JARS Journal of Applied Remote Sensing

RESEARCH PAPER

Fast sparse adversarial attack for synthetic aperture radar target recognition

Xuanshen Wan, Wei Liu,* Chaoyang Niu, Wanjie Lu, and Yuanli Li
PLA Information Engineering University, Institute of Data and Target Engineering, Zhengzhou, China

ABSTRACT. With the rapid development of artificial intelligence technology, deep learning has achieved significant advantages in synthetic aperture radar automatic target recognition (SAR-ATR). However, previous research showed that the addition of small perturbations not easily detected by the human eye can lead to SAR-ATR model recognition errors; that is, they are affected by adversarial attacks. To solve the problem of long computation time in existing SAR sparse adversarial attack algorithms, we propose a SAR fast sparse adversarial attack (FSAA) algorithm. First, an end-to-end sparse adversarial attack framework is developed based on the lightweight generator ResNet model using two different upsampling modules to control the amplitude and position of the adversarial perturbation. A loss function for the generator is then constructed, which mainly consists of the linear addition of the attack loss, the amplitude distortion loss, and the sparsity loss. Finally, the SAR image is mapped through the trained generator model in a one-step process to generate sparse adversarial perturbations quickly and effectively. Compared with the existing SAR sparse adversarial attack algorithm, the experimental results show that the generation speed of the proposed method is at least 30 times higher when the perturbation is less than 0.05% of the pixels in the entire image, and the recognition rate of the model is >13%.

Keywords: synthetic aperture radar; automatic target recognition; adversarial attack; sparsity; ResNet generator

Paper 240368G received Jun. 13, 2024; revised Oct. 16, 2024; accepted Nov. 13, 2024; published Dec. 4, 2024.

1 Introduction

Synthetic aperture radar (SAR) is advantageous as it can acquire target images under all weather conditions; hence, it is widely used in military and civilian applications.[1–3] Recently, deep neural network (DNN) models have achieved remarkable results in SAR automatic target recognition (ATR).[4–10] However, the latest research shows that SAR-ATR based on DNN models has security issues; that is, they are vulnerable to attacks by adversarial examples.[11] Research on SAR image adversarial attack algorithms helped to better understand the working mechanisms and internal decision-making of the SAR-ATR model and contributed to developing a more robust SAR-ATR model.

In a previous study, researchers first examined adversarial examples of attacks on neural network models in optical images.[12] Szegedy et al.[13] were the first to effectively reduce the success rate of DNN model recognition by adding tiny, carefully crafted perturbations to the original images. To solve the problem of perturbed pixels, which constitute a large proportion of existing adversarial example generation methods, some researchers have devoted themselves

*Address all correspondence to Wei Liu, greatliuliu@163.com

Handling Editor: Nicola Acito, Associate Editor

to the study of sparse adversarial attack algorithms in recent years. Such algorithms only need to change a small number of pixels in an image to perform an adversarial attack. Su et al.[14] proposed a single-pixel adversarial perturbation generation method based on differential evolution. They considered an extreme condition where changing just one pixel in the image could enable an effective attack on a DNN model. Modas et al.[15] proposed the SparseFool algorithm. The experimental results showed that SparseFool could effectively improve the success rate of the attacks.

In recent years, SAR-ATR adversarial example generation methods have gradually become key research areas. Huang et al.[11] used the fast gradient sign method[16] and basic iterative method[17] algorithms to prove that SAR images are vulnerable to adversarial example attacks. To improve stealth attacks, researchers have limited the perturbation area to the target area. Meng et al.[18] proposed the target region perturbation generator (TRPG) algorithm, which first uses the Gabor algorithm to perform texture segmentation on the SAR image to obtain the mask of the target area and then constructs the perturbation in the target area. Du et al.[19] used the maximum between-class variance method to complete the labeling of target and background regions at the pixel level, which enabled attackers to generate SAR image adversarial examples by adding small-scale perturbations to specific regions. Peng et al.[20] proposed a SAR target-segmentation-based adversarial attack (TSAA), which added perturbations only in the target area and successfully attacked the mainstream DNN model. Zhou et al.[21] further narrowed the scope of perturbations and successfully attacked the mainstream DNN model on the moving and stationary target acquisition and recognition (MSTAR) dataset using an algorithm from Ref. 15. In recent studies, Huang et al.[22] proposed a new method called intra-class transformation and inter-class nonlinear fusion attack. Meanwhile, Wan et al.[23] introduced the transferable universal adversarial network, which is based on the concept of generative adversarial networks. This method utilizes a dual-game framework between a generator and a discriminator to construct adversarial perturbations. However, both approaches are classified as global attacks, which come with the drawback of large perturbation ranges.

However, existing SAR sparse adversarial attack algorithms require a lot of time for iteration and optimization and are therefore not suitable for SAR adversarial attack scenarios with high real-time requirements. This study proposes a fast sparse adversarial attack (FSAA) algorithm that designs a generator-based sparse adversarial attack framework and uses two different upsampling modules to control the amplitude and location of the perturbations. The constructed loss function was used to guide the generator to update the parameters. This effectively reduced the amplitude of the perturbation and the number of perturbation pixels and improved the attack concealment of the adversarial samples. In addition, after the generator is trained, it only needs to map the input sample through the generator model in one step to quickly and effectively generate sparse adversarial perturbations in the SAR image.

