack," in *Proceedings of CVPR*, 2021, pp. 10 430–10 439.

[19] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *Proceedings of ICLR*, 2017.

[20] T. B. Brown, D. Mané, A. Roy, M. Abadi, and J. Gilmer, "Adversarial patch," *arXiv preprint arXiv:1712.09665*, 2017.

[21] K. Eykholt, I. Evtimov, E. Fernandes, B. Li, A. Rahmati, C. Xiao, A. Prakash, T. Kohno, and D. Song, "Robust physical-world attacks on deep learning visual classification," in *Proceedings of CVPR*, 2018, pp. 1625–1634.

[22] K. Xu, G. Zhang, S. Liu, Q. Fan, M. Sun, H. Chen, P.-Y. Chen, Y. Wang, and X. Lin, "Adversarial t-shirt! evading person detectors in a physical world," in *Proceedings of ECCV*, 2020, pp. 665–681.

[23] A. Sayles, A. Hooda, M. Gupta, R. Chatterjee, and E. Fernandes, "Invisible perturbations: Physical adversarial examples exploiting the rolling shutter effect," in *Proceedings of CVPR*, 2021, pp. 14 666–14 675.

[24] R. Duan, X. Mao, A. K. Qin, Y. Chen, S. Ye, Y. He, and Y. Yang, "Adversarial laser beam: Effective physical-world attack to dnns in a blink," in *Proceedings of CVPR*, 2021, pp. 16 062–16 071.

[25] A. Gnanasambandam, A. M. Sherman, and S. H. Chan, "Optical adversarial attack," in *Proceedings of ICCV*, 2021, pp. 92–101.

[26] Y. Zhong, X. Liu, D. Zhai, J. Jiang, and X. Ji, "Shadows can be dangerous: Stealthy and effective physical-world adversarial attack by natural phenomenon," in *Proceedings of CVPR*, 2022.

[27] E. Logan, T. Brandon, T. Dimitris, S. Ludwig, and M. Aleksander, "A rotation and a translation suffice: fooling cnns with simple transformations," *arXiv preprint arXiv:1712.02779*, 2018.

[28] X. Ma, B. Li, Y. Wang, S. M. Erfani, S. Wijewickrema, G. Schoenebeck, M. E. Houle, D. Song, and J. Bailey, "Characterizing adversarial subspaces using local intrinsic dimensionality," in *Proceedings of ICLR*, 2018.

[29] S. Ma and Y. Liu, "Nic: Detecting adversarial samples with neural network invariant checking," in *Proceedings of NDSS*, 2019.

[30] G. Cohen, G. Sapiro, and R. Giryes, "Detecting adversarial samples using influence functions and nearest neighbors," in *Proceedings of CVPR*, 2020, pp. 14 453–14 462.

[31] P. Yang, J. Chen, C.-J. Hsieh, J.-L. Wang, and M. Jordan, "Ml-loo: Detecting adversarial examples with feature attribution," in *Proceedings of AAAI*, vol. 34, no. 04, 2020, pp. 6639–6647.

[32] M. Yin, S. Li, Z. Cai, C. Song, M. S. Asif, A. K. Roy-Chowdhury, and S. V. Krishnamurthy, "Exploiting multi-object relationships for detecting adversarial attacks in complex scenes," in *Proceedings of ICCV*, 2021, pp. 7858–7867.

[33] K. M. Carter, R. Raich, and A. O. Hero III, "On local intrinsic dimension estimation and its applications," *IEEE Transactions on Signal Processing*, vol. 58, no. 2, pp. 650–663, 2009.

[34] C. Guo, M. Rana, M. Cisse, and L. van der Maaten, "Countering adversarial images using input transformations," in *Proceedings of ICLR*, 2018.

[35] A. E. Aydemir, A. Temizel, and T. T. Temizel, "The effects of jpeg and jpeg2000 compression on attacks using adversarial examples," *arXiv preprint arXiv:1803.10418*, 2018.

[36] G. K. Dziugaite, Z. Ghahramani, and D. M. Roy, "A

study of the effect of jpg compression on adversarial images," *arXiv preprint arXiv:1608.00853*, 2016.

[37] S. Tian, G. Yang, and Y. Cai, "Detecting adversarial examples through image transformation," in *Proceedings of AAAI*, 2018.

[38] J. Cui, S. Liu, L. Wang, and J. Jia, "Learnable boundary guided adversarial training," in *Proceedings of ICCV*, 2021, pp. 15 721–15 730.

[39] G. Jin, X. Yi, W. Huang, S. Schewe, and X. Huang, "Enhancing adversarial training with second-order statistics of weights," in *Proceedings of CVPR*, 2022.

[40] Y. Dong, K. Xu, X. Yang, T. Pang, Z. Deng, H. Su, and J. Zhu, "Exploring memorization in adversarial training," *Proceedings of ICLR*, 2022.

