A Physically Realizable Adversarial Attack Method Against SAR Target Recognition Model

Fan Zhang, *Senior Member, IEEE*, Yameng Yu, *Student Member, IEEE*, Fei Ma, *Member, IEEE*, and Yongsheng Zhou, *Member, IEEE*

***Abstract*—Deep learning has shown remarkable proficiency in tasks related to synthetic aperture radar (SAR) interpretation. However, several studies have highlighted the inherent vulnerability of deep neural networks when faced with deliberately constructed adversarial examples (AEs). Current SAR adversarial attack research only focuses on generating AEs in the image domain, without considering SAR imaging systems. This approach can lead to two issues. First, the generated SAR AEs lack visual coherence. Second, they cannot be obtained as corresponding attack images through real SAR imaging systems. In this article, we propose a physically realizable SAR adversarial attack method, which includes two submodules based on optimization methods: Target perturbation generation and background perturbation generation. The former module utilizes attention mechanisms to extract the target regions from SAR images, followed by selecting the optimal small regions for perturbation placement. The latter module, on the other hand, utilizes scattering models to generate realistic scatterer images as perturbations, which are then optimized to identify the optimal position in the background regions. The individual attack performance of these two attack modules on five well-established SAR automatic target recognition models is demonstrated to be highly effective. Moreover, the combination of these attack modules achieves a fooling rate of approximately 90% and demonstrates superior adversarial transferability. The experimental results of this study provide an effective foundation for physical realization of SAR AEs.**

***Index Terms*—Adversarial example (AE), automatic target recognition (ATR), physical attack, synthetic aperture radar (SAR).**

I. Introduction

DUE to the imaging ability of day-and-night and weather independence, synthetic aperture radar (SAR) has been widely used for remote sensing for more than 30 years. It plays a significant role in the geographical survey, climate change research, environment monitoring, military information processing, and other applications [1], [2], [3]. With the wide applications of SAR in the remote sensing field, target information extraction from SAR data has become a hot research topic, especially the automatic target recognition (ATR).

In recent years, there has been significant progress in deep learning technology, particularly in the field of computer vision. As a result, an increasing number of studies have begun to explore the application of deep learning in SAR ATR [4], [5], [6]. Chen et al. [7] achieved remarkable recognition performance in SAR ATR using a fully convolutional network, achieving an accuracy rate of 99%. Li et al. [8] also proposed a deep neural network (DNN) called DeepSAR-Net, which achieved an accuracy rate of 98.36% by incorporating multiple convolutional layers, normalization layers, and pooling layers, along with the ReLU activation function. In addition, Wang et al. [9] introduced SSF-Net, which demonstrated an exceptionally high accuracy rate of 99.55%. Chen et al. [10] presented three strategies for network compression and acceleration in DNNs for SAR ATR, addressing the challenges of long training periods and large network sizes. These strategies achieve a 40× compression of the networks without loss of accuracy and reduce the number of multiplications by 15×. Zhang et al. [11] proposed an improved DNN for SAR ATR that addresses the limited sample issue through feature augmentation and ensemble learning strategies. By concatenating cascaded features from selected convolutional layers and utilizing the AdaBoost rotation forest classifier, the proposed method achieves a 20% improvement in recognition accuracy, even with only ten training samples per class. In the future, SAR ATR algorithms will fully leverage the potential of deep learning technology through continuous optimization and iteration of network depth, parameter quantities, and module utilization.

However, research has uncovered security vulnerabilities in DNNs. Initially, Szegedy et al. [12] identified two intriguing properties. Their findings indicate that semantic information in a neural network is conveyed not by individual neurons, but rather by the network's holistic representation. Subsequently, they proposed a perturbation method that takes advantage of the nonlinear nature of neural networks, rendering it imperceptible to human observers. However, when this perturbation is added to the input, DNNs confidently make incorrect judgments. This behavior of fooling DNNs by adding perturbations is called adversarial attack, and the modified images are referred to as adversarial examples (AEs). In contrast, Goodfellow et al. [13] argue that the linear behavior of neural networks alone is sufficient to induce AEs. Building on this insight, they proposed

Manuscript received 29 January 2024; revised 14 May 2024 and 16 June 2024; accepted 24 June 2024. Date of publication 28 June 2024; date of current version 12 July 2024. This work was supported by the National Natural Science Foundation of China under Grant 62271034. *(Corresponding author: Fei Ma.)*

Fan Zhang is with the College of Information Science and Technology and the Interdisciplinary Research Center for Artificial Intelligence, Beijing University of Chemical Technology, Beijing 100029, China (e-mail: zhangf@mail.buct.edu.cn).

Yameng Yu, Fei Ma, and Yongsheng Zhou are with the College of Information Science and Technology, Beijing University of Chemical Technology, Beijing 100029, China (e-mail: 15190950316@163.com; mafei@mail.buct.edu.cn; zhyosh@mail.buct.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2024.3420690

the fast gradient sign method (FGSM), which modifies the input in the direction of the opposite gradient, consequently resulting in misclassification. Then a series of gradient-based adversarial attack methods has been introduced, such as iterative FGSM [14], momentum iterative FGSM [15], and projected gradient descent (PGD) [16].

Due to limitations in physical world, such as distance, viewing angles, and lighting conditions, techniques exhibiting excellent attack performance in digital simulations tend to yield suboptimal results in real world. In order to carry out adversarial attack in the physical world, Sharif et al. [17] proposed a method targeted at face recognition systems. Attackers can wear eyeglass frames that have adversarial perturbations in order to evade face detection. Brown et al. [18] introduced adversarial patch, which can be printed and placed near the target, misleading recognition software. However, adversarial patch exhibits an anomalous appearance. Duan et al. [19] took into account the visual plausibility of adversarial patch and proposed AdvCAM, which improved the stealthiness of perturbations in the physical world. Liu et al. [20] introduced a generative adversarial network (GAN) called perceptually sensitive GAN. The network exploits the target model's sensitivity to adversarial patch by incorporating an attention mechanism to predict critical regions for patch placement. As a result, the method produces AEs that are not only more realistic, but also more aggressive.

Existing research has started to focus on the interpretability of SAR ATR [21], and there are also security risks associated with SAR ATR models [22], [23]. Li et al. [24] conducted a study on adversarial attacks in remote sensing image recognition. Their findings revealed that DNNs trained on SAR images are susceptible to AEs. Zhang et al. [25] further proposed an adversarial attack method by combining the characteristics of SAR images. This method, based on the C&W [26] algorithm, results in higher fooling rate, as well as lower perturbation intensity and coverage range. Peng et al. [27] proposed the Speckle-Variant Attack, which cleverly converts SAR speckle noise into adversarial perturbation. Xia et al. [28] proposed a systematic SAR perturbation generation method called SAR-PeGA, which achieves excellent attack performance by utilizing two-dimensional phase modulation interference. These studies demonstrate the existence of various adversarial perturbations in SAR images, posing a threat to the security of SAR ATR models.

The mentioned SAR adversarial attacks mainly focus on the image domain, involving fine-tuning at the pixel level. However, compared to physical adversarial attacks on optical images, SAR encounters more challenges. Specifically, when captured at different pitch and azimuth angles, optical images exhibit minimal changes, while SAR images show notable differences under such variations. In addition, methods like printing can be used to simulate adversarial perturbations in optical images, but achieving the same effect for SAR adversarial perturbations through imaging proves challenging. Therefore, successfully executing SAR adversarial attacks necessitates a comprehensive understanding of SAR image characteristics.

Due to the imaging characteristics of SAR, attempting to mimic optical methods, where adversarial perturbations are first computed and then optically implemented, is not feasible. Therefore, we focus on the physical realizability and propose a comprehensive method for generating SAR AEs. Our approach starts by considering the addition of adversarial perturbations on the target surface. Building upon the work in [25], we further reduce the range of perturbations. Simultaneously, we relax constraints on perturbation intensity, allowing perturbations to be visible. Moreover, the noise-rich background areas commonly found in SAR images can also be leveraged. Following an optical camouflage strategy, we envision adding strong scatterers to the background. The overall method comprises the following two modules.

1) Target perturbation generation based on SAR sticker. Due to the fact that this adversarial perturbation ultimately appears as a small sticker on the image, we refer to it as the SAR sticker. Initially, an attention mechanism is applied to position the sticker in the image region of greatest concern to SAR ATR models. Following this, optimization algorithms are utilized to iteratively improve the appearance of the sticker, thereby improving attack performance.
2) Background perturbation generation based on strong scatterer. First, a simulation of strong scatterer image is performed based on the radar imaging parameters of the input dataset. Subsequently, the particle swarm optimization (PSO) algorithm is utilized to search for the optimal arrangement position in the background region, aiming to achieve the desired attack performance.

The two aforementioned modules generate two forms of adversarial perturbations, namely, SAR sticker and strong scatterer. SAR sticker can be implemented by placing special reflective materials on the target surface. On the other hand, strong scatterer can be achieved by placing scatterers in the target's surroundings, such as corner reflectors. The main contributions are given as follows.

1) We propose a comprehensive adversarial attack method specifically designed for SAR ATR models, which offers an initial solution to the challenge of implementing SAR adversarial perturbations in real scenarios.
2) Our approach generates adversarial perturbations in both the target and background regions of SAR images. These perturbations can be implemented through the use of corresponding imaging materials, providing a solid foundation for enabling intelligent camouflage of targets in practical applications.
3) A fooling rate exceeding 90% was achieved when attacking five well-established SAR ATR models. In addition, the method demonstrates good performance for adversarial transferability.