The main contributions of this study are as follows:

1. In contrast to existing iteration-based sparse adversarial attack algorithms, this study uses the designed generator model to obtain adversarial perturbations by mapping the input samples in one step, which saves considerable time and improves the efficiency of the attack.
2. In the construction of the loss function, the proposed algorithm uses the L_2-norm and L_1-norm to reduce the amplitude of the perturbation and the number of perturbed pixels to further improve the concealment of adversarial examples.
3. The MSTAR dataset was used to evaluate the proposed algorithm. The experimental results show that FSAA can effectively attack the DNN model by perturbing less than eight pixels within 0.0025 s. At the same time, the recognition rate of the DNN model is less than 13%.

The remainder of this paper is organized as follows: Sec. 2 explains the principle of the algorithm in detail, Sec. 3 presents the experimental results and analysis, and Sec. 4 presents the conclusions.

2 Method

A general flowchart of the FSAA proposed in this study is shown in Fig. 1. First, the original SAR image x is input into the generator $G(\cdot)$ to obtain the adversarial perturbation image δ,

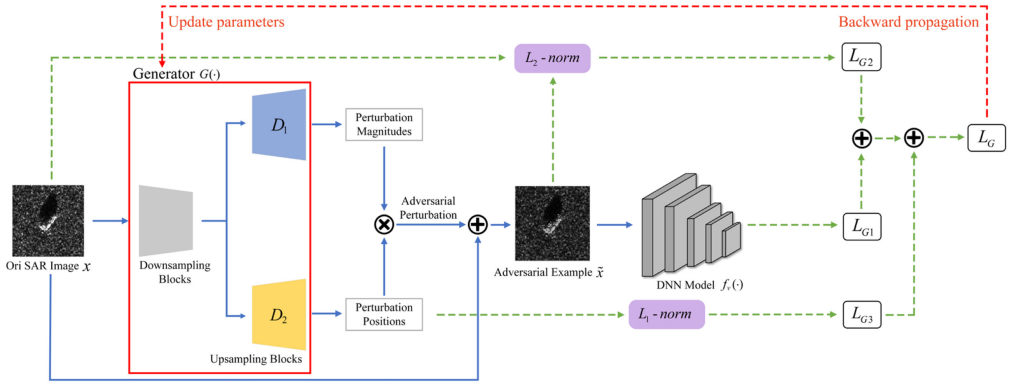

Fig. 1 Overall framework of the FSAA algorithm.

following which the adversarial example $\tilde{x}$ is obtained by adding x and δ to effectively attack the DNN-based SAR-ATR model $f_v(\cdot)$. The designed generator includes one downsampling module and two upsampling modules. The upsampling module of the upper-branch D_1 is mainly used to generate the amplitude value of the adversarial perturbation, and the upsampling module of the lower-branch D_2 is mainly used to generate the position information of the adversarial perturbation.

2.1 Structure of the Generator

The essence of the generator is an encoder and decoder model. In this paper, the choice of this structure mainly considers the following two factors: First, as the size of the adversarial perturbation should match the size of the original SAR image, the input and output sizes of the generator must be consistent; second, to improve the real-time performance of SAR attacks, the structure of the generator must be designed to choose a lightweight model. As shown in Fig. 2 and Table 1, the FSAA algorithm selects the ResNet[24] model as the main structure of the generator and modifies it based on this structure to fit the algorithm in this paper. Specifically, the upsampling module of the generator is divided into two parts, D_1 and D_2. The output of D_1 is a vector diagram representing the amplitude of the perturbation. This module is mainly used to control the perturbation amplitude of each pixel in the perturbation image. The main purpose of D_2 is to generate a sparse perturbation image. The output ρ is converted into a binary discrete vector $R \in \{0,1\}$ by setting a hyperparameter γ. The specific equation is as follows:

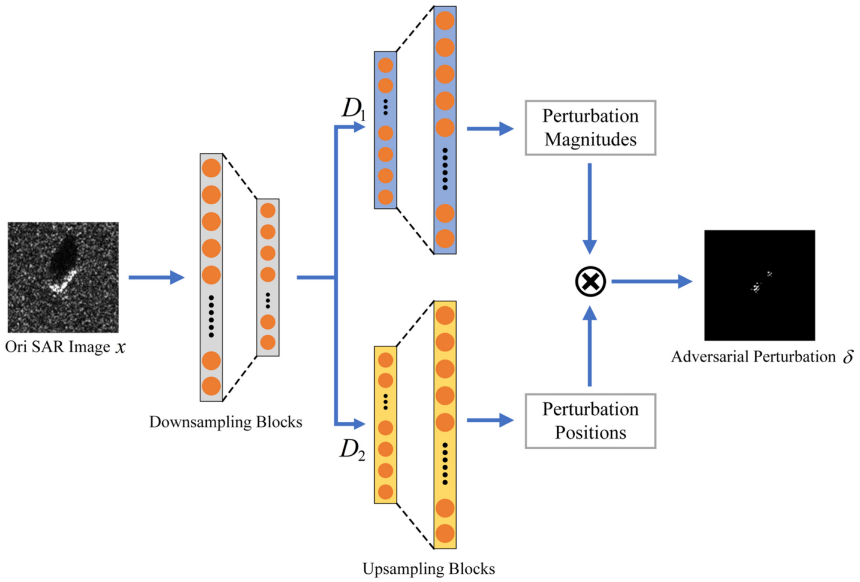

Fig. 2 Schematic of the generator structure.

Table 1 Input–output relationships for each module of ResNet.

