remote sensing

MDPI

Article

BUM: Bayesian Uncertainty Minimization for Transferable Adversarial Examples in SAR Recognition

Hongqiang Wang [1,2,3], Yuqing Lan [1,3,*], Fuzhan Yue [2], Zhenghuan Xia [2] and Tao Zhang [2]

[1] School of Software, Beihang University, Beijing 100191, China
[2] State Key Laboratory of Space Information System and Integrated Application, Beijing Institute of Satellite Information Engineering, Beijing 100095, China
[3] Jiangxi Research Institute, Beihang University, Beijing 100191, China
* Correspondence: lanyuqing@buaa.edu.cn

Highlights

What are the main findings?

- We derive a generalization error bound for adversarial transferability, proving that minimizing the surrogate's epistemic uncertainty mathematically tightens the transfer gap to unknown black-box models.
- Guided by this theory, the proposed Bayesian Uncertainty Minimization framework achieves state-of-the-art attack success rates on MSTAR and FUSAR datasets, successfully bridging the challenging transferability barrier between CNNs and Vision Transformers.

What is the implication of the main finding?

- Minimizing epistemic uncertainty forces the generation of structure-aware perturbations that target semantic scattering centers shared across architectures, ensuring both high attack efficacy and statistical stealthiness.
- The results reveal a critical, architecture-agnostic vulnerability in current SAR recognition systems, suggesting that future defense strategies must incorporate uncertainty estimation to resist generalized attacks.

Abstract

Adversarial examples pose a significant threat to Deep Neural Networks (DNNs) underpinning Synthetic Aperture Radar (SAR) Automatic Target Recognition (ATR) systems, as these models exhibit acute susceptibility to such malicious inputs. While white-box attacks achieve high success rates, their transferability to unknown black-box models—particularly across different network architectures (e.g., from CNNs to Vision Transformers)—remains a significant challenge. Existing gradient-based iterative methods often overfit the specific decision boundary of the surrogate model, resulting in poor generalization. To address this, we propose a novel generative attack framework termed BUM. Instead of merely maximizing the classification error, BUM explicitly models and minimizes the epistemic uncertainty of the surrogate model. By leveraging Monte Carlo (MC) Dropout to simulate a Bayesian ensemble, we train a generator to craft perturbations that are consistently adversarial across stochastic sub-models. This regularization forces the attack to target high-level, structure-aware semantic features shared among architectures, rather than low-level, model-specific artifacts. Extensive experiments on the MSTAR and FUSAR datasets demonstrate the superior black-box transferability of BUM.

Check for updates

Academic Editor: John Trinder

Received: 13 January 2026
Revised: 10 February 2026
Accepted: 17 February 2026
Published: 26 February 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

Keywords: adversarial attack; black-box transferability; Bayesian Uncertainty; generative model

1. Introduction

SAR plays a pivotal role in modern military reconnaissance, battlefield surveillance, and remote sensing due to its capability for day-and-night, all-weather, and high-resolution imaging [1,2]. Target recognition technology based on SAR images is of significant importance for resource development, maritime cruising, and marine environmental protection [3–5].

In recent years, with the rapid development of deep learning technology, Convolutional Neural Networks (CNNs) have achieved superior performance surpassing traditional methods in SAR ATR tasks. However, despite the extremely high recognition accuracy demonstrated by deep neural networks, research indicates that they exhibit extreme vulnerability to adversarial examples. Attackers need only superimpose imperceptible perturbations onto the original images to mislead the model into making incorrect judgments. In safety-critical scenarios such as military confrontation, this vulnerability creates serious security risks, potentially leading to the evasion of high-value targets or inducing catastrophic false alarms [6–10]. In safety-critical scenarios such as military confrontation, this vulnerability creates serious security risks, potentially leading to the evasion of high-value targets or inducing catastrophic false alarms. As illustrated in Figure 1, a typical adversarial attack scenario involves an adversary generating a specific perturbation via an algorithm or device. By injecting this imperceptible noise into the SAR imagery of a high-value target (e.g., an M1 tank), the ATR system is manipulated into misclassifying it as a different entity (e.g., a T60). Such failures in recognition reliability can have strategic consequences in battlefield surveillance and maritime monitoring, which motivates our investigation into transferable attacks across these diverse architectures.