The rest of this article is organized as follows. Section II introduces the proposed method, which includes the target perturbation generation module employing SAR sticker and the background perturbation generation module utilizing strong scatterers. Section III presents the individual attack effects of each module on five well-established SAR ATR models, as well as their combined attack performances. Section IV delves into

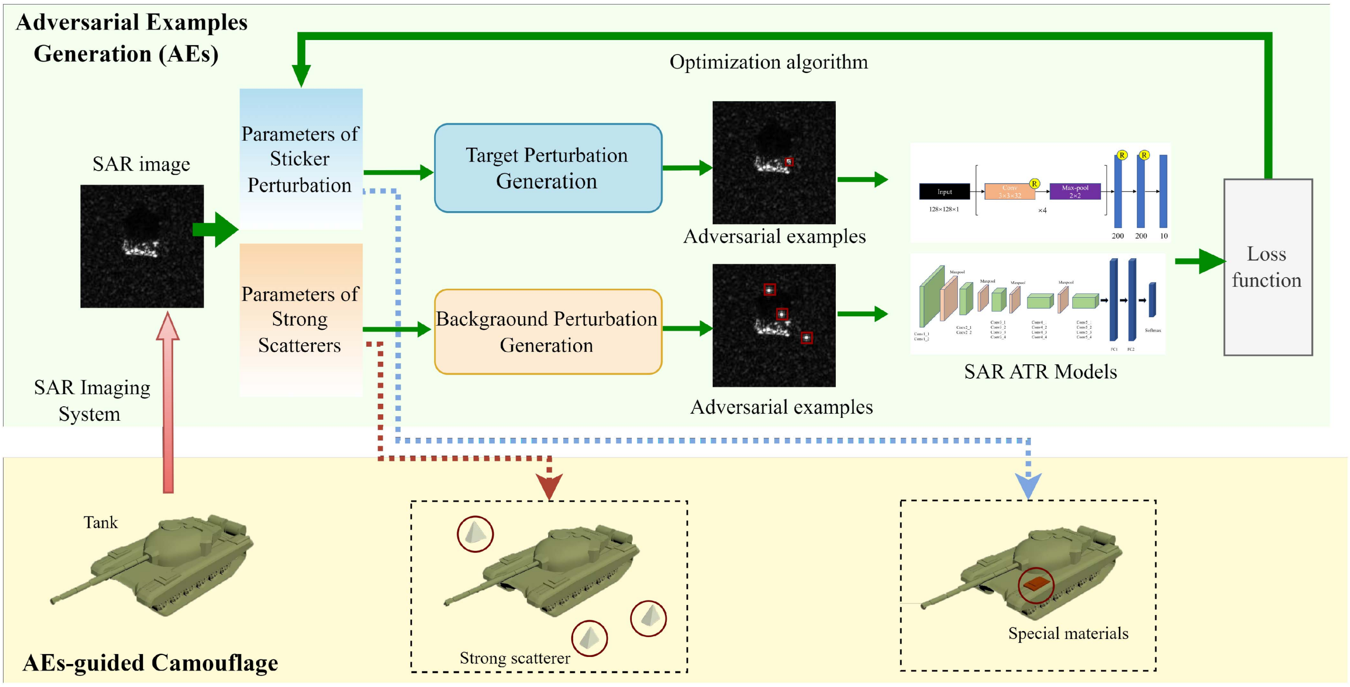


Fig. 1. Illustration of our physically realizable SAR adversarial attack method.

the stability and universality of our method and compares various adversarial attack strategies. Finally, Section V concludes this article.

II. Proposed Method

This section presents the framework of our physically realizable SAR adversarial attack method, illustrated in Fig. 1. Two perturbation generation modules have been devised. One generates SAR sticker in target surface, while the other generates strong scatterer images in target's surroundings. These perturbations simulate real-world situations, utilizing special reflective materials and corner reflectors for intelligent camouflage of targets.

It is important to note that the two perturbation generation modules depicted in Fig. 1 subsequently employ optimization algorithms. However, there are distinctions between the two. For the sake of coherence in the narrative, we have integrated the specific optimization algorithms into the introductions of each perturbation generation module.

A. Problem Formulation

Given an input image $x \in \mathbb{R}^{h\times w}$ with its class label t_0 and a well-trained DNN classifier $F : \mathbb{R}^{h\times w} \rightarrow \{1, \ldots, m\}$, it follows that $F(x) \rightarrow t_0$. Here, h and w represent the height and width of the input image, respectively, and m denotes the number of classes of the classifier. Our objective is to generates AEs $x' = x + \delta$ by strategically placing adversarial perturbations δ onto specific regions of the input image x, thereby causing misclassification $F(x') \rightarrow t, t \neq t_0$. Our objective function is defined as

$$\begin{aligned} \min \quad & Z(x') = F(x')_{t_0} - \max\{F(x')_t : t \neq t_0\} \\ \text{such} \quad & x' \in [0, 1]^{h\times w} \end{aligned} \tag{1}$$

where $F(x')_{t_0}$ represents the score of the image being classified as the true class, and $F(x')_t, t \neq t_0$ represents the scores of the image being classified as other classes. We simultaneously optimize both functions, lowering the score of the true class while increasing the score of the wrong class. Furthermore, to enhance the effectiveness of the AEs pixels, we impose constraints on the pixel range of perturbation δ.

In addition, during the design phase, we impose size limitations on perturbations to primarily focus on improving their attack performance. In this optimization process, we consider objective functions such as content loss, style loss, and smoothness loss. However, based on the experimental results, it is observed that these objective functions have a negligible impact on smaller-sized perturbations. Hence, we have chosen not to consider them.

B. Framework of SAR Target Perturbation Generation Based on SAR Sticker

Considering the actual resolution of SAR images, one pixel often corresponds to an area of 0.3 × 0.3 m (significantly larger than the commonly used optical datasets). It is essential to compress the size of adversarial perturbations while ensuring effective attack performance. Therefore, our initial focus is on generating perturbations on the surfaces of target in SAR images.

Specifically, we begin by identifying the most advantageous placement position for the SAR sticker on the input image. The gradient-weighted class activation mapping (Grad-CAM)

Algorithm 1: Method of Target Perturbation Generation.

```
Input:
    SAR image $X$ and its label $t_0$;
    Initial sticker perturbation $\delta_0$;
    Number of iterations $N$.
Output:
    SAR adversarial image $X'$;
    1st-stage: Sticker position Initialization
1: Extracting the target region of the input SAR image $X$
   using Grad-CAM method;
2: Calculating the region with the highest weight in the
   CAM, then $X_1 = X + \delta_0$;
    2nd-stage: Sticker pixel optimization
3: Input $X_1$ into the SAR ATR model and calculate the
   score of $Z(X_1)$;
4: If $Z(X_1) < 0$ or current iteration count $> N$, proceed to
   step 8; otherwise, proceed to execute step 5;
5: Backpropagation is used to obtain the gradient in $\delta_0$
   region, which is then used to fine-tune that particular
   area pixel, resulting $\delta'$;
6: $X' = X + \delta'$;
7: $X'$ will serve as the new input for step 3;
8: return $X'$.
```

method, as introduced by [29], is employed to extract the region of interest for the SAR ATR model. Once the position for SAR sticker is determined, a robust objective function (1) is employed to ensure attack efficiency. Through continuous iterative optimization of the sticker's pixels, this process continues until the generated AEs successfully fool the SAR ATR model. Fig. 2 provides an overview of this module.

1) Grad-CAM technique is a crucial tool for exploring the interpretability of DNNs. This method effectively visualizes the correlation between the features extracted from DNNs and the original image, providing a comprehensive understanding of how the model produces predictions. This aligns perfectly with our intention of finding the optimal placement position for SAR sticker. It functions by multiplying feature maps, acquired during forward propagation, with gradient information obtained during backpropagation. This multiplication results in a weighted feature map that serves as the basis for visualizing the model's classification decisions. In this module, Grad-CAM operates as a plugin, producing an attention map for the SAR image from the model. The core formula for Grad-CAM is as follows:

$$L^{t}_{\text{Grad-CAM}} = ReLU\left(\sum_{k} \alpha_k^t A^k\right). \tag{2}$$

The left-hand side of the formula represents the CAM of the model for class t. A^k represents the feature maps of the selected convolutional layer, and α_k^t is the weight obtained by the model through backpropagation. The calculation formula for α_k^t is as follows:

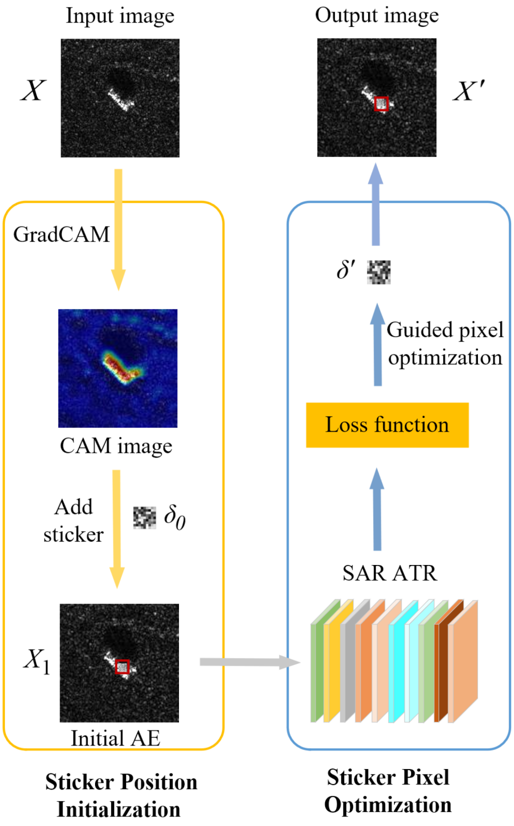


Fig. 2. Illustration of target perturbation generation module based on SAR sticker.

$$\alpha_k^t = \frac{1}{S}\sum_i \sum_j \frac{\partial y^t}{\partial A_{ij}^k} \tag{3}$$

where i and j denote the width and height of the feature map A^k. The term $\frac{1}{S}\sum_i \sum_j$ represents the result of a global average pooling operation on the calculated gradients, while S corresponds to the product of i and j. The remaining section of the equation specifies the backpropagation of the loss for class t through the feature map.