[41] C. Zhu, Y. Cheng, Z. Gan, S. Sun, T. Goldstein, and J. Liu, "Freelb: Enhanced adversarial training for natural language understanding," in *Proceedings of ICLR*, 2020.

[42] A. T. Bui, T. Le, Q. H. Tran, H. Zhao, and D. Phung, "A unified wasserstein distributional robustness framework for adversarial training," in *Proceedings of ICLR*, 2022.

[43] T. Pang, K. Xu, Y. Dong, C. Du, N. Chen, and J. Zhu, "Rethinking softmax cross-entropy loss for adversarial robustness," in *Proceedings of ICLR*, 2020.

[44] A. Bui, T. Le, H. Zhao, P. Montague, O. deVel, T. Abraham, and D. Phung, "Improving adversarial robustness by enforcing local and global compactness," in *Proceedings of ECCV*, 2020, pp. 209–223.

[45] M. Guo, Y. Yang, R. Xu, Z. Liu, and D. Lin, "When nas meets robustness: In search of robust architectures against adversarial attacks," in *Proceedings of CVPR*, 2020, pp. 631–640.

[46] X. Xie, Y. Liu, Y. Sun, G. G. Yen, B. Xue, and M. Zhang, "Benchenas: A benchmarking platform for evolutionary neural architecture search," *IEEE Transactions on Evolutionary Computation*, 2022.

[47] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the Sixth International Symposium on Micro Machine and Human Science*, 1995, pp. 39–43.

[48] A. Azulay and Y. Weiss, "Why do deep convolutional networks generalize so poorly to small image transformations?" *Journal of Machine Learning Research*, vol. 20, no. 184, pp. 1–25, 2019.

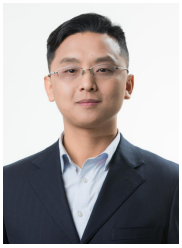
[49] H. D. Luke, "The origins of the sampling theorem," *IEEE Communications Magazine*, vol. 37, no. 4, pp. 106–108, 1999.

[50] A. J. Jerri, "The shannon sampling theoremłits various extensions and applications: A tutorial review," in *Proceedings of the IEEE*, vol. 65, no. 11, 1977, pp. 1565–1596.

[51] A. Torralba and A. A. Efros, "Unbiased look at dataset bias," in *Proceedings of CVPR*, 2011, pp. 1521–1528.

[52] B. Panigrahi, V. R. Pandi, and S. Das, "Adaptive particle swarm optimization approach for static and dynamic economic load dispatch," *Energy conversion and management*, vol. 49, no. 6, pp. 1407–1415, 2008.

Wenbo Jiang received his B.S. degree in information security from University of Electronic Science and Technology of China (UESTC) in 2017. Currently, he is a PhD student at the School of Computer Science and Engineering, University of Electronic Science and Technology of China (UESTC), China. His research interests include adversarial machine learning and model extraction attacks.



Hongwei Li (M'12CSM'18) is currently the Head and a Professor at Department of Information Security, School of Computer Science and Engineering, University of Electronic Science and Technology of China. He received the Ph.D. degree from University of Electronic Science and Technology of China in June 2008. He worked as a Postdoctoral Fellow at the University of Waterloo from October 2011 to October 2012. His research interests include network security and applied cryptography. He is the Senior Member of IEEE, the Distinguished Lecturer of IEEE Vehicular Technology Society.

Guowen Xu is currently a Research Fellow with Nanyang Technological University, Singapore. He received the PhD degree from the University of Electronic Science and Technology of China (UESTC) in 2020. He is the recipient of the Best Paper Award of the 26th IEEE International Conference on Parallel and Distributed Systems (ICPADS 2020) , the Best Student Paper Award from the Sichuan Province Computer Federation, and the 40th IEEE Conference on Computer Communications (INFOCOM 2021) Student Conference Award.

Tianwei Zhang is an assistant professor in School of Computer Science and Engineering, at Nanyang Technological University. His research focuses on computer system security. He is particularly interested in security threats and defenses in machine learning systems, autonomous systems, computer architecture and distributed systems. He received his Bachelor's degree at Peking University in 2011, and the Ph.D degree in at Princeton University in 2017.

Rongxing Lu (S'09-M'11-SM'15-F'21) is currently an associate professor at the Faculty of Computer Science (FCS), University of New Brunswick (UNB), Canada. He was awarded the most prestigious "Governor General's Gold Medal", when he received his PhD degree from the Department of Electrical & Computer Engineering, University of Waterloo, Canada, in 2012; and won the 8th IEEE Communications Society (ComSoc) Asia Pacific (AP) Outstanding Young Researcher Award, in 2013. He is presently an IEEE Fellow. Dr. Lu currently serves as the Vice-Chair (Publication) of IEEE ComSoc CIS-TC. Dr. Lu is the Winner of 2016-17 Excellence in Teaching Award, FCS, UNB.