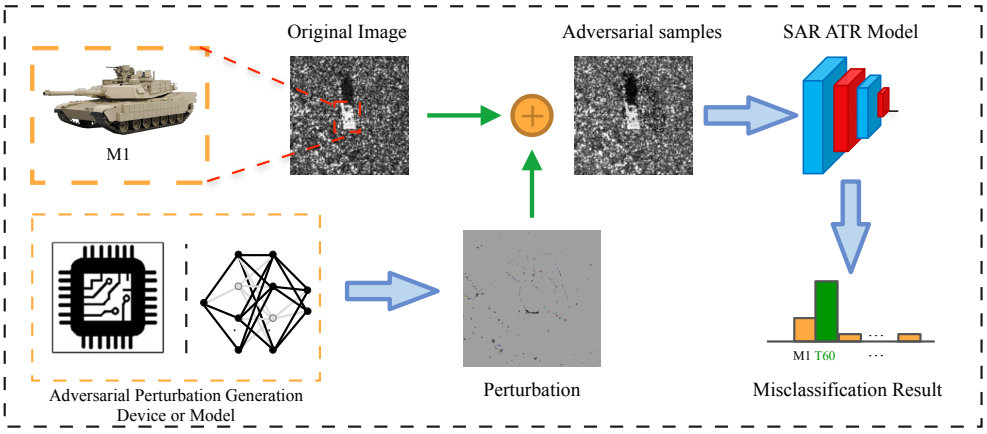


Figure 1. Schematic illustration of an adversarial attack scenario in SAR ATR. An adversary utilizes a perturbation generation device or model to craft a minute perturbation. When superimposing this perturbation onto the original SAR image of a military target (e.g., an M1 tank), the resulting adversarial example retains the visual appearance of the original but successfully misleads the SAR ATR model into an incorrect classification (e.g., misidentifying it as a T60). This demonstrates the severe security vulnerability of deep learning-based recognition systems in hostile environments.

Currently, in the field of optical imagery, numerous adversarial attack algorithms have been proposed. Based on the generation strategy of adversarial examples, these can be categorized into four types: gradient-based methods [11–15], optimization-based

methods [16–20], boundary-based methods [21–26], and Generative Adversarial Network (GAN)-based methods [27–32]: (1). Gradient-based methods utilize the model's gradient information to rapidly construct perturbations in the direction that increases the model's loss. Representative algorithms include FGSM [11], BIM [33], PGD [34], and MIM [35]. However, single-step attacks (like FGSM) produce coarser perturbations, while iterative attacks (like BIM), though stronger, incur higher computational costs and are prone to overfitting specific models; (2). Optimization-based methods model the generation of adversarial examples as a constrained optimization problem to find the minimal perturbation. Representative works include the C&W algorithm [16], L-BFGS [17], and the Fast C&W algorithm [19]. These methods are highly aggressive with small perturbations but come with very high computational costs and complex, time-consuming optimization processes. (3). Boundary-based methods iterate to estimate the distance from the sample point to the decision boundary and add perturbations along the boundary direction. A representative work is DeepFool [21]. These algorithms aim to generate minimal perturbations that are difficult for humans to perceive and are relatively efficient; however, errors in estimating the decision boundary may result in lower attack success rates on complex models compared to optimization methods. (4). GAN-based methods involve training a GAN whose generator can directly and rapidly produce adversarial examples. Representative works include AdvGAN [27], AdvGAN++ [29], Natural GAN [30], and Rob-GAN [31]. Once trained, these methods offer extremely fast generation speeds suitable for real-time attacks, but the training process itself is unstable, and the diversity of generated samples may be limited.

In contrast, adversarial attack algorithms targeting SAR recognition models are still in a nascent stage. Early research primarily focused on transferring adversarial attack algorithms from optical images to the SAR domain. Research in [36] demonstrates that while CNNs achieve very high success rates in SAR image classification tasks, they exhibit significant vulnerability when facing adversarial examples generated by FGSM and BIM, where carefully designed minute perturbations can severely destroy their recognition capabilities. Literature [37] utilized the gradient-based FGSM attack algorithm to generate adversarial examples, attacking a then state-of-the-art Graph Neural Network (GNN) model while ensuring the perturbations remained invisible, resulting in a recognition accuracy loss of over 80%. To comprehensively evaluate the security of SAR ship recognition models, Xu et al. [20] utilized a combination of gradient-based, boundary-based, and black-box adversarial attack methods to test mainstream CNNs. Their research, combined with visualization techniques, revealed the models' vulnerability to minute perturbations and proposed targeted robustness enhancement strategies based on these findings.