2) After extracting the CAM of input image, we perform convolution using a kernel of the same size as the SAR sticker, resulting in X_1 as shown in Fig. 3. Subsequently, we move on to the sticker pixel optimization. The newly generated AE X' is continuously fed as input to SAR ATR models until the objective function (1) is met. The detailed process of this module is shown in Algorithm 1.

In addition, to meet the pixel constraint of AEs, we utilize substitution variable and introduce a new variable ζ to replace the perturbation variable δ. This mapping relationship ensures that the pixel range of δ remains between 0 and 1 throughout the optimization process. This can be expressed as

$$\delta = \frac{1}{2}(\tanh(\zeta) + 1). \tag{4}$$

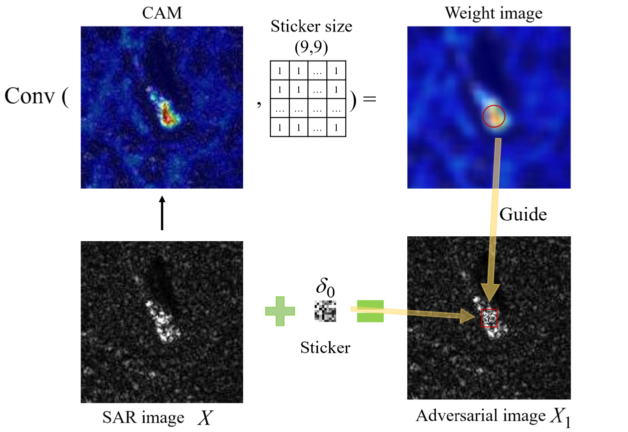


Fig. 3. Illustration of how CAM guides the optimal placement of the sticker.

C. Framework of SAR Background Perturbation Generation Based on Strong Scatterer

In the previous discussion, we considered the characteristics of SAR images and proposed placing small-scale perturbations in the target surface. It is worth noting that SAR images often exhibit background regions filled with speckle noise. Considering the complexity of real-world environments, we contemplate generating perturbations in the background region as well to further enhance the attack performance. We opt to use some common strong scatterers in the SAR calibration field as perturbaters using their SAR images [30].

Fig. 4 provides an overview of our SAR background perturbation generation module. This process involves three main steps. First, using the simple scattering model, we simulate strong scatterer images with controlled size and intensity based on the relevant scattering coefficients of the input SAR image. Second, by employing the PSO algorithm, we find the optimal positions for placing these scatterer images on the SAR image, aiming for the most effective attack. Finally, we halt the iteration upon reaching the maximum count or when the desired performance criteria are satisfied, and output the generated AEs.

1) The purpose of this module is to generate images of strong scatterer, which exhibit high scattering capabilities in SAR images, appearing as bright points. With a focus on physical realization, the images of strong scatterer we generate must closely resemble actual SAR images. Therefore, we opt for the simple scattering model for simulation. First, we need to calculate its backscatter coefficient, and the formula is provided by [31]

$$\sigma = \frac{4\pi a^4}{3\lambda^2} \tag{5}$$

where σ represents the scattering coefficient of the strong scatterer, while a represents its orthogonal side length. In addition, λ refers to the wavelength of the radar wave.

After computing the backscatter coefficient, we simulate the echo signals of strong scatterers using relevant parameters derived from the input image [32] to ensure authenticity. Subsequently, we apply the classic Range-Doppler algorithm (RDA) in SAR imaging to process these echo signals. The RDA, a well-known radar imaging technique, captures variations in range and Doppler dimensions of the echo signals to produce an image. By inputting the previously acquired echo signals into the RDA, the image representing strong scatterer is obtained, as shown in the upper part of Fig. 4.

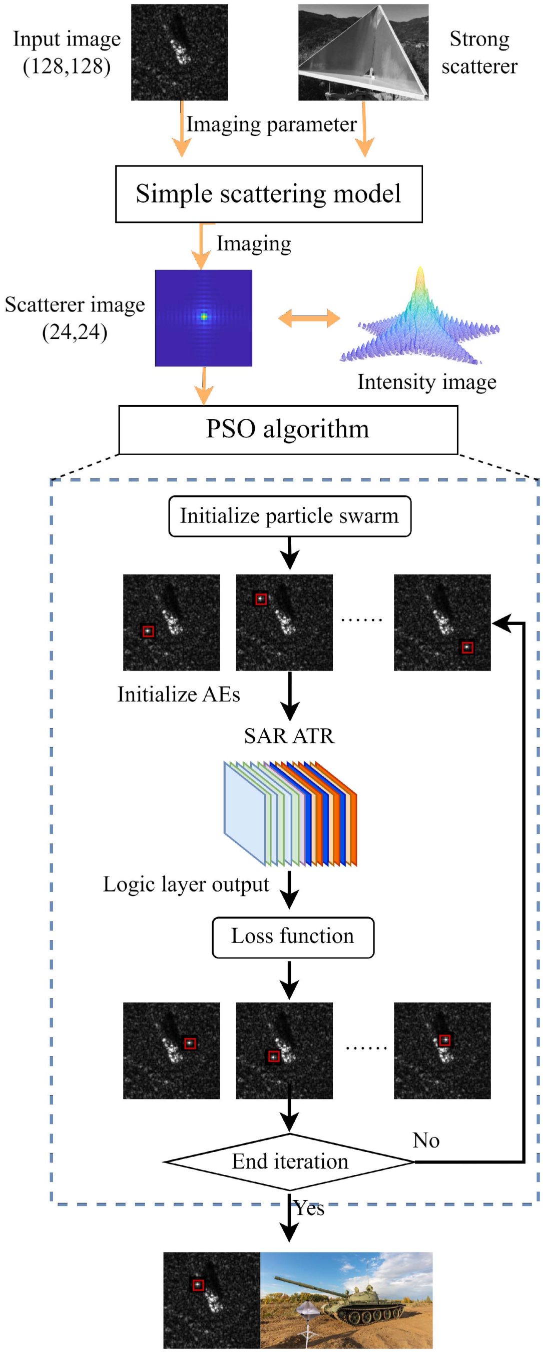


Fig. 4. Illustration of background perturbation generation module based on strong scatterer.

2) The PSO algorithm, inspired by swarm intelligence and the foraging behavior of bird flocks, is an optimization technique. It simulates the movement of individual particles within the search space to find the optimal solution

Algorithm 2: Method of Background Perturbation Generation.

```
Input:
   SAR image $X$ and its label $t_0$;
   Parameters for PSO initialization:
   $u = 0.5, c1 = 1.5, c2 = 1.5$; Number of iterations $N$.
Output:
   SAR adversarial image $X'$;
1: Retrieving the imaging parameters of the input SAR image $X$;
2: Calculating the scattering coefficient $\sigma$ of the strong scatterer;
3: Computing the scatterer image using the SAR-RDA;
4: Initializing the PSO algorithm to generate a set of potential adversarial images $X_1$.
5: Input $X_1$ into the SAR ATR model and calculate the score of $Z(X_1)$;
6: If $Z(X_1) < 0$, proceed to step 9; otherwise, proceed to execute step 7;
7: Updating the particle parameters using functions (6) and (7), resulting $X'$;
8: $X'$ will serve as the new input for step 5;
9: return $X'$.
```

to a problem, without relying on gradient information. This optimization approach offers three main advantages for our purposes. First, since we have already obtained specific perturbation images, our subsequent efforts focus on strategically placing them on the input image. Second, as mentioned in the introduction, adding subtle perturbations to an image can disrupt the model's prediction. Considering the effectiveness of our attack, we cannot modify other pixels. Finally, PSO algorithm is capable of generating a batch of AEs with good attack performance, making it a straightforward and powerful tool for our application.

In PSO algorithm, a population of particles is maintained, where each particle represents a candidate solution for function (1) and possesses its own velocity within the solution space. In this attack module, the search space encompasses all coordinate positions within the input SAR image, and each particle carries a set of coordinates for the placement of strong scatterer images. In each iteration, particles continuously explore the solution space by updating their velocities and positions using their current position and velocity, as well as the best historical position of the individual particle and the overall best position of the particle swarm. The equations for updating particle velocity and position are presented below. Velocity update formula

$$\begin{aligned} v_k(n+1) = u \cdot v_k(n) + c_1 \cdot r_1 \cdot (pbest_k - l_k(n)) \\ + c_2 \cdot r_2 \cdot (gbest - l_k(n)). \end{aligned} \quad (6)$$

Position update formula

$$l_k(n+1) = l_k(n) + v_k(n+1). \quad (7)$$

TABLE I
CONFIGURATION OF EXPERIMENT PLATFORM

Programming language	Python3.8
Deep learning API	PyTorch 1.8.0
CPU	Intel Core i7-10700k
Memory	32GB
GPU	GeForce GTX 1080Ti

The current iteration number is denoted by n, while $l_k(n)$ and $v_k(n)$ represent the current position and velocity of the particle k, respectively. The best individual position of the particle is represented by $pbest_k$, and the best position within the group is represented by $gbest$. The inertia parameter is denoted by u, while the learning factors are represented by $c1$ and $c2$. In addition, $r1$ and $r2$ refer to random numbers that are uniformly distributed between 0 and 1.

Finally, we evaluate the quality of each candidate solution using a cost function, which corresponds to the score value of function (1). If the score is less than 0 or reaches the maximum number of iterations, we terminate the algorithm. The detailed process of this module is shown in Algorithm 2.

III. Experimental Results and Analysis

This main part of this section assesses the performance of the proposed method across various experimental scenarios, which are divided into three parts. First, the individual attack outcomes of the two introduced modules are independently analyzed and presented. Subsequently, the combined attack results from their collaboration are provided. The performance will be evaluated based on two key indicators: 1) the fooling rate; and 2) transferability. The SAR ATR models used in the experiments are implemented using the PyTorch backend. The basic configuration of the platform is detailed in Table I.