Module	Input size	Output size
Input	1 × 128 × 128	64 × 128 × 128
Downsampling_1	64 × 128 × 128	128 × 64 × 64
Downsampling_2	128 × 64 × 64	256 × 32 × 32
Residual_1~6	256 × 32 × 32	256 × 32 × 32
Upsampling_1	256 × 32 × 32	128 × 64 × 64
Upsampling_2	128 × 64 × 64	64 × 128 × 128
Output	64 × 128 × 128	1 × 128 × 128

$$R_i = \begin{cases} 0, & R_i \le \gamma \\ 1, & R_i > \gamma \end{cases}, \tag{1}$$

where R_i represents the binary value of the i'th position.

2.2 Loss Function Design

A loss function L_G was designed for the generator, which is mainly composed of the linear addition of the attack loss L_{G1}, amplitude distortion function L_{G2}, and sparse loss L_{G3}. We set the weight value for each part of the loss function. The specific equation for the loss function is as follows:

$$L_G = \lambda_1 \cdot L_{G1} + \lambda_2 \cdot L_{G2} + \lambda_3 \cdot L_{G3}, \tag{2}$$

where λ_1, λ_2, and λ_3 are the weights of L_{G1}, L_{G2}, and L_{G3}, respectively.

First, the attack loss function L_{G1} is introduced. To improve the effectiveness of the adversarial examples, it is necessary to increase the confidence of the DNN model in identifying $\tilde{x}$ as other categories and decrease the confidence of $\tilde{x}$ being identified as the true category C_{tr}. Therefore, the equation of L_{G1} is as follows:

$$L_{G1}(f_v(\tilde{x}), C_{\text{tr}}) = -\log\left(\frac{\sum_{i \neq C_{\text{tr}}} \exp(f_v(\tilde{x})_i)}{\sum_i \exp(f_v(\tilde{x})_i)}\right). \tag{3}$$

Next, the amplitude distortion loss is defined as L_{G2}. In this study, the L_2 norm was introduced to measure the degree of distortion of the original SAR image x and the adversarial example $\tilde{x}$ to ensure that the adversarial example generated by the algorithm in this study cannot be detected by the human eye. The equation is expressed as follows:

$$\begin{aligned} L_{G2}(x, \tilde{x}) &= \|\tilde{x} - x\|_2 \\ &= \left(\sum_i |\Delta x_i|^2\right)^{\frac{1}{2}}. \end{aligned} \tag{4}$$

Finally, to improve the sparsity of adversarial perturbations, as shown in Eq. (5), the L_1-norm is used in this paper to limit the number of non-zero elements in the binary discrete vector R. As R only contains the values 0 and 1, a value of 1 indicates that the pixel value at that position is perturbed, and a value of 0 implies that the pixel value at that position is not perturbed

$$L_{G3} = \|R\|_1. \tag{5}$$

2.3 Training Process of the Generator

In this section, the entire training process of the generator is described in detail. Specifically, a dataset χ and training batch size M are provided. χ is randomly divided into N batches $\{s_1, s_2, \ldots, s_N\}$ according to M. s_i represents all SAR images in each batch in dataset χ. The loss function defined in Sec. 2.2 is then used to continuously update the parameters of the

Algorithm 1 Complete training process of the generator.

Input: Dataset χ; surrogate model $f_v(\cdot)$; batch size M; true class C_{tr}; training iteration number T; learning rate η; training loss function of the generator L_G.

Output: The parameter θ_G of the well-trained generator.

1: Randomly initialize θ_G

2: **For** $t = 1$ to T **do**

3: According to M, randomly divide χ into N batches$\{s_1, s_2, \ldots, s_N\}$

4: **For** $n = 1$ to N **do**

5: Calculate $L_G(\theta_G, f_v, s_n, C_{\text{tr}})$

6: Update $\theta_G = \theta_G - \eta \cdot \partial/\partial\theta_G \cdot L_G$

7: **End For**

8: **End For**

generator $G(\cdot)$. Finally, the parameter information of the generator is saved. Therefore, in the test phase, only a one-step mapping of the generator is required to generate sparse adversarial perturbation images.

3 Experiments

3.1 Dataset and Implementation Details

3.1.1 *Dataset*

The experiment used the MSTAR[25] dataset. This dataset was published by the Defense Advanced Research Projects Agency in 1996 and contains SAR images of Soviet military vehicles at different azimuths and depression angles. As shown in Table 2, the MSTAR dataset contains 10 categories of military targets under standard operating conditions (SOCs). The training dataset contained 2747 images acquired at a depression angle of 17 deg, and the test dataset contained 2426 images acquired at a depression angle of 15 deg. Figure 3 shows the SAR images of each target category in the MSTAR dataset.

Table 2 Details of the MSTAR dataset under SOCs.