Existing adversarial attack methods mainly focus on maximizing classification error under strict perturbation amplitude constraints. Although these methods can deceive a specific substitute model, they often suffer from poor transferability [38,39]. A fundamental limitation is that these attacks tend to overfit the specific decision boundary of the substitute model. Due to the inherent discrepancies between the substitute model and the unknown black-box model (e.g., differences in architecture, training data, or preprocessing), perturbations that are highly effective on the former often fail to mislead the latter. This phenomenon is exacerbated in SAR images, where the coherent imaging mechanism and speckle noise make robust feature extraction distinct from optical photography, rendering simple gradient noise ineffective against robust black-box detectors.

To address the challenge of poor transferability in black-box SAR attacks, it is essential to generate perturbations that disrupt the "shared" vulnerable features across different models, rather than exploiting the idiosyncratic weaknesses of a single architecture. Recent theoretical findings suggest that minimizing the model discrepancy within a hypothesis

space can effectively bound the generalization error of adversarial attacks. Inspired by this, we propose that a transferable adversarial example should not only mislead a single deterministic model but should remain effective across a distribution of plausible models. In this paper, we propose a novel Bayesian Uncertainty Minimization framework for generating highly transferable adversarial examples against SAR ATR systems. Unlike traditional methods that rely on a static substitute model, we introduce a Bayesian approximation via MC Dropout to simulate an ensemble of diverse sub-models during the training of the adversarial generator. Specifically, we train a generator to craft perturbations that minimize the epistemic uncertainty (predictive variance) of the Bayesian surrogate model while maximizing the classification error. By forcing the generator to fool a stochastic distribution of models consistently, the resulting adversarial examples are prevented from overfitting to a specific decision boundary, thereby moving towards the manifold of shared vulnerabilities. This significantly enhances the attack success rate against unknown black-box models with diverse architectures (e.g., CNNs, Transformers).

The main contributions of this paper are summarized as follows:

1. We provide a theoretical analysis of adversarial transferability from a Bayesian perspective. Specifically, we derive a generalization error bound for the target model, mathematically demonstrating that minimizing the epistemic uncertainty of the surrogate model tightens this bound and guarantees improved transferability.
2. Guided by our theoretical derivation, we propose a novel generator-based attack framework. This framework incorporates MC Dropout to estimate uncertainty and introduces a specific loss function designed to minimize the derived uncertainty term, thereby forcing the generated perturbations to be robust against architectural variations.
3. We conduct extensive experiments on the MSTAR and FUSAR dataset. The results validate our theoretical findings and demonstrate that our method significantly outperforms state-of-the-art benchmarks in black-box scenarios, achieving superior transferability across diverse architectures including CNNs and Vision Transformers.

2. Related Work

2.1. SAR Target Recognition

SAR image interpretation primarily encompasses two core tasks: object detection and Automatic Target Recognition (ATR). The field has witnessed a paradigm shift from traditional handcrafted feature extraction [40,41] to data-driven deep learning methodologies. In the domain of object detection, early methods largely relied on statistical distributions [42], such as the Constant False Alarm Rate (CFAR) algorithm. However, with the rise of deep learning, detector architectures like Faster R-CNN and YOLO have been extensively adapted for SAR imagery to handle multi-scale targets and complex clutter. To facilitate these advancements, Zhang et al. released the SSDD (SAR Ship Detection Dataset) [43], which serves as a premier benchmark for maritime target detection. Furthermore, large-scale datasets such as AIR-SARShip and HRSID have subsequently propelled research in high-resolution ship detection. SAR ATR has evolved from template-based methods to data-driven deep learning approaches. Early works relied on manual feature extraction, such as geometric and scattering features. With the advent of deep learning, CNNs have become the dominant paradigm. Recently, Vision Transformers have also been applied to SAR interpretation, leveraging self-attention mechanisms to capture global dependencies. To facilitate research, several benchmark datasets have been released. The MSTAR dataset [44] is the standard for ground target recognition. For maritime applications, the SSDD (SAR Ship Detection Dataset) [43] and the large-scale FUSAR-Ship dataset [45] have enabled the training of more robust and complex models. In this work, we evaluate our method on both MSTAR and FUSAR to ensure broad applicability. However, while these models

achieve remarkable performance metrics on benchmarks like SSDD and MSTAR, their design prioritizes accuracy over security. The robustness of these detection and recognition systems against adversarial perturbations remains a critical vulnerability, which motivates our investigation into transferable attacks across these diverse architectures.