A. Experimental Data and Evaluation Metrics

1) MSTAR Dataset [32]: We utilized the widely-used MSTAR dataset in our experiments. Specifically, we selected the MSTAR data with a depression angle of 17° as the training set and the data with a depression angle of 15° as the test set. The training set comprised 2747 images, while the test set consisted of 2425 images. This choice ensured that both the training and test sets were derived from distinct datasets, mitigating the risk of overfitting. Fig. 5 depicts the correspondence between optical and SAR images of ten categories. Table II provides a detailed distribution overview of the MSTAR dataset.

2) Well-Trained CNN Model: To evaluate the attack performance, we selected the lightweight network SAR-CNN, along with VGG19, ResNet50, DenseNet121, and MobileNetV2 as the target models. SAR-CNN consists of four overlapping convolutional and pooling layers followed by three fully connected layers [25]. The remaining four models have been proven to perform well in optical domain, and their specific structures are not elaborated here.

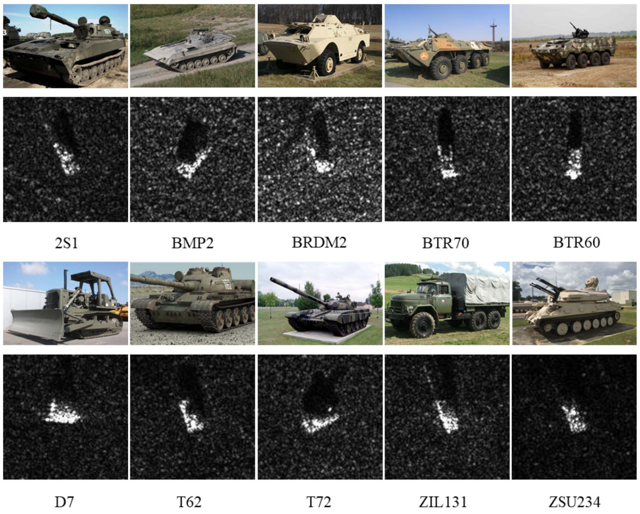


Fig. 5. Examples of ten types of ground vehicle targets in the MSTAR dataset.

TABLE II
MSTAR Dataset Details

Index	Name	Train data	Test data
0	2S1	299	274
1	BRDM2	298	274
2	BTR60	256	195
3	D7	299	274
4	T62	299	273
5	ZIL131	299	274
6	ZSU234	299	274
7	BMP2	233	195
8	BTR70	233	296
9	T72	232	196
Total	–	2747	2425

TABLE III
Training and Testing Accuracy of Five SAR ATR Models

Model	Training	Testing
SAR-CNN	100%	96.37%
VGG19	99.64%	94.19%
DenseNet121	100%	97.4%
ResNet50	100%	97.11%
MobileNetV2	100%	97.16%

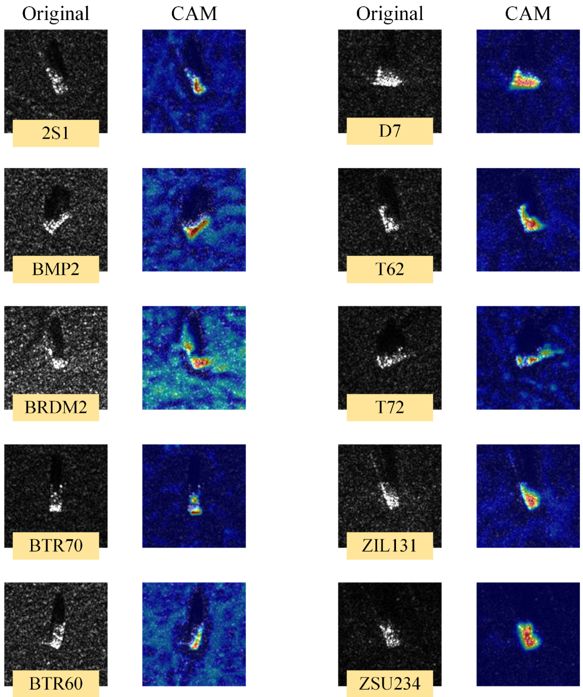


Fig. 6. Comparison of CAM and original images in MSTAR.

The experiments were conducted with an input image size of 128 × 128. The training process utilized the SGD optimizer with the following hyperparameters: learning rate of 0.01, decay of 1e-6, momentum of 0.9, epoch of 30, and batch size of 32. The training results are presented in Table III.

3) Evaluation Metrics: The fooling rate is introduced as a measure to validate the efficacy of the proposed method. It quantifies the relationship between the number of successful attacks, denoted as N_{success}, and the total number of AEs, denoted as N_{all}. The fooling rate can be mathematically expressed as

$$\text{fooling rate} = \frac{N_{\text{success}}}{N_{\text{all}}}. \tag{8}$$

Second, Transferability assesses the fooling rate of AEs generated on one model when applied to other models. This parameter is pivotal in both adversarial attacks and defenses, as a high transferability signifies the AEs' ability to consistently exert influence across different models. In the subsequent experiments, we will input the AEs, generated for each SAR ATR model, into the other four ATR models.We will record their fooling rates and conduct an analysis of the results.

B. SAR Adversarial Attack Using Only the Target Perturbation Generation Module

This subsection focuses on conducting attack experiments aimed at perturbing the target region using SAR sticker. We begin by introducing the process of extracting the target region through Grad-CAM and showcasing the resulting visual effects. Following that, we provide a thorough evaluation and discussion of the generated set of AEs within this specific scenario.

Here, we showcase the identification of attention regions using Grad-CAM. Among the ten categories, CAMs for seven effectively aligned with the vehicle targets. Only three categories displayed CAMs that partially covered the target regions and shadow regions. For visual representation of the experiment, one image from each category was randomly selected, and its corresponding original image and CAM were presented. The experimental findings are depicted in Fig. 6.

The SAR-CNN model was employed to generate AEs for each image in the MSTAR test dataset. A subset of images

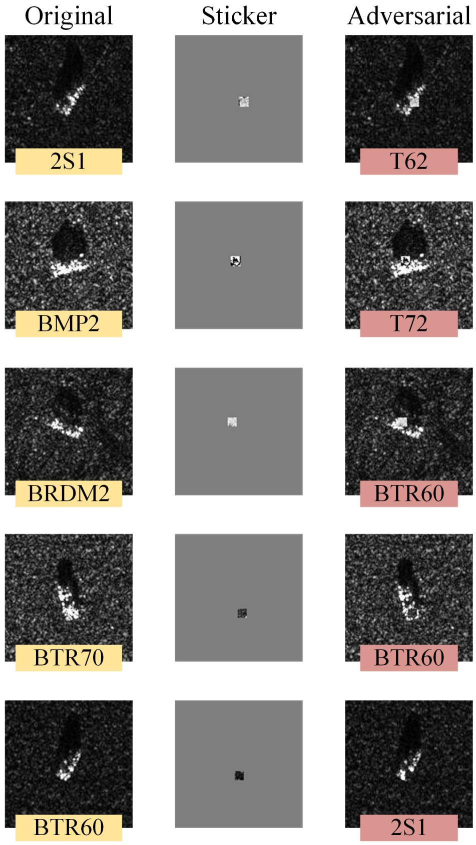


Fig. 7. SAR AEs exclusively utilized target perturbation generation module.

misclassified by the SAR-CNN model was excluded. We defined the size of the SAR sticker as 9 × 9, covering around 0.5% of the total image pixels, with 20% of the pixels in the target region. A total of 1740 AEs were generated, with an average of 660 iterations per image. Consequently, the overall fooling rate reached approximately 76%. To visually demonstrate the effectiveness and stealthiness of SAR sticker, Fig. 7 displays randomly selected images from five categories. Furthermore, experiments were conducted on four other SAR ATR models, resulting in fooling rates of 34.3% for VGG19, 69.6% for ResNet50, 34.9% for DenseNet121, and 33% for MobileNetV2. Fig. 8 illustrates the transferability performance of this module, with an average fooling rate of approximately 10%. These fooling rates were achieved by attacking the black-box target model using AEs generated by the surrogate model.

C. SAR Adversarial Attack Using Only the Background Perturbation Generation Module

In this subsection, we will discuss the results obtained from background perturbation generation module on varying scatterer quantities, scatterer intensities, and transferability between targeted models. Our findings highlight a significant decrease in accuracy observed in widely-used SAR ATR models when subjected to this module. Moreover, we note an enhancement in attack performance corresponding to the increment in scatterers. Modifying the size and intensity of the scatterer further amplifies the fooling rate. Finally, our transferability experiments validate the effectiveness of using AEs generated by surrogate models to attack black-box models. These results underscore the practicality and effectiveness of introducing highly scatterers into the background environment to achieve the intended attack performance.

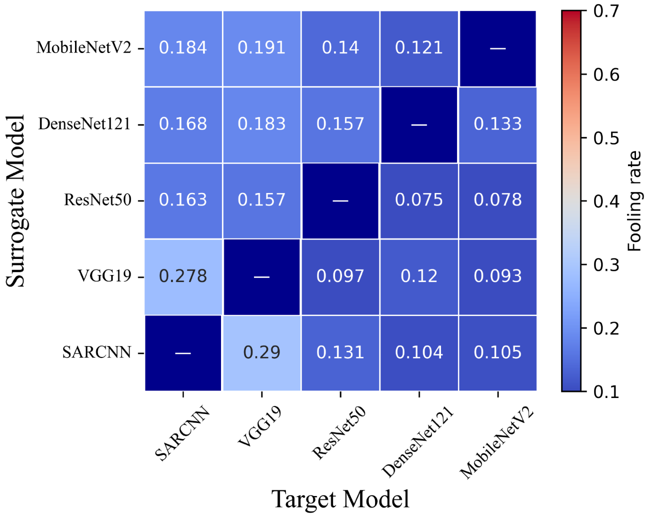


Fig. 8. Transferability of the generated AEs by target perturbation generation module. The surrogate model refers to the one used to craft AEs, and the target model denotes the unknown target victim classifier.