	Training data		Testing data	
Target class	Depression angle (deg)	Number	Depression angle (deg)	Number
2S1	17	299	15	274
BMP2	17	233	15	196
BRDM2	17	298	15	274
BTR60	17	256	15	195
BTR70	17	233	15	196
D7	17	299	15	274
T62	17	299	15	273
T72	17	232	15	196
ZIL131	17	299	15	274
ZSU234	17	299	15	274

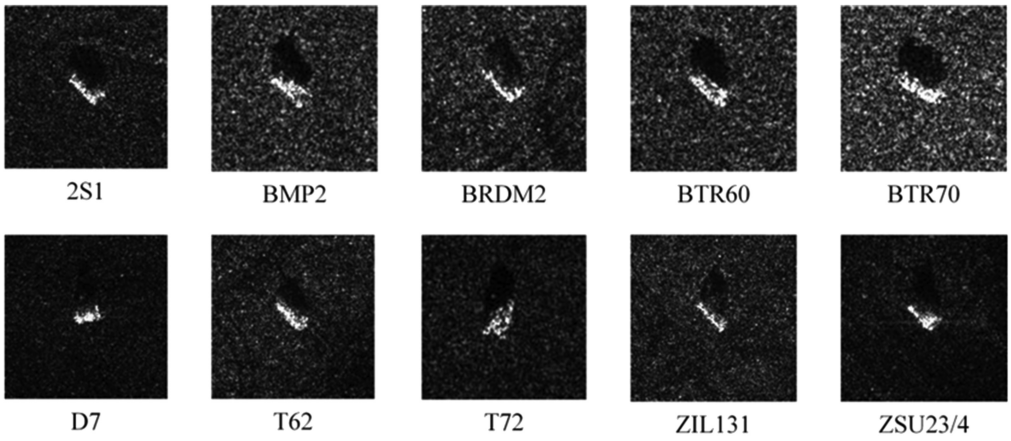

Fig. 3 SAR images of the MSTAR dataset.

3.1.2 *Implementation details*

For the selection of the DNN model, the proposed algorithm was evaluated with five typical DNN models: DenseNet121,[26] GoogLeNet,[27] InceptionV3,[28] ResNet50,[29] and Shufflenet.[30] For data preprocessing, all images of the MSTAR dataset in the experiment were resized to 128×128 pixels, and 10% of the training dataset was randomly sampled to obtain the validation dataset. When training the DNN recognition model, the number of training rounds and the batch size were set to 50 and 32, respectively, and the learning rate was set to 0.001. As shown in Fig. 4, the classification accuracies of the five DNN models of the MSTAR test dataset are 98.72%, 98.06%, 96.17%, 97.98%, and 96.66%, respectively.

In the baseline comparison method setting, to verify the effectiveness of the algorithm in this study, four SAR sparse adversarial attack algorithms were selected for comparative analysis: Local aggregative attack (LAA),[19] SparseFool,[7,16] TRPG,[18] and TSAA.[20] The parameters of the individual algorithms were set according to the literature. A Windows 10 operating system,

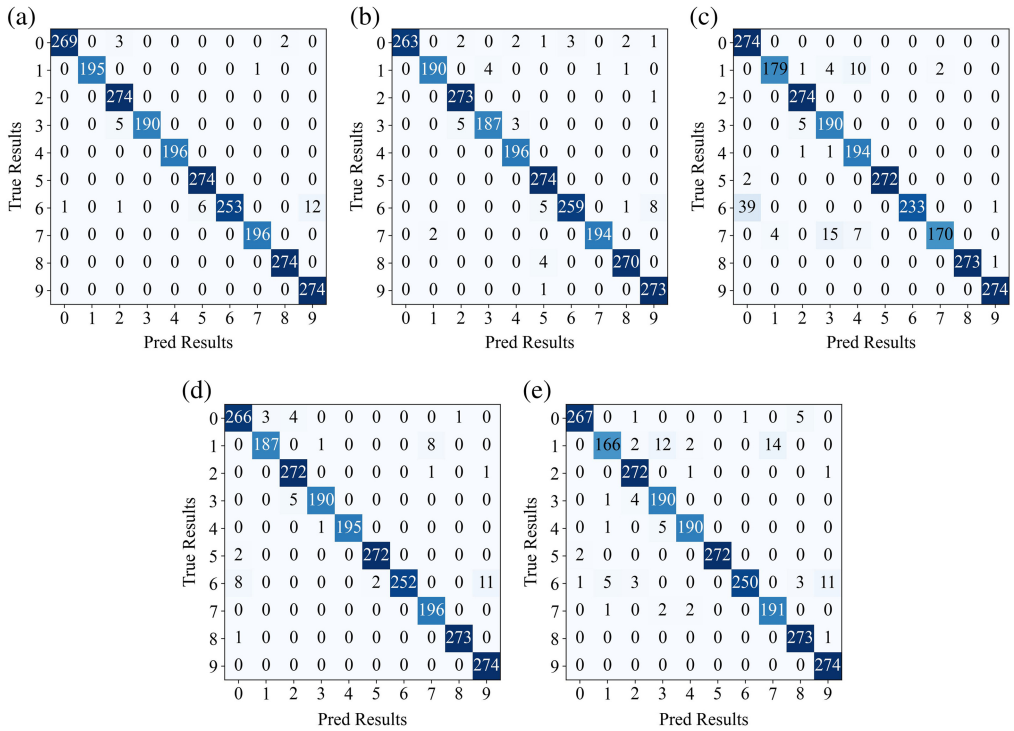

Fig. 4 Confusion matrix of DNN models on the MSTAR dataset. (a) DenseNet121; (b) GoogLeNet; (c) InceptionV3; (d) ResNet50; and (e) Shufflenet.

PyTorch deep learning development framework, and Python as the development language were used for the experiment. The CPU used in the experiment was an Intel Core i9-11900H and the GPU was an NVIDIA GeForce RTX 3080 Laptop GPU.