2.2. AI Security for Remote Sensing

The intersection of adversarial attacks and remote sensing is an emerging research hotspot. Unlike optical images, SAR images are sensitive to viewing angles and speckle noise, making them uniquely vulnerable. To enhance physical interpretability and realizability, recent studies have exploited the electromagnetic scattering mechanisms inherent to SAR imaging. Methods such as SMGAA [46], SCMA [47], and ASC-STA [48] leverage the Attributed Scattering Center (ASC) model to guide perturbation generation, specifically by manipulating robust backscattering structures or injecting adversarial scatterers. By targeting these physically significant features rather than adding generic pixel-wise noise, these approaches ensure that the attacks remain effective across different models while adhering to the unique imaging characteristics of SAR targets. Peng et al. [49] investigates the challenging Synthetic-to-Measured (S2M) setting. It demonstrates that attackers can generate effective perturbations using only synthetic data (SAMPLE dataset) that successfully transfer to victim models trained on real measured data, revealing severe security risks even when real-world training data is inaccessible.

While the aforementioned methods focus on exploiting domain-specific physics or data discrepancies to improve attack realism and transferability. In the optical imagery, domain, M3D [50] established a theoretical link between attack transferability and model discrepancy. M3D employs a generator trained against two distinct substitute models, optimizing the perturbation to minimize the maximum prediction divergence between them. This min-max game theoretically lowers the upper bound of the generalization error on the target model. Although M3D provides a solid theoretical foundation, it relies on a finite set of deterministic models (typically only two) to approximate the worst-case model discrepancy, which can be computationally expensive and may yield a loose bound for highly distinct architectures. Our BUM framework advances this theoretical lineage by adopting a Bayesian perspective. Instead of relying on a fixed pair of substitute models like M3D, BUM utilizes MC Dropout to simulate an infinite ensemble of stochastic submodels within a single architecture. This approach offers a tighter and more efficient approximation of the hypothesis space than M3D, enabling superior cross-architecture transferability without the need to train multiple disparate surrogate models.

3. Materials and Methods

In this section, we first formulate the problem of black-box adversarial attacks in the context of SAR target recognition. The key notations used in this section and their definitions are summarized in Table 1. Then, we revisit the generalization error bound for adversarial transferability, which serves as the theoretical foundation of our approach. Finally, we establish the connection between model discrepancy and Bayesian uncertainty, introducing our proposed BUM attack framework.

3.1. Problem Definition

We consider a black-box attack scenario on a SAR Automatic Target Recognition (ATR) system. The target model, denoted as $f_t : \mathcal{X} \to \mathcal{Y}$, is treated as an oracle with no knowledge of its parameters or architecture. The data domain consists of samples drawn from a distribution $\mathcal{D}$ over $\mathcal{X} \times \mathcal{Y}$, where each instance comprises an input image $\boldsymbol{x} \in \mathbb{R}^{C \times H \times W}$ and its corresponding ground-truth label y. The attacker's objective is to

craft an adversarial example $\boldsymbol{x}_{adv}$ using a substitute model f_s, such that the target model is misled, i.e., $f_t(\boldsymbol{x}_{adv}) \neq y$.

Table 1. List of symbols and their definitions.

Symbol	Definition
$\mathcal{D}$	Distribution of SAR images
$\mathcal{X}, \mathcal{Y}$	Input image space and label space
$\boldsymbol{x}, y$	Clean input image and its ground-truth label
$\boldsymbol{x}_{adv}$	Generated adversarial example
f_s, f_t	Surrogate model (white-box) and target model (black-box)
G_θ	Adversarial generator parameterized by θ
ϵ	Maximum perturbation budget (L_∞ norm constraint)
$\text{Clip}_{\boldsymbol{x},\epsilon}(\cdot)$	Projection function ensuring pixel values are within the ϵ-ball
Q	Posterior distribution of the surrogate model induced by MC Dropout
h	A stochastic sub-model sampled from distribution Q
$\bar{h}_Q$	Ensemble consensus (centroid) of the Bayesian surrogate
K	Number of stochastic forward passes (Monte Carlo samples)
$\boldsymbol{M}$	Binary dropout mask
$\boldsymbol{z}_k, \bar{\boldsymbol{z}}$	Logit vector of the k-th pass and the ensemble average logits
$\boldsymbol{p}_k, \bar{\boldsymbol{p}}$	Probability vector of the k-th pass and the ensemble average probabilities
$\mathcal{L}_{atk}$	Attack loss (margin-based logit loss)
$\mathcal{L}_{unc}$	Epistemic uncertainty loss (predictive variance)
$\lambda_{atk}, \lambda_{unc}$	Weighting hyperparameters for attack and uncertainty losses
κ	Confidence margin parameter in the attack loss
$\mathcal{R}(h)$	Expected risk (error) of hypothesis h
ξ	Irreducible domain shift between surrogate and target models