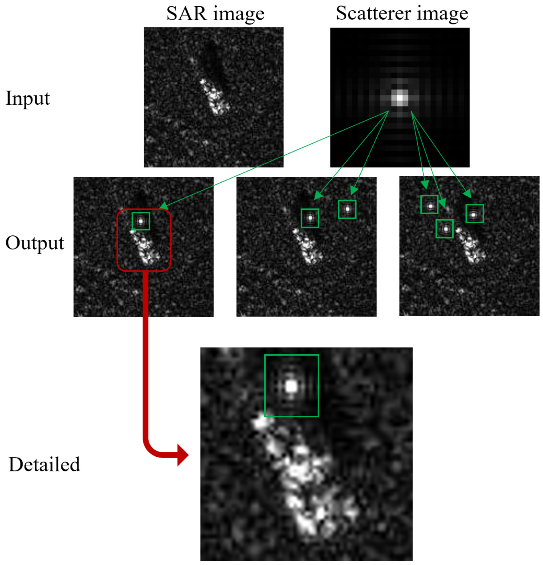


Fig. 9. SAR AEs generated by three different quantities of strong scatterer.

1) *Results on Different Quantities of Scatterer:* The performance on the five evaluated models, considering varying quantities of scatterers, is presented in Table IV. The key findings are as follows: First, this module significantly fools the five SAR ATR models. Even with just one strong scatterer measuring 24 × 24 (accounting for only 0.4% of the input image), the fooling rate can reach an average of 45%. Second, as the number of scatterers increases, the attack performance further improves. Fig. 9 displays the AEs corresponding to the three different quantities of scatterers reported in Table IV. Consequently,

TABLE IV
FOOLING RATE OF FOUR DIFFERENT QUANTITIES OF STRONG SCATTERER ON FIVE SAR ATR MODELS

Scatterer number	SAR-CNN	VGG19	ResNet50	DenseNet121	MobileNetV2	Average
One-scatterer	0.727	0.349	0.416	0.390	0.371	0.451
Two-scatterer	0.891	0.606	0.673	0.650	0.586	0.681
Three-scatterer	0.938	0.762	0.810	0.782	0.716	0.802
Five-scatterer	0.948	0.831	0.870	0.844	0.778	0.854

TABLE V
FOOLING RATE OF THREE INTENSITY LEVELS OF SCATTERERS ON FIVE SAR ATR MODELS

SAR ATR Model	24 × 24	32 × 32	48 × 48
SAR-CNN	0.727	0.888	0.946
VGG-19	0.349	0.592	0.826
ResNet50	0.416	0.673	0.884
DenseNet121	0.390	0.544	0.772
MobileNetV2	0.371	0.508	0.697
Average	0.451	0.641	0.825

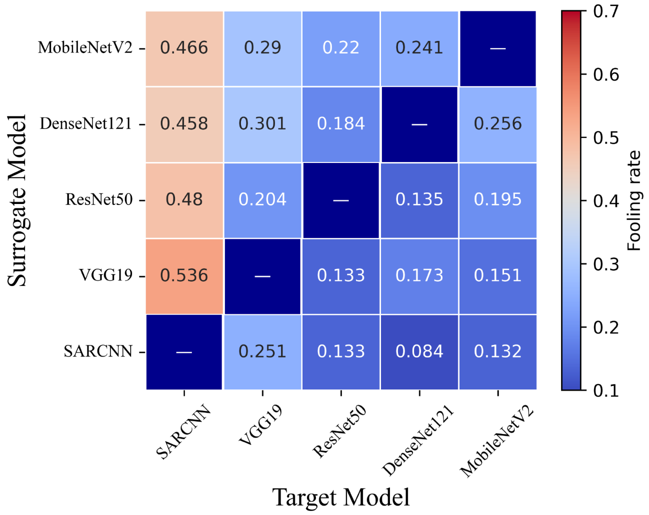


Fig. 10. Transferability of the generated AEs by background perturbation generation module.

incorporating more reflectors in practical scenarios can yield a higher fooling rate.

2) Results on Different Intensity of Scatterer (Presented as Size): In Table V, the impact of three intensity levels of scatterers on five evaluation models is reported. We tested the attack performance at three sizes: 24 × 24 (0.4% of the input image), 32 × 32 (1%), and 48 × 48 (2%). For the same model, as we continuously increased the size of scatterer, the fooling rate significantly increased. Overall, starting from an initial fooling rate of 45.1%, continuously enhancing the scattering intensity resulted in an average fooling rate increase of 20%.

3) Results on Transferability: Fig. 10 clearly illustrates the transferability performance on various target models. It is evident that the fooling rates of SAR-CNN and VGG19 are approximately 46% and 25%, respectively. Meanwhile, the AEs generated on MobileNetV2 resulted in an average fooling rate of around 30% for the other four models. These results indicate that, in practical scenarios, introducing some strongly scatterers into the background environment has the potential to achieve effective camouflage for the target.

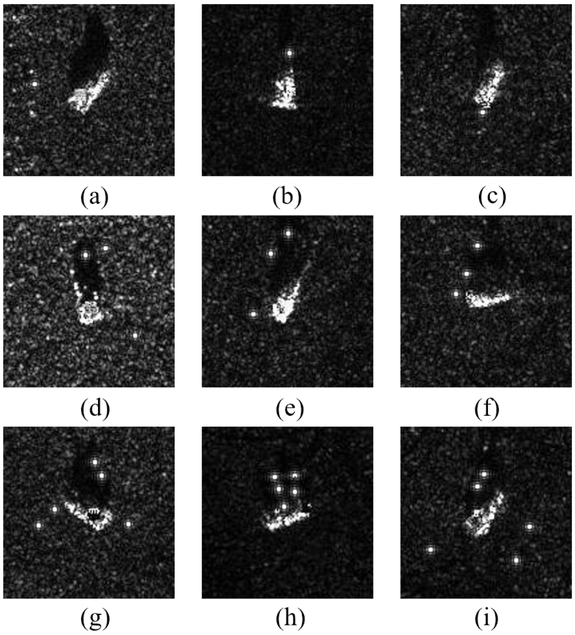
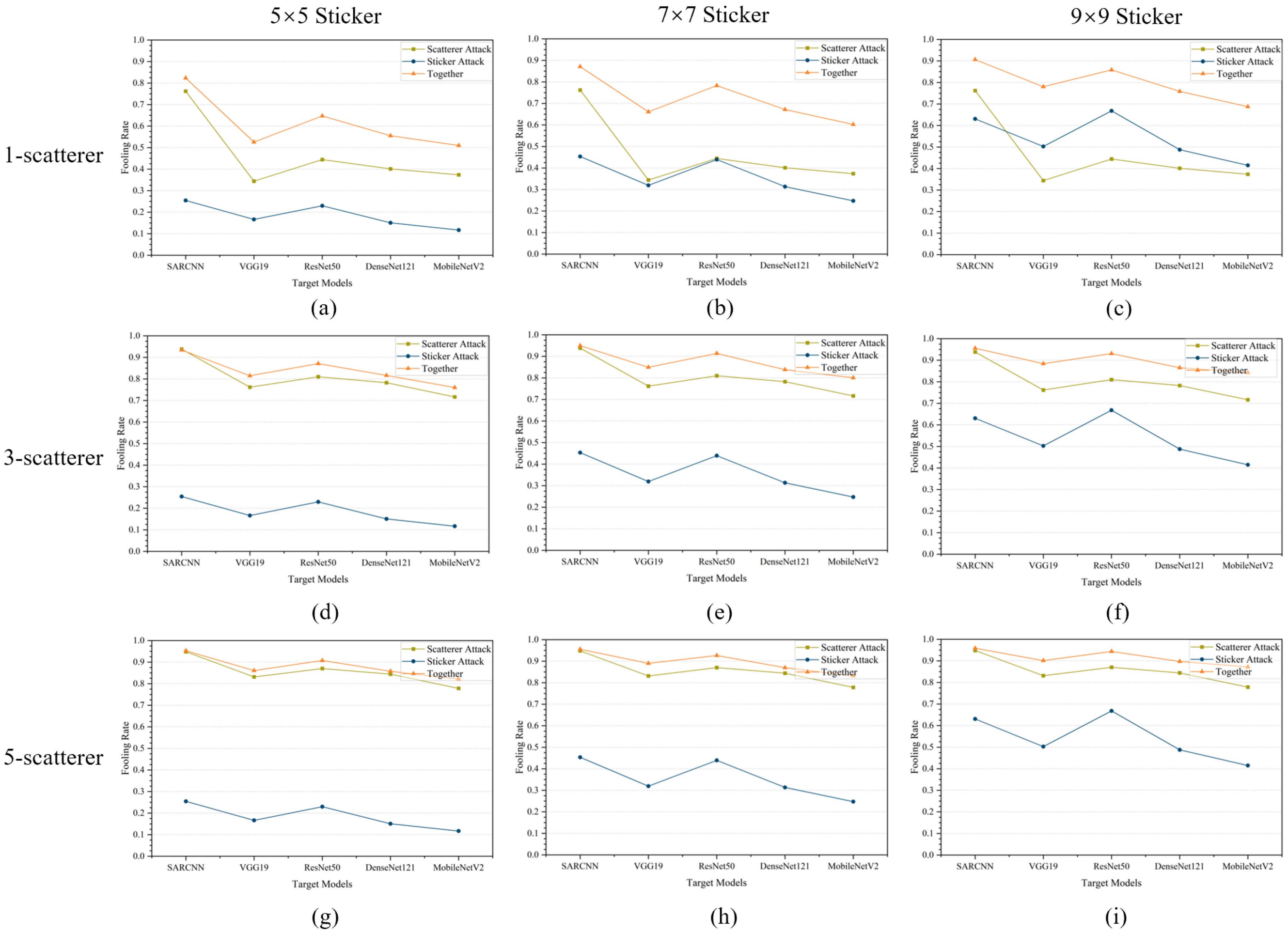


Fig. 11. SAR AEs generated by combined attack. (a) Combined one-scatterer and size 9 sticker. (b) Combined one-scatterer and size 7 sticker. (c) Combined one-scatterer and size 5 sticker. (d) Combined three-scatterer and size 9 sticker. (e) Combined three-scatterer and size 7 sticker. (f) Combined three-scatterer and size 5 sticker. (g) Combined five-scatterer and size 9 sticker. (h) Combined five-scatterer and size 7 sticker. (i) Combined five-scatterer and size 5 sticker.