3.2 Evaluation Metrics

First, the effectiveness of the attack was measured based on the attack success rate $\tilde{A}cc$. The value of $\tilde{A}cc$ reflects the probability that the DNN model recognizes the adversarial example $\tilde{x}$ as a true category C_{tr}. Hence, the smaller the value, the higher the attack success rate of the adversarial example. The specific equation is as follows:

$$\tilde{A}cc = \frac{1}{N}\sum_{n=1}^{N} Z(\arg\max_i(f_v(\tilde{x}_n)_i) == C_{\text{tr}}), \tag{6}$$

where C_{tr} represents the true category of the input sample, N represents the number of samples, and $Z(\cdot)$ represents the discriminant function. When this condition is met, the output value is 1; otherwise, it is 0.

The second is the concealment of attacks. As shown in the following equation, the structural similarity (SSIM)[31] is used to measure the similarity between the original input sample and the adversarial example. The higher the structural similarity, the better the concealment of the attack. The equation is expressed as follows:

$$\text{SSIM}(x, \tilde{x}) = \frac{(2\mu_x\mu_{\tilde{x}} + C_1)(2\sigma_{x\tilde{x}} + C_2)}{(\mu_x^2 + \mu_{\tilde{x}}^2 + C_1)(\sigma_x^2 + \sigma_{\tilde{x}}^2 + C_2)}, \tag{7}$$

where μ_x, $\mu_{\tilde{x}}$ and σ_x, $\sigma_{\tilde{x}}$ are the mean and standard deviation of the corresponding images, respectively; $\sigma_{x\tilde{x}}$ represents the covariance; C_1 and C_2 are constants used to maintain the stability of the metric and are generally set to values close to 0.

In addition, the sparsity S is introduced to calculate the proportion of changed pixel points to the total number of image points:

$$S = \frac{N_\delta}{N_{\text{total}}}, \tag{8}$$

where N_δ represents the number of modified pixels, and N_{total} represents the total number of pixels in the image.

Finally, to evaluate the real-time performance of the attack, T_{attack} is introduced to measure the time required to generate a single adversarial example. The equation is as follows:

$$T_{\text{attack}} = \frac{\text{Time}}{N}, \tag{9}$$

where Time represents the total time required to generate N adversarial examples.

3.3 Attack Performance Comparison

To verify the attack effectiveness and concealment of the FSAA algorithm proposed in this paper, the attack performance of the different algorithms on the five DNN models in Sec. 3.1.2 is examined in this section. Table 3 lists the attack effectiveness of the different algorithms. Overall, the proposed algorithm shows the strongest attack effectiveness for each DNN model. Taking the GoogLeNet model as an example, the recognition rate of the DNN model on the adversarial example constructed by the proposed algorithm was 4.58%, and the lowest recognition rate of the baseline algorithm was 12.24%. Compared with the baseline algorithm, the proposed algorithm improved the attack effectiveness by 7.66%. We believe this improvement originates from the attack loss function designed in this study, which can effectively guide the generator to construct adversarial examples with strong attack performance. Second, the experimental results in terms of attack concealment are listed in Table 4. The proposed algorithm achieved the best concealment when attacking each DNN model. Taking the attack on the InceptionV3 model as an example, the SSIM value between the adversarial example generated by the algorithm in this study and the original example was 0.9896, and the highest SSIM value of the comparison algorithm was 0.9892. Thus, the higher the SSIM value, the higher the similarity between the

Table 3 Attack effectiveness of different algorithms on DNN models.

Model	DenseNet121 (%)	GoogLeNet (%)	InceptionV3 (%)	ResNet50 (%)	Shufflenet (%)
LAA	22.75	12.77	14.47	14.41	27.18
SparseFool	18.91	15.34	15.69	14.18	14.88
TRPG	10.53	12.24	25.97	10.33	28.26
TSAA	24.21	26.53	20.78	14.67	17.39
FSAA	**10.18**	**4.58**	**11.38**	**9.69**	**12.82**

Note: bold values indicate the optimal values.

Table 4 Attack concealment of different algorithms on DNN models.

Model	DenseNet121	GoogLeNet	InceptionV3	ResNet50	Shufflenet
LAA	0.9865	0.9854	0.9887	0.9885	0.9825
SparseFool	0.9802	0.9811	0.9827	0.9830	0.9840
TRPG	0.9872	0.9832	0.9892	0.9829	0.9838
TSAA	0.9884	0.9819	0.9879	0.9873	0.9822
FSAA	**0.9885**	**0.9860**	**0.9896**	**0.9893**	**0.9858**

Note: bold values indicate the optimal values.

Table 5 Sparsity of adversarial perturbations generated by different algorithms.

Model	DenseNet121 (%)	GoogLeNet (%)	InceptionV3 (%)	ResNet50 (%)	Shufflenet (%)
LAA	0.19	0.21	0.20	0.19	0.15
SparseFool	0.16	0.19	0.31	0.18	0.25
TRPG	0.48	0.62	1.41	0.40	1.34
TSAA	4.37	4.31	4.36	4.56	4.14
FSAA	**0.04**	**0.02**	**0.02**	**0.03**	**0.02**

Note: bold values indicate the optimal values.

adversarial example and original sample, that is, the better the concealment of the adversarial example. Compared with the baseline algorithm, the proposed algorithm improved the attack concealment by 0.0004. We believe that this improvement arises from the amplitude distortion loss function, which can greatly reduce the amplitude of the perturbation, thereby improving the similarity between the adversarial example and original sample. The experimental results for perturbation sparsity are listed in Table 5. The sparsity of the algorithms proposed in this study was less than 0.05%, whereas the lowest sparsity of the comparison algorithm was 0.15%. Thus, the proposed algorithm significantly improved the sparsity of the perturbation. This is because the loss function L_{G3} uses the L_1-norm to limit the number of perturbed pixels, thereby greatly improving the sparsity of the perturbation.