To ensure the stealthiness and physical feasibility of the attack, the perturbation is constrained by a norm budget ϵ, such that $\|\boldsymbol{x}_{adv} - \boldsymbol{x}\|_p \leq \epsilon$, where $p = \infty$ is commonly used. Instead of optimizing perturbations for individual images iteratively, we adopt a generative approach. We train a generator G_θ, parameterized by θ, which takes the clean image $\boldsymbol{x}$ as input and generates the adversarial perturbation. The adversarial example is formulated as:

$$\boldsymbol{x}_{adv} = \text{Clip}_{\boldsymbol{x},\epsilon}(\boldsymbol{x} + G_\theta(\boldsymbol{x})), \tag{1}$$

where $\text{Clip}_{\boldsymbol{x},\epsilon}(\cdot)$ projects the input onto the ℓ_∞-ball of radius ϵ around $\boldsymbol{x}$ and ensures valid pixel values. The core challenge is to optimize θ using only the information from f_s to maximize the attack success rate on the unseen f_t.

3.2. Theoretical Analysis: Generalization Bound via Bayesian Uncertainty

In this subsection, we provide a rigorous theoretical justification for the proposed BUM framework. We derive a generalization error bound for black-box transferability.

Let $\mathcal{D}$ be the distribution of input images. Let $\mathcal{H}$ be the hypothesis space of robust classifiers. The attacker aims to attack an unknown black-box model $h_b \in \mathcal{H}$. Instead of a single deterministic substitute model, we define the substitute as a posterior distribution Q over $\mathcal{H}$, induced by Monte Carlo Dropout. Let $h \sim Q$ represent a stochastic sub-model sampled from this distribution.

We define the expected error (risk) of a hypothesis h on the adversarial distribution as $\mathcal{R}(h) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\ell(h(\boldsymbol{x}_{adv}), f_{true}(\boldsymbol{x}))]$, where ℓ is a distance metric satisfying the triangle inequality (e.g., ℓ_1 norm on probability vectors). Our goal is to bound the risk on the unknown black-box model, $\mathcal{R}(h_b)$. We introduce the ensemble consensus of the Bayesian surrogate, denoted as $\bar{h}_Q(\boldsymbol{x}) = \mathbb{E}_{h\sim Q}[h(\boldsymbol{x})]$.

Using the triangle inequality of the metric ℓ, the error of the black-box model can be decomposed as:

$$\mathcal{R}(h_b) \leq \underbrace{\mathcal{R}(\bar{h}_Q)}_{\text{(I) Error of Consensus}} + \underbrace{\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\ell(\bar{h}_Q(\boldsymbol{x}_{adv}), h_b(\boldsymbol{x}_{adv}))]}_{\text{(II) Transferability Gap}}, \tag{2}$$

where Term (I) represents the attack failure rate of the ensemble average, and Term (II) represents the discrepancy between our Bayesian surrogate's consensus and the unknown black-box model.

3.2.1. Bounding Term (I): The Consensus Risk

To derive the upper bound for the target model's risk, we first establish the relationship between the risk of the ensemble consensus and the expected risk of individual stochastic sub-models. This relationship relies on the convexity of the loss function. First, we recall the definition of Jensen's Inequality, a fundamental theorem in convex analysis.

Definition 1 (Jensen's Inequality)**.** *Let $\phi : \mathbb{R}^d \to \mathbb{R}$ be a convex function, and let Z be a random variable taking values in the domain of ϕ. Then, the function of the expected value is less than or equal to the expected value of the function:*

$$\phi(\mathbb{E}[Z]) \leq \mathbb{E}[\phi(Z)]. \tag{3}$$

Equality holds if and only if Z is constant or ϕ is linear.

Based on this definition, we derive the following lemma for our Bayesian surrogate framework.