D. SAR Adversarial Attack Using Both Aforementioned Modules

In this subsection, we conduct simultaneous SAR sticker on the target surface and strong scatterer on the target's surroundings. As illustrated in Fig. 11, we experimented with nine different combinations of attack strategies, aiming for a comprehensive exploration of outcomes. During these experiments, we assess the fooling rate for the SAR sticker,strong scatterer, and the combined attack individually. Finally, we visually demonstrate the effectiveness of this method by transferability experiment and providing accompanying data analysis.

1) Analysis of the Fooling Rate in Five SAR ATR Models: We conducted a series of tests involving three quantities of strong

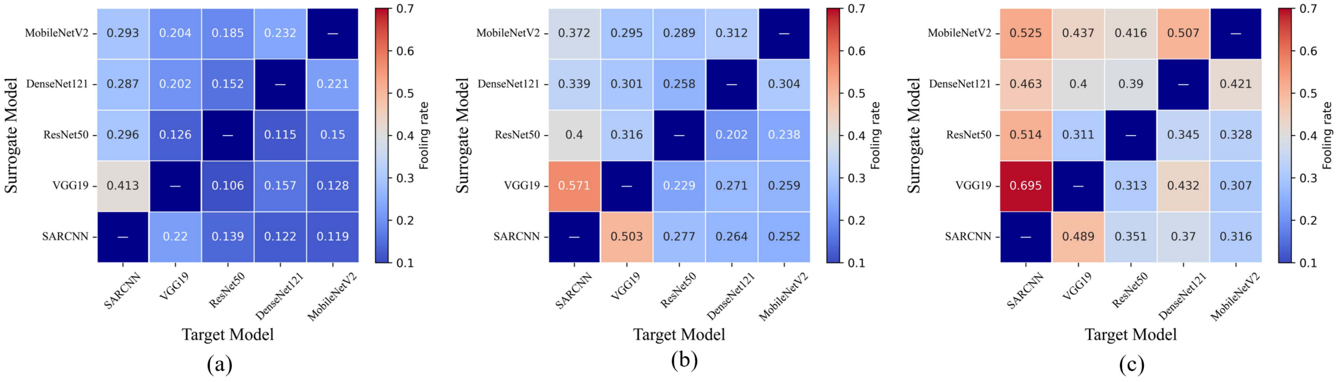


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Fig. 12. Fooling rates of proposed method with different sizes of stickers and different number of scatterers. (a) 5 × 5 sticker and one scatterer. (b) 7 × 7 sticker and one scatterer. (c) 9 × 9 sticker and one scatterer. (d) 5 × 5 sticker and three scatterers. (e) 7 × 7 sticker and three scatterers. (f) 9 × 9 sticker and three scatterers. (g) 5 × 5 sticker and five scatterers. (h) 7 × 7 sticker and five scatterers. (i) 9 × 9 sticker and three scatterers.



(a) (b) (c)

Fig. 13. Transferabiltiy of the generated AEs by combined attack. (a) Combined one-scatterer and size 5 sticker. (b) Combined one-scatterer and size 9 sticker. (c) Combined five-scatterer and size 9 sticker.

scatterers (one-scatterer, three-scatterer, and five-scatterer) and SAR sticker of different sizes (size 5, size 7, and size 9). In total, we examined the performance of nine attack combinations. Fig. 12 provides a detailed representation of the individual effects of the two modules in nine different scenarios, as well as the performance of their combined attack. For instance, Fig. 12(f) illustrates the attack performance of the three-scatterer (represented by the brown line), the SAR sticker of size 9 (represented by the blue line), and their combined attack performance (represented by the orange line).

Overall, when employing the SAR sticker alone (with a focus on changes in the blue line), the fooling rate increases by

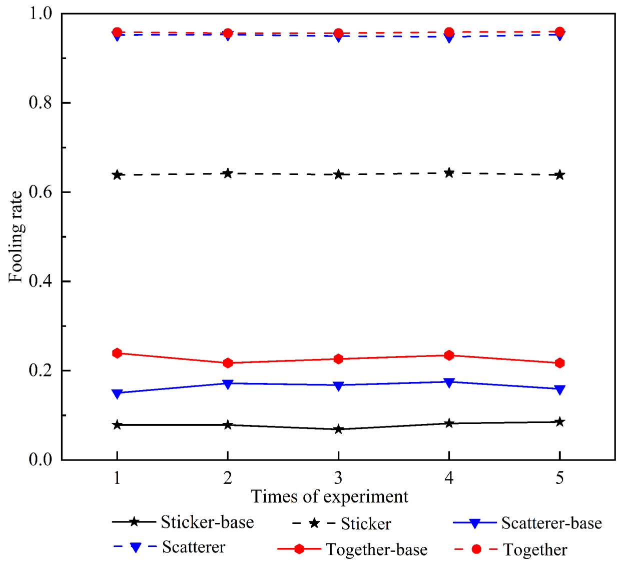


Fig. 14. Comparative analysis of baseline and proposed method performance.

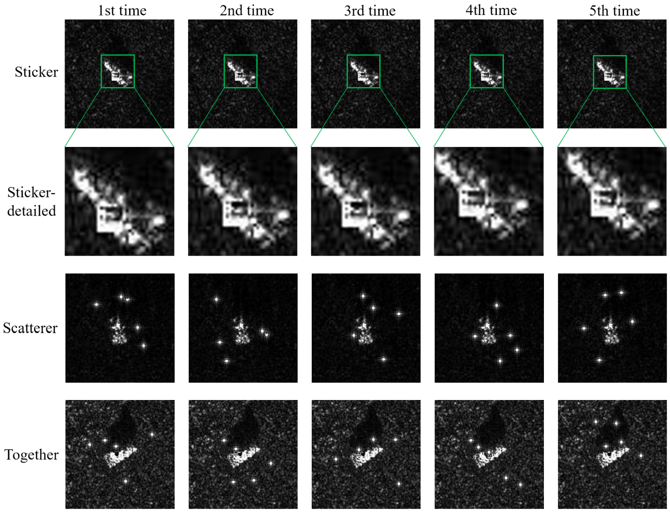


Fig. 15. Display of SAR AEs from five repetitive experiments.

approximately 20% for each incremental increase in sticker size. Conversely, when using only strong scatterers (with a focus on changes in the brown line), the fooling rate remarkably improves with an increasing number of scatterers, reaching saturation at five scatterers. Even under the most stringent conditions of combined attacks, such as the combination of a one-scatterer and a size 5 sticker shown in Fig. 12(a), the fooling rate of combined attacks can still be maintained at 60%. By further relaxing the size restriction on stickers, as illustrated in Fig. 12(c), the fooling rate approaches 80%. Lastly, as depicted in Fig. 12(i), increasing the number of scatterers in a combined attack yields a fooling rate as high as 90%. This signifies that by altering the pixel values of 20% of the target vehicle's surface and positioning five strong scatterers around the vehicle, we can effectively camouflage our target.

2) Analysis of the Fooling Rate in Transferability: Fig. 13(a) presents the combination of a one-scatterer and a size 5 sticker. In this scenario, the fooling rate remains relatively low, usually ranging from approximately 10% to 20%. Fig. 13(b) illustrates the combination of a one-scatterer and a size 9 sticker. In this setup, the fooling rate is generally approximately 10% higher compared to the previous case. Lastly, Fig. 13(c) showcases the combination of a five-scatterer and a size 9 sticker. Notably, this configuration yields promising results, indicating that the generated AEs possess universal attack performance.

IV. Discussion

In the forthcoming discussion, we embark on a meticulous exploration of our experimental results, shedding light on the robustness and efficacy of our proposed methods. We commence by establishing a baseline performance metric, providing a benchmark for the methods discussed above. This is followed by an indepth analysis of the stability of our methods, demonstrating their resilience under varying conditions. It is expected that in operational scenarios, not just one algorithm is run over a SAR image, but multiple ATR algorithms with different strategies are employed. Therefore, we shift our focus to the performance on non-DNN models and across different datasets. Finally, we draw a comparison between our proposed method and state-of-the-art methods. This comprehensive discussion aims to provide a holistic understanding of our work.

A. Baseline Performance and Method Stability

In this subsection, we undertake a comprehensive evaluation of our baseline experiments and rigorously test the stability of our proposed methods. Initially, we establish a baseline performance metric using a randomized generation method for both position (for the SAR sticker and strong scatterer previously discussed) and appearance (for the SAR sticker). This provides a reference point for assessing the effectiveness of methods in optimizing both position and appearance. Subsequently, we conduct stringent stability tests on our methods, simulating real-world scenarios where the setup of strong scatterers may not be ideal. This ensures the robustness of our attack strategies under complex conditions.