The following conclusions can be drawn from the analysis of the above experimental results: First, compared with the other four SAR sparse adversarial attack algorithms, the proposed algorithm can construct the best adversarial attack examples in terms of attack effectiveness. Second, in this study, to conceal the attack, the SSIM was introduced to measure the similarity between the adversarial examples and the original samples. The experimental results show that the

proposed algorithm has the highest SSIM. Therefore, the adversarial examples constructed by the proposed algorithm can maintain a high degree of similarity with the original samples; that is, the concealment is good. Third, in terms of the sparsity of the perturbation, to improve the physical feasibility of the algorithm, the proposed algorithm focuses on reducing the number of perturbed pixels when constructing the loss function; that is, it improves the sparsity of the adversarial perturbation. The results show that the proposed algorithm only needs to perturb less than eight pixels to perform an effective attack on the DNN model. Therefore, the proposed algorithm has the strongest sparsity among all SAR sparse adversarial attack algorithms.

3.4 Comparison of Real-time Performance

Following the defined equation for the attack time loss T_{attack} in Sec. 3.2, this section further evaluates the time loss of the different algorithms in constructing adversarial examples for the five DNN models. The experimental results are listed in Table 6. The time taken by the proposed algorithm to construct adversarial examples on all DNN models was less than 0.0025 s, and the fastest time among the compared algorithms was 0.0971 s. Therefore, the time cost of the proposed algorithm was the lowest when constructing a single adversarial example, and the computational speed increased by at least 30 times. This is because other SAR sparse adversarial attacks require numerous iterative operations to generate perturbed images. The proposed algorithm fully uses the mapping relationship of the generator in the design, and only needs to map the input example through the generator model in one step to obtain the adversarial perturbation image, effectively reducing the operation time.

3.5 Visualization of the Adversarial Examples

In this section, Shufflenet is used as an example to visualize adversarial perturbations and examples generated by different sparse adversarial attack algorithms on the MSTAR dataset, as shown in Fig. 5. Combined with the experimental conclusions in Sec. 3.3, the perturbed image is shown in the second row of Fig. 5. Compared with other SAR sparse adversarial attack algorithms, the perturbation constructed by the proposed algorithm requires the least number of image pixels to be changed. As shown in the figure, less than eight pixels need to be perturbed to perform an adversarial attack on the DNN model. In practical applications, an attacker can alter the image resulting from SAR imaging by adding absorbing or highly scattering materials around the target. Therefore, the sparse adversarial perturbation constructed in this study is physically feasible while reducing the time cost.

3.6 Ablation Study

In this section, the effect of amplitude distortion and loss of sparsity on the perturbations is further investigated. The experimental results are listed in Table 7. The SSIM, sparsity, and time loss are selected to measure the attack performance of the adversarial examples under different loss functions. From the data in the table, it is evident that when the amplitude loss function is missing, the SSIM of the adversarial examples constructed by the generator becomes lower than that of the original samples, and the sparsity and generation speed remain similar to those of FSAA. When the sparsity loss function is absent, the sparsity of the adversarial perturbation

Table 6 Time cost of generating a single adversarial example.

Model	DenseNet121 (s)	GoogLeNet (s)	InceptionV3 (s)	ResNet50 (s)	Shufflenet (s)
LAA	61.2398	36.1987	49.5689	31.8370	33.5758
SparseFool	1.3638	0.2581	0.2859	0.1368	0.0971
TRPG	1.9560	0.8819	0.9922	0.9920	0.7665
TSAA	7.5996	1.9757	2.4517	1.6717	1.5936
FSAA	**0.0022**	**0.0021**	**0.0022**	**0.0024**	**0.0021**

Note: bold values indicate the optimal values.

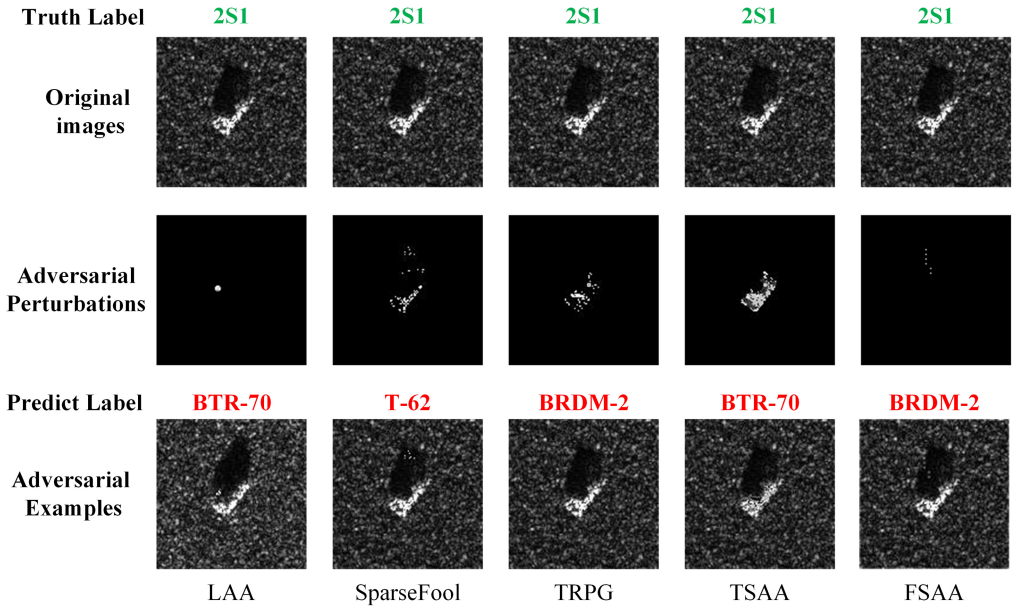

Fig. 5 Original SAR images, adversarial examples, and adversarial perturbations generated by different algorithms. The words in green represent the original correct label category, and the words in red represent the incorrect label category.