Lemma 1 (Convexity of Ensemble Risk)**.** *Let $\ell(\cdot, y)$ be a convex loss function (e.g., cross-entropy or margin-based loss) with respect to the model prediction logit vectors. Let Q be the posterior distribution of the hypothesis space induced by MC Dropout, and let $\bar{h}_Q = \mathbb{E}_{h\sim Q}[h]$ denote the ensemble consensus model. Then, the risk of the ensemble consensus is upper-bounded by the expected risk of the stochastic sub-models:*

$$\mathcal{R}(\bar{h}_Q) \leq \mathbb{E}_{h\sim Q}[\mathcal{R}(h)]. \tag{4}$$

Proof. We provide a step-by-step derivation. Recall that the risk function on the adversarial distribution is defined as the expectation of the loss over the input data:

$$\mathcal{R}(h) = \mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}[\ell(h(\boldsymbol{x}_{adv}), y)]. \tag{5}$$

Consider a fixed input sample $\boldsymbol{x}_{adv}$. By definition, the output of the ensemble consensus model $\bar{h}_Q$ is the expectation of the outputs of the sub-models sampled from Q:

$$\bar{h}_Q(\boldsymbol{x}_{adv}) = \mathbb{E}_{h\sim Q}[h(\boldsymbol{x}_{adv})]. \tag{6}$$

Since we assume the loss function $\ell(\boldsymbol{z}, y)$ is convex with respect to the prediction logits $\boldsymbol{z}$, we can apply Jensen's Inequality (Definition 1) by setting $\phi(\cdot) = \ell(\cdot, y)$ and the random variable $Z = h(\boldsymbol{x}_{adv})$:

$$\ell\big(\mathbb{E}_{h\sim Q}[h(\boldsymbol{x}_{adv})], y\big) \leq \mathbb{E}_{h\sim Q}[\ell(h(\boldsymbol{x}_{adv}), y)]. \tag{7}$$

Substituting the definition of the consensus model into the left-hand side of Equation (7), we get:

$$\ell(\bar{h}_Q(\boldsymbol{x}_{adv}), y) \leq \mathbb{E}_{h\sim Q}[\ell(h(\boldsymbol{x}_{adv}), y)]. \tag{8}$$

Since the inequality holds for any individual sample $\boldsymbol{x}_{adv}$, it preserves order under the expectation operator over the data distribution $\mathcal{D}$. We take the expectation $\mathbb{E}_{\boldsymbol{x}\sim\mathcal{D}}$ on both sides:

$$\mathbb{E}_{\boldsymbol{x}}\left[\ell(\bar{h}_Q(\boldsymbol{x}_{adv}), y)\right] \leq \mathbb{E}_{\boldsymbol{x}}\left[\mathbb{E}_{h\sim Q}[\ell(h(\boldsymbol{x}_{adv}), y)]\right]. \tag{9}$$

By Fubini's theorem (assuming integrability), we can swap the order of expectations on the right-hand side:

$$\mathcal{R}(\bar{h}_Q) \leq \mathbb{E}_{h\sim Q}[\mathbb{E}_{\boldsymbol{x}}[\ell(h(\boldsymbol{x}_{adv}), y)]]. \tag{10}$$

Finally, substituting the definition of risk $\mathcal{R}(h)$ back into the equation, we conclude the proof:

$$\mathcal{R}(\bar{h}_Q) \leq \mathbb{E}_{h\sim Q}[\mathcal{R}(h)]. \tag{11}$$

□

3.2.2. Bounding Term (II): The Transferability Gap

The second term, $\|\bar{h}_Q - h_b\|_1$, represents the distance between our surrogate centroid and the unseen black-box model. Since h_b is unknown, we cannot minimize this distance directly. We derive an upper bound through a three-step geometric relaxation.

We assume the black-box model h_b shares task-specific semantics with our hypothesis space $\mathcal{H}$. Let h^* be the ideal model within the support of our distribution Q that is closest to h_b:

$$h^* = \arg\min_{h\in \text{supp}(Q)} \|h - h_b\|_1. \tag{12}$$

Using the triangle inequality, we separate the distance into an optimizable term and an irreducible constant:

$$\|\bar{h}_Q - h_b\|_1 \leq \underbrace{\|\bar{h}_Q - h^*\|_1}_{\text{Distance to Ideal Proxy}} + \underbrace{\|h^* - h_b\|_1}_{\xi \text{ (Irreducible Domain Shift)}}. \tag{13}$$

Here, ξ represents the inherent architectural gap (e.g., CNN vs. Transformer) which cannot be optimized.