Fig. 14 presents a comparative analysis of the fooling rates achieved by the baseline experiments and our proposed methods, with each experiment being repeated five times. The solid lines represent the baseline experiments. For the SAR sticker, we used randomly initialized appearances and positions; for the strong scatterer, we randomly placed five scatterers on the image in each instance. These baseline methods achieved relatively low fooling rates. In contrast, the dashed lines of the same color represent the results of our methods, which incorporated optimizations for the appearance and position. In five repetitions of the experiment, our methods consistently achieved high fooling rates. Taking the SAR sticker represented in Fig. 14 as an example, the black solid line represents the baseline, which used randomly placed and appearing stickers, achieving a fooling rate near 0.1. However, observing the black dashed line, which represents our method with optimized appearance and position, the fooling rate consistently remained above 0.6. Fig. 15 presents the results of the same image subjected to the proposed method across five repeated experiments. Upon horizontal comparison, each

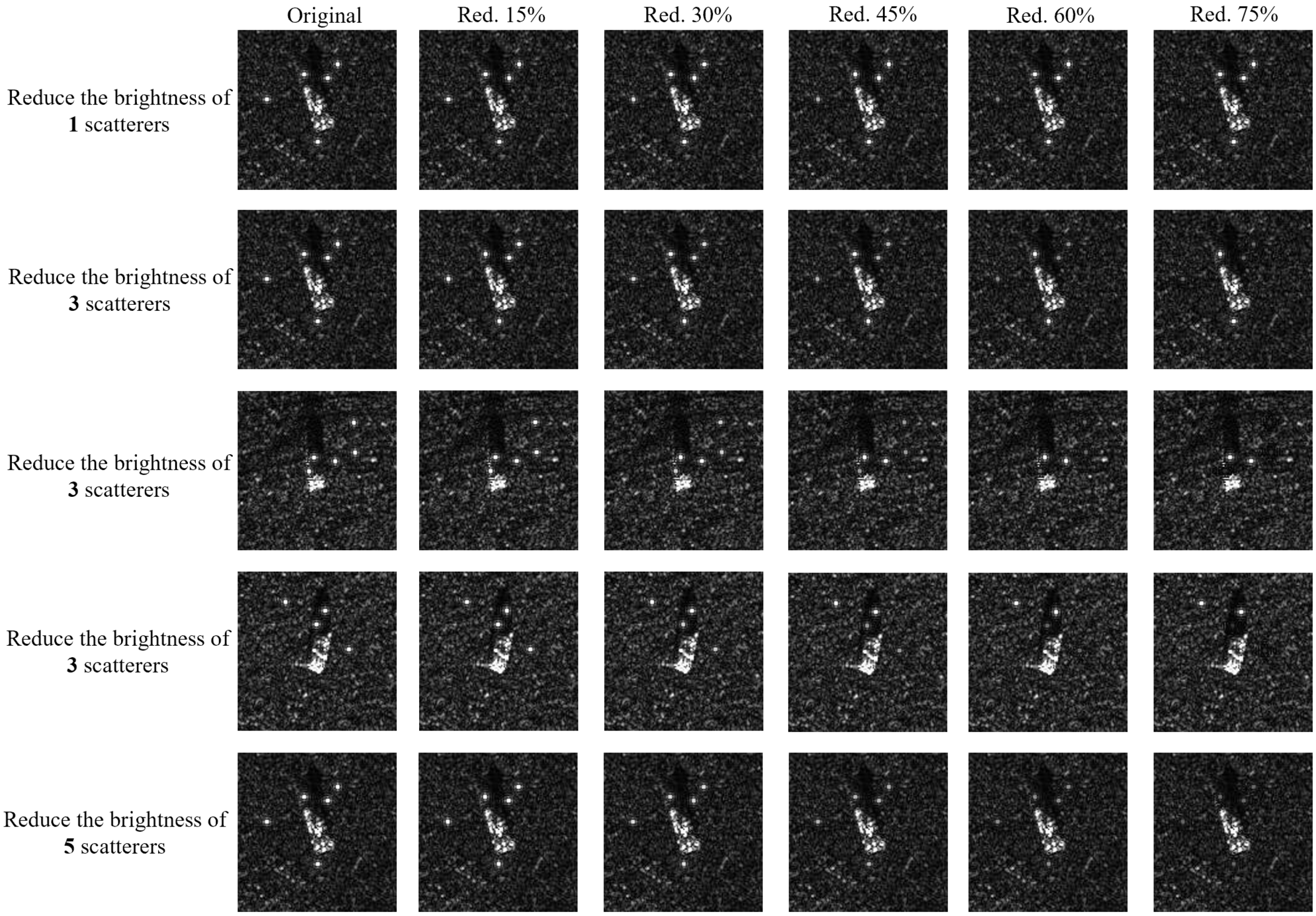

Original
Red. 15%
Red. 30%
Red. 45%
Red. 60%
Red. 75%
Reduce the brightness of **1** scatterers
Reduce the brightness of **3** scatterers
Reduce the brightness of **3** scatterers
Reduce the brightness of **3** scatterers
Reduce the brightness of **5** scatterers

Fig. 16. Visual demonstration of each stage in the scatterer brightness reduction experiment.

TABLE VI
EFFECT OF SCATTERER BRIGHTNESS REDUCTION ON THE CONFIDENCE OF MAXIMUM MISCLASSIFICATION IN A FIVE-SCATTERER SCENARIO

Number of scatterers	Brightness reduction				
with reduced brightness	15%	30%	45%	60%	75%
1	0.9999	0.9998	0.9997	0.9996	0.9992
3	0.9999	0.9993	0.9965	0.9679	0.8843
3	0.9991	0.9999	0.9999	0.9995	0.9856
3	1.0	0.9999	0.9951	0.6756	0.9998
5	0.9998	0.9978	0.9314	**0.1903**	**0.0019**

TABLE VII
VARIATIONS IN MODEL RECOGNITION ACCURACY WHEN APPLYING OUR METHOD TO SAMPLE AND SARSIM

Dataset	Training	Testing	Sticker	Scatterer	Togeter
SAMPLE [32]	1.0	0.922	0.163	0.016	0.001
SARsim [33]	1.0	0.912	0.101	0	0

generated AE exhibits differences, yet they consistently achieve effective target camouflage. These results demonstrate a significant increase in the fooling rates achieved by our methods compared to the baseline.

In real-world scenarios, the setup of corner reflectors may not always be optimal. We tested the impact of this by simulating a five-scatterer scenario where multiple reflectors were incorrectly positioned, resulting in a reduction in scattering brightness. The highest confidence level in the nonoriginal categories, given by the SAR ATR model after the attack, will be recorded. Data from Table VI shows that even with a stepwise decrease in brightness of 15% for one or more scatterers, the attack performance largely remains stable. It is only when all five scatterers significantly lose brightness that the camouflage effect notably decreases (see the bolded cases of 60% and 75% brightness reduction in Table VI). This indicates that even if one or more corner reflectors are not ideally placed, the overall camouflage effect is not greatly

TABLE VIII
FOOLING RATES OF SAR AES GENERATED BY DIFFERENT METHODS ON MSTAR DATASET

Method	SAR-CNN	VGG19	ResNet50	DenseNet121	MobileNetV2	Average
Adversarial Patch [18]	0.083	0.063	0.073	0.059	0.036	0.063
Pixel Attack [34]	0.252	0.584	0.153	0.499	0.415	0.380
Auto PGD [35]	0.704	0.826	0.932	0.930	0.958	0.870
Auto Attack [35]	0.723	0.830	0.941	**0.934**	**0.958**	0.877
Ours	**0.959**	**0.901**	**0.944**	0.897	0.872	**0.915**

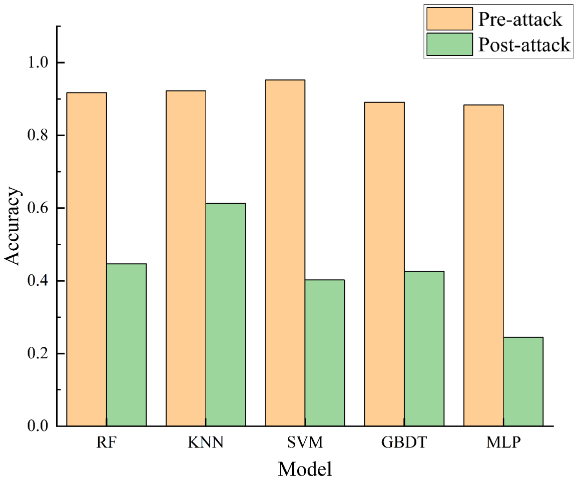


Fig. 17. Performance degradation of classical machine learning models under our attack methods.

impacted. To provide a more vivid illustration, Fig. 16 displays the results at various stages of the scatterer brightness reduction experiment, corresponding to Table VI.

B. Non-DNN Models and Cross-Dataset Performance

In this work, we undertake a comprehensive investigation into the performance of our method on non-DNN models and across various datasets. While adversarial attacks are primarily employed to explore security concerns associated with diverse DNNs, we are equally interested in exploring the efficacy of our method when applied to non-DNN models. In addition, we also validate its performance on other SAR datasets.

Fig. 17 illustrates the performance of five classical machine learning models, namely random forest, k-nearest neighbors, support vector machine, gradient boosting decision tree, and multilayer perceptron, before and after the implementation of our attack method. The orange bar chart represents the preattack accuracy of various models on the MSTAR dataset, all hovering around 0.9. The green bar chart illustrates a significant drop in the accuracy of each model to approximately 0.4 postattack. This striking contrast underscores the potency of our methods, even when applied to non-DNN models. It suggests that while DNNs are often the primary target of adversarial attacks, non-DNN models are not immune and can also be significantly impacted.

Table VII provides a comprehensive depiction of the performance of our methods on two supplementary datasets, namely SAMPLE [32] and SARsim [33]. The second and third columns present the training and testing accuracies, respectively, of the SAR-CNN model on these datasets. The ensuing three columns illustrate the accuracy following the execution of our attack method. The results indicate that even with the application of the SAR sticker alone, the accuracy significantly deteriorates to below 0.2 on both datasets. Upon further implementation of the strong scatterer and their combined assault, the SAR-CNN model loses its ability to recognize effectively.

C. Performance Comparison

A comparative experiment with state-of-the-art methods is designed to evaluate the performance of the proposed method. The Adversarial Patch [18] and Pixle Attack [34], featured in Table VIII, are classic physical attack methods designed to introduce visually noticeable block-wise perturbations into images. In recent years, Auto PGD [35] and Auto Attack [35] have emerged as adversarial attack methods posing significant threats to DNNs. Overall, our method demonstrates the best average attack performance, with the highest fooling rates observed in SAR-CNN, VGG19, and ResNet50. DenseNet121 and MobileNetV2 lag behind by approximately 4 and 8 percentage points, respectively. It is worth emphasizing that our method places a greater emphasis on physical realizability compared to Auto PGD and Auto Attack. The adversarial attack performance of our method is significantly superior than the physically-based Adversarial Patch and Pixle Attack.