Table 7 Ablation study of FSAA on the MSTAR dataset.

Model	Attack	SSIM	Sparsity (%)	Time (s)
DenseNet121	W/o magnitude loss	0.9176	**0.04**	**0.0021**
	W/o sparsity loss	0.9761	100	0.0022
	FSAA	**0.9885**	**0.04**	0.0022
GoogLeNet	W/o magnitude loss	0.9045	0.03	**0.0021**
	W/o sparsity loss	0.9722	100	0.0024
	FSAA	**0.9860**	**0.02**	**0.0021**
InceptionV3	W/o magnitude loss	0.9682	0.04	**0.0022**
	W/o sparsity loss	0.9801	100	0.0026
	FSAA	**0.9896**	**0.02**	**0.0022**
ResNet50	W/o magnitude loss	0.8883	**0.03**	**0.0024**
	W/o sparsity loss	0.9778	100	0.0025
	FSAA	**0.9893**	**0.03**	**0.0024**
Shufflenet	W/o magnitude loss	0.9197	**0.02**	**0.0021**
	W/o sparsity loss	0.9821	100	0.0023
	FSAA	**0.9858**	**0.02**	**0.0021**

Note: bold values indicate the optimal values.

increases to 100%; that is, it becomes a global perturbation. At the same time, the SSIM is slightly lower than that of FSAA, but the generation speed remains at a similar level. Based on the above analysis, the amplitude distortion and sparse loss proposed in this study can effectively limit the amplitude of the perturbations and increase the sparsity, respectively.

4 Conclusion

In this paper, a fast sparse SAR adversarial attack algorithm called FSAA is proposed. The designed end-to-end sparse adversarial attack framework was used to quickly obtain adversarial perturbations from the input samples through one-step mapping. Compared with existing iteration-based SAR algorithms for sparse adversarial attacks, this algorithm significantly improved the speed of adversarial sample generation. In addition, a loss function for the generator was developed, which effectively guaranteed the success rate, concealment, and sparsity of the attack.

In the future, we will further investigate the SAR sparse adversarial attack algorithm in a black-box environment.

Disclosures

The authors declare no conflicts of interest.

Code and Data Availability

The data presented in this article are publicly available at https://figshare.com/s/c66a2f7925bae67607c9. The code generated or used during the study is available from the corresponding author by request.

Acknowledgments

We deeply appreciate the support of the National Natural Science Foundation of China (Grant No. 42201472). The authors also thank the editors and reviewers for sharing their expert opinions on our paper, which has benefited from their constructive comments and suggestions.

References

1. F. Zhang et al., "Multiple mode SAR raw data simulation and parallel acceleration for Gaofen-3 mission," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **11**(6), 2115–2126 (2018).
2. A. Moreira et al., "A tutorial on synthetic aperture radar," *IEEE Geosci. Remote Sens. Mag.* **1**(1), 6–43 (2013).
3. G. C. Anagnostopoulos, "SVM-based target recognition from synthetic aperture radar images using target region outline descriptors," *Nonlinear Anal. Theory Methods Appl.* **71**(12), 2934–2939 (2009).
4. D. Vint et al., "Automatic target recognition for low resolution foliage penetrating SAR images using CNNs and GANs," *Remote Sens.* **13**(4), 596 (2021).
5. C. Du et al., "Factorized discriminative conditional variational auto-encoder for radar HRRP target recognition," *Signal Process.* **158**, 176–189 (2019).
6. L. Wang et al., "Few-shot class-incremental SAR target recognition based on hierarchical embedding and incremental evolutionary network," *IEEE Trans. Geosci. Remote Sens.* **61**, 1–11 (2023).
7. J. Tang et al., "Incremental SAR automatic target recognition with error correction and high plasticity," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **15**, 1327–1339 (2022).
8. Y. Kwak, W.-J. Song, and S.-E. Kim, "Speckle-noise-invariant convolutional neural network for SAR target recognition," *IEEE Geosci. Remote Sens. Lett.* **16**(4), 549–553 (2019).
9. W. Zeng et al., "Multiview synthetic aperture radar target recognition method using joint sparse representation and random weight matrix," *J. Appl. Remote Sens.* **17**(1), 016513 (2023).
10. L. Zou et al., "Synthetic aperture radar target recognition via deep attention convolutional network assisted by multiscale residual despeckling network," *J. Appl. Remote Sens.* **17**(1), 016502 (2023).
11. T. Huang et al., "Adversarial attacks on deep-learning-based SAR image target recognition," *J. Netw. Comput. Appl.* **162**, 102632 (2020).
12. X. Peng et al., "IOPA-FracAT: research on improved one-pixel adversarial attack and fractional defense in hyperspectral image classification," in *36th Chinese Control and Decis. Conf. (CCDC)*, July, Xi'an, China, pp. 1527–1532 (2024).
13. C. Szegedy et al., "Intriguing properties of neural networks," arXiv, 19 February 2014. http://arxiv.org/abs/1312.6199 (accessed 15 November 2022).
14. J. Su, D. V. Vargas, and K. Sakurai, "One pixel attack for fooling deep neural networks," *IEEE Trans. Evol. Comput.* **23**(5), 828–841 (2019).
15. A. Modas, S.-M. Moosavi-Dezfooli, and P. Frossard, "SparseFool: a few pixels make a big difference," in *IEEE/CVF Conf. Comput. Vis. and Pattern Recognit. (CVPR)*, June, IEEE, Long Beach, CA, USA, pp. 9079–9088 (2019).