We now focus on $\|\bar{h}_Q - h^*\|_1$. Since h^* resides within the support of Q, the distance from the centroid $\bar{h}_Q$ to h^* is naturally bounded by the maximum radius of the distribution Q:

$$\|\bar{h}_Q - h^*\|_1 \leq \sup_{h\in \text{supp}(Q)} \|\bar{h}_Q - h\|_1. \tag{14}$$

Directly minimizing the supremum (worst-case radius) is intractable. By the norm equivalence property in finite probability spaces, for any non-negative random variable $Z_h = \|\bar{h}_Q - h\|_1$, its L_∞ norm (supremum) is bounded by its L_1 norm (expectation) scaled by a distribution-dependent constant C:

$$\sup_{h\in \text{supp}(Q)} \|\bar{h}_Q - h\|_1 \leq C \cdot \mathbb{E}_{h\sim Q}[\|\bar{h}_Q - h\|_1]. \tag{15}$$

The term $\mathbb{E}_{h\sim Q}[\|\bar{h}_Q - h\|_1]$ is precisely the Epistemic Uncertainty (predictive variance) of our Bayesian surrogate.

3.2.3. The Main Theorem

Combining Equation (4) and the derivation of Term (II), we arrive at our final generalization bound.

Theorem 1 (Uncertainty-Aware Transferability Bound)**.** *For a Bayesian surrogate defined by distribution Q, the generalization error on an unknown target model h_b is bounded by:*

$$\mathcal{R}(h_b) \leq \underbrace{\mathbb{E}_{h\sim Q}[\mathcal{R}(h)]}_{\mathcal{L}_{atk}:\text{Expected Surrogate Risk}} + \underbrace{C \cdot \mathbb{E}_{\boldsymbol{x}}\left[\mathbb{E}_{h\sim Q}\|h(\boldsymbol{x}_{adv}) - \bar{h}_Q(\boldsymbol{x}_{adv})\|_1\right]}_{\mathcal{L}_{unc}:\text{Epistemic Uncertainty}} + \xi. \tag{16}$$

Previous methods like M3D [50] attempt to approximate the supremum term $\sup \|h_1 - h_2\|$ using two deterministic models. This provides a loose bound because the "worst-case" pair is difficult to locate in high-dimensional spaces. In contrast, our BUM framework optimizes the expectation (the second term in Equation (16)). Geometrically, minimizing the average radius (variance) implicitly and tightly constrains the diameter of the hypothesis space. As $\mathcal{L}_{unc} \to 0$, the distribution Q collapses to a singularity, ensuring $\bar{h}_Q \approx h^*$, thereby maximizing the transferability probability.

3.3. Bayesian Uncertainty Minimization Attack Framework

Based on the theoretical analysis above, we propose the BUM attack framework. As illustrated in Figure 2 (placeholder for architecture diagram), the framework consists of a generator G_θ and a Bayesian surrogate model f_B. Unlike traditional attacks that rely on fixed gradients, we model the substitute network as a stochastic process to simulate the hypothesis space variations.

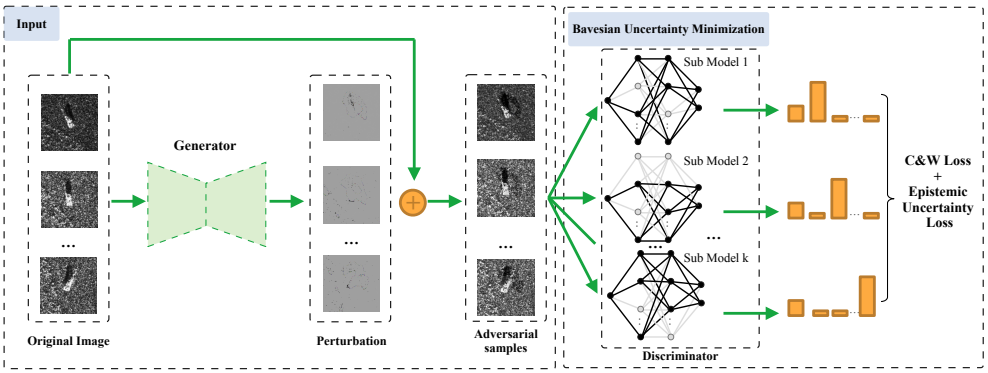


Figure 2. The overall framework of the proposed BUM attack. The method leverages MC Dropout within the source model to estimate epistemic uncertainty. The adversarial perturbation is optimized by jointly maximizing the classification loss and minimizing the predictive variance across stochastic forward passes, thereby generating transfer-robust adversarial examples. The black lines depict the connections of active neurons (not masked), while the gray lines represent dropped-out (masked) neurons during stochastic sampling with MC Dropout.