V. CONCLUSION

In this article, we address the current research on SAR AEs. It is noted that existing studies mainly focus on the image domain, where the generated perturbations are difficult to implement in the physical world. To overcome this limitation, we propose two attack modules: 1) The SAR sticker, which focuses on the target surface; and 2) the strong scatterer, which targets the background region. Taking into account the physical characteristics of adversarial perturbations, SAR stickers can be implemented by incorporating specialized reflective materials onto the surface of the target. Similarly, strong scatterers can be strategically positioned around the target using materials such as corner reflectors. We separately evaluate the attack performance of the two types of AEs, as well as their combined attack performance. The results show that these physically realizable AEs can achieve a high fooling rate of 90% for SAR ATR models, along with notable adversarial transferability. The final discussion section demonstrates the robustness and efficacy of the proposed methods across various experimental dimensions. In future work, we will explore further refinement of the generation rules for SAR images to improve the alignment between the perturbations in the SAR sticker and the target vehicles.

References

[1] Z. Yue et al., "A novel semi-supervised convolutional neural network method for synthetic aperture radar image recognition," *Cogn. Comput.*, vol. 13, 795–806, 2021.

[2] F. Ma, F. Zhang, D. Xiang, Q. Yin, and Y. Zhou, "Fast task-specific region merging for SAR image segmentation," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, pp. 1–16, 2022.

[3] Y. Zhou, H. Liu, F. Ma, Z. Pan, and F. Zhang, "A sidelobe-aware small ship detection network for synthetic aperture radar imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–16, 2023.

[4] F. Ma, X. Sun, F. Zhang, Y. Zhou, and H.-C. Li, "What catch your attention in SAR images: Saliency detection based on soft-superpixel lacunarity cue," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, pp. 1–17, 2023.

[5] F. Gao et al., "SAR target incremental recognition based on features with strong separability," *IEEE Trans. Geosci. Remote Sens.*, vol. 62, pp. 1–13, 2024.

[6] H. Huang, F. Gao, J. Sun, J. Wang, A. Hussain, and H. Zhou, "Novel category discovery without forgetting for automatic target recognition," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 17, pp. 4408–4420, 2024.

[7] S. Chen, H. Wang, F. Xu, and Y.-Q. Jin, "Target classification using the deep convolutional networks for SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4806–4817, Aug. 2016.

[8] Y. Li, J. Wang, Y. Xu, H. Li, Z. Miao, and Y. Zhang, "DeepSAR-Net: Deep convolutional neural networks for SAR target recognition," in *Proc. IEEE 2nd Int. Conf. Big Data Anal.*, 2017, pp. 740–743.

[9] W. Wang, C. Zhang, J. Tian, J. Ou, and J. Li, "A SAR image target recognition approach via novel SSF-Net models," *Comput. Intell. Neurosci.*, vol. 2020, 2020, Art. no. 8859172.

[10] H. Chen, F. Zhang, B. Tang, Q. Yin, and X. Sun, "Slim and efficient neural network design for resource-constrained SAR target recognition," *Remote Sens.*, vol. 10, 2018, Art. no. 1618.

[11] F. Zhang, Y. Wang, J. Ni, Y. Zhou, and W. Hu, "SAR target small sample recognition based on CNN cascaded features and AdaBoost rotation forest," *IEEE Geosci. Remote Sens. Lett.*, vol. 17, no. 6, pp. 1008–1012, Jun. 2020.

[12] C. Szegedy et al., "Intriguing properties of neural networks," 2013, *arXiv:1312.6199*.

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

[14] J. Wang, "Adversarial examples in physical world," in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 4925–4926.

[15] Y. Dong et al., "Boosting adversarial attacks with momentum," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 9185–9193.

[16] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," 2017, *arXiv:1706.06083*.

[17] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter, "Accessorize to a crime: Real and stealthy attacks on state-of-the-art face recognition," in *Proc. ACM SIGSAC Conf. Comput. Commun. Secur.*, 2016, pp. 1528–1540.

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

[19] R. Duan, X. Ma, Y. Wang, J. Bailey, A. K. Qin, and Y. Yang, "Adversarial camouflage: Hiding physical-world attacks with natural styles," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 997–1005.

[20] A. Liu et al., "Perceptual-sensitive GAN for generating adversarial patches," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 1028–1035.

[21] M. Datcu, Z. Huang, A. Anghel, J. Zhao, and R. Cacoveanu, "Explainable physics-aware trustworthy artificial intelligence: A paradigm shift for synthetic aperture radar," *IEEE Geosci. Remote Sens. Mag.*, vol. 11, no. 1, pp. 8–25, Mar. 2023.

[22] T. Jianglan, L. I. U. Jialei, M. A. Jiazhi, S. H. I. Longfei, and G. Yifu, "Discrete spectrum cover signal waveform design and generation method," *J. Radars*, vol. 12, no. 6, pp. 1275–1289, 2023.

[23] X. I. E. Feng, L. I. U. Huanyu, H. U. Xikun, Z. Ping, and L. I. Junbao, "A radar anti-jamming method under multi jamming scenarios based on deep reinforcement learning in complex domains," *J. Radars*, vol. 12, no. 6, pp. 1290–1304, 2023.

[24] H. Li et al., "Adversarial examples for CNN-Based SAR image classification: An experience study," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 1333–1347, 2021.

[25] F. Zhang, T. Meng, D. Xiang, F. Ma, X. Sun, and Y. Zhou, "Adversarial deception against SAR target recognition network," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 4507–4520, 2022.

[26] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *Proc. IEEE Symp. Secur. Privacy*, 2017, pp. 39–57.

[27] B. Peng, B. Peng, J. Zhou, J. Xia, and L. Liu, "Speckle-variant attack: Toward transferable adversarial attack to SAR target recognition," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

[28] W. Xia, Z. Liu, and Y. Li, "SAR-PeGA: A generation method of adversarial examples for SAR image target recognition network," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 59, no. 2, pp. 1910–1920, Apr. 2023.

[29] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM:Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 618–626.

[30] F. Zhang, X. Yao, H. Tang, Q. Yin, Y. Hu, and B. Lei, "Multiple mode SAR raw data simulation and parallel acceleration for Gaofen-3 mission," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 11, no. 6, pp. 2115–2126, Jun. 2018.

[31] T. N. Praveen, M. V. Raju, B. D. Vishwanath, P. Meghana, T. R. Manjula, and G. Raju, "Absolute radiometric calibration of RISAT-1 SAR image using peak method," in *Proc. 2018 3rd IEEE Int. Conf. Recent Trends Electron., Inf. Commun. Technol. (RTEICT)*, 2018, pp. 456–460.

[32] B. Lewis, T. Scarnati, E. Sudkamp, J. Nehrbass, S. Rosencrantz, and E. Zelnio, "A SAR dataset for ATR development: The synthetic and measured paired labeled experiment (SAMPLE)," *Proc. SPIE*, vol. 10987, pp. 39–54, 2019.

[33] D. Malmgren-Hansen, A. Kusk, J. Dall, A. A. Nielsen, R. Engholm, and H. Skriver, "Improving SAR automatic target recognition models with transfer learning from simulated data," *IEEE Geosci. Remote Sens. Lett.*, vol. 14, no. 9, pp. 1484–1488, Sep. 2017.

[34] J. Pomponi, S. Scardapane, and A. Uncini, "Pixle: A fast and effective black-box attack based on rearranging pixels," in *Proc. Int. Joint Conf. Neural Netw.*, Padua, Italy, 2022, pp. 1–7.

[35] F. Croce et al., "Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks," in *Proc. 37th Int. Conf. Mach. Learn.*, vol. 119, 2020, pp. 2206–2216.

Fan Zhang (Senior Member, IEEE) received the B.E. degree in communication engineering from the Civil Aviation University of China, Tianjin, China, in 2002, the M.S. degree in signal and information processing from Beihang University, Beijing, China, in 2005, and the Ph.D. degree in signal and information processing from the Institute of Electronics, Chinese Academy of Sciences, Beijing, in 2008.

He is a Full Professor of electronic and information engineering with the Beijing University of Chemical Technology, Beijing. His research interests include remote sensing image processing, high-performance computing, and artificial intelligence.

Dr. Zhang is also an Associate Editor for *IEEE Access* and a reviewer of IEEE Transactions on Geoscience and Remote Sensing, the *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, the *IEEE Geoscience and Remote Sensing Letters*, and the *Journal of Radars*.

Yameng Yu (Student Member, IEEE) is currently working toward the master's degree in information and communication engineering with the College of Information Science and Technology, Beijing University of Chemical Technology, Beijing, China.

His research interests include SAR target recognition and adversarial attack.

Fei Ma (Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electronic and information engineering from the Beijing University of Aeronautics and Astronautics (BUAA), Beijing, China, in 2013, 2016, and 2020 respectively.

He is currently an Associate Professor with the College of Information Science and Technology, Beijing University of Chemical Technology, Beijing, China. His research interests include radar signal processing, image processing, machine learning, and target detection.

Yongsheng Zhou (Member, IEEE) received the B.E. degree in communication engineering from the Beijing Information Science and Technology University, Beijing, China, in 2005, and the Ph.D. degree in signal and information processing from the Institute of Electronics, Chinese Academy of Sciences, Beijing, China, in 2010.

From 2010 to 2019, he was with the Academy of Opto-Electronics, Chinese Academy of Sciences, and is currently a Professor of electronic and information engineering with the College of Information Science and Technology, Beijing University of Chemical Technology, Beijing, China. His research interests include target detection and recognition from microwave remotely sensed images, digital signal, and image processing.