16. I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," arXiv, March 20, 2015. http://arxiv.org/abs/1412.6572 (accessed 15 November 2022).
17. A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," arXiv, February 10, 2017. http://arxiv.org/abs/1607.02533 (accessed 6 December 2022).
18. T. Meng, F. Zhang, and F. Ma, "A target-region-based SAR ATR adversarial deception method," in *7th Int. Conf. Signal and Image Process. (ICSIP)*, July, IEEE, Suzhou, China, pp. 142–146 (2022).
19. M. Du et al., "Local aggregative attack on SAR image classification models," in *IEEE 6th Adv. Inf. Technol., Electron. and Autom. Control Conf. (IAEAC)*, November, Beijing, China, pp. 1519–1524 (2022).
20. B. Peng et al., "Target segmentation based adversarial attack for SAR images," in *CIE Int. Conf. Radar (Radar)*, December, IEEE, Haikou, Hainan, China, pp. 2146–2150 (2021).
21. J. Zhou et al., "Sparse adversarial attack of SAR image," *J. Signal Process.* **37**(9), 1633–1643 (2021).
22. X. Huang, Z. Lu, and B. Peng, "Enhancing transferability with intra-class transformations and inter-class nonlinear fusion on SAR images," *Remote Sens.* **16**, 2539 (2024).
23. X. Wan et al., "Black-box universal adversarial attack for DNN-based models of SAR automatic target recognition," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.* **17**, 8673–8696 (2024).
24. J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," arXiv March 26, 2016. http://arxiv.org/abs/1603.08155 (accessed 24 March 2024).
25. E. R. Keydel, S. W. Lee, and J. T. Moore, "MSTAR extended operating conditions: a tutorial," *Proc. SPIE* **2757**, 228–242 (1996).
26. G. Huang et al., "Densely connected convolutional networks," in *IEEE Conf. Comput. Vis. and Pattern Recognit. (CVPR)*, July, IEEE, Honolulu, HI, pp. 2261–2269 (2017).
27. C. Szegedy et al., "Going deeper with convolutions," in *IEEE Conf. Comput. Vis. and Pattern Recognit. (CVPR)*, June, IEEE, Boston, MA, USA, pp. 1–9 (2015).
28. C. Szegedy et al., "Rethinking the inception architecture for computer vision," in *IEEE Conf. Comput. Vis. and Pattern Recognit. (CVPR)*, June, IEEE, Las Vegas, NV, USA, pp. 2818–2826 (2016).
29. S. Xie et al., "Aggregated residual transformations for deep neural networks," in *IEEE Conf. Comput. Vis. and Pattern Recognit. (CVPR)*, July, IEEE, Honolulu, HI, pp. 5987–5995 (2017).
30. X. Zhang et al., "ShuffleNet: an extremely efficient convolutional neural network for mobile devices," in *IEEE/CVF Conf. Comput. Vis. and Pattern Recognit.*, June, Salt Lake City, UT, IEEE, pp. 6848–6856 (2018).
31. Z. Wang et al., "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.* **13**(4), 600–612 (2004).

Xuanshen Wan is currently working toward an MS degree in information and communication engineering with the School of Data and Target Engineering, Information Engineering University, Zhengzhou, China. His research interests include SAR adversarial attack and synthetic aperture radar automatic target recognition (SAR-ATR).

Wei Liu received his BS, MS, and PhD degrees from Information Engineering University, Zhengzhou, China, in 2001, 2003, and 2016, respectively. He is an associate professor at Information Engineering University, Zhengzhou, China. His research interests include pattern recognition, remote sensing information processing, and deep learning.

Chaoyang Niu received his BS and MS degrees in information engineering from Zhengzhou Information Technology Institute, Henan, in 2003 and 2006, respectively, and his PhD in signal and information processing from Zhengzhou Institute of Surveying and Mapping, Henan, in 2011. In 2016, he became an associate professor with the Data and Target Engineering Institute, Information Engineering University. His research interests include pattern recognition, UAV remote sensing, and optical and radar imagery processing.

Wanjie Lu received his BS degree in photogrammetry and remote sensing and his PhD in surveying and mapping from the Information Engineering University, Zhengzhou, China, in 2016 and 2020, respectively. He is currently a lecturer at the Data and Target Engineering Institute, Information Engineering University, Zhengzhou. His research interests include UAV remote sensing, image processing, deep learning algorithms, and spatial information services.

Yuanli Li received her BS degree in electronic and information engineering from the Henan Polytechnic University, Jiaozuo, China, in 2022. She is now studying for an MS degree at Information Engineering University. Her research interests include semantic descriptions of SAR images.