3.3.1. Stochastic Surrogate Modeling via MC Dropout

To implement the Bayesian approximation without the prohibitive computational cost of training an ensemble of distinct networks, we employ MC Dropout [51] during the inference phase of the surrogate model. Let the surrogate model f_s contain L layers, with parameters $\boldsymbol{W}$. We introduce a binary dropout mask $\boldsymbol{M}_l$ for each layer l, where each element is sampled from a Bernoulli distribution with probability $1 - p$. The predictive output for an input $\boldsymbol{x}$ is given by $f_s(\boldsymbol{x}; \boldsymbol{W} \odot \boldsymbol{M})$.

During the training of the generator, for every generated adversarial example $\boldsymbol{x}_{adv} = \text{Clip}_\epsilon(\boldsymbol{x} + G_\theta(\boldsymbol{x}))$, we perform K stochastic forward passes through the surro-

gate model. In each pass $k \in \{1, \ldots, K\}$, a different random dropout mask $\boldsymbol{M}^k$ is applied, yielding a set of probabilistic outputs:

$$\mathcal{P} = \{\boldsymbol{p}_1, \boldsymbol{p}_2, \ldots, \boldsymbol{p}_K\}, \tag{17}$$

where $\boldsymbol{p}_k = \text{Softmax}(f_s(\boldsymbol{x}_{adv}; \boldsymbol{W} \odot \boldsymbol{M}^k))$ represents the predicted probability distribution of the k-th stochastic sub-model. This ensemble $\mathcal{P}$ effectively approximates the posterior distribution of predictions within the hypothesis space $\mathcal{H}$.

Unlike static substitution models in traditional attacks, the surrogate model f_s in our framework functions similarly to a discriminator in a GAN. To prevent the generator from overfitting to a fixed uncertainty landscape, we dynamically update f_s during training. Specifically, f_s is optimized to minimize the classification loss $\mathcal{L}_{cls}$ on clean images while simultaneously maximizing the epistemic uncertainty $\mathcal{L}_{unc}$ on the generated adversarial examples. The optimization objective for the surrogate is formulated as minimizing $\mathcal{L}_D = \mathcal{L}_{cls} - \mathcal{L}_{unc}$. This adversarial game ensures that the generator learns to find robust vulnerabilities that persist even when the surrogate attempts to resist by increasing its uncertainty estimation.

3.3.2. Objective Functions

The optimization of the generator G_θ involves two primary objectives: ensuring the attack success (misclassification) and minimizing the predictive uncertainty (model discrepancy).

Untargeted Attack Loss. To encourage the adversarial example to mislead the model effectively, we adopt a margin-based logit loss, inspired by the Carlini–Wagner (C&W) attack [16]. Unlike the cross-entropy loss, which suffers from gradient saturation, the margin loss operates directly on the logits, providing stronger gradients for optimization. Let $\boldsymbol{z}_k \in \mathbb{R}^C$ denote the logits output of the k-th stochastic pass, and $\bar{\boldsymbol{z}} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{z}_k$ be the ensemble average logits. For an untargeted attack against the ground-truth label y, we aim to maximize the difference between the maximum logit of the incorrect classes and the logit of the true class. The loss is defined as:

$$\mathcal{L}_{atk} = \max\left(-\kappa, \bar{z}_y - \max_{i \neq y} \bar{z}_i\right), \tag{18}$$

where κ is a confidence parameter (set to 20 in our experiments) that guarantees a margin of separation. Minimizing $\mathcal{L}_{atk}$ pushes the true class score significantly below the most likely incorrect class, ensuring robust misclassification.

Epistemic Uncertainty Loss. To minimize the maximum model discrepancy, we enforce the generator to minimize the variance among the K stochastic predictions. We quantify this epistemic uncertainty using the mean L_1 distance between each individual prediction $\boldsymbol{p}_k$ and the ensemble consensus $\bar{\boldsymbol{p}} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{p}_k$:

$$\mathcal{L}_{unc} = \frac{1}{K}\sum_{k=1}^{K} \|\boldsymbol{p}_k - \bar{\boldsymbol{p}}\|_1. \tag{19}$$

By minimizing $\mathcal{L}_{unc}$, the generator is forced to find a "sweet spot" in the adversarial subspace where the majority of sub-models in the hypothesis space agree on the incorrect prediction. This significantly reduces the likelihood that the attack overfits to a specific architectural configuration.

Overall Optimization. The total loss function for training the generator is a weighted sum of the attack loss and the uncertainty loss:

$$\mathcal{L}_{total} = \lambda_{atk}\mathcal{L}_{atk} + \lambda_{unc}\mathcal{L}_{unc}, \tag{20}$$

where λ_{atk} and λ_{unc} are hyperparameters balancing the two objectives.

3.4. Training Algorithm

The training process is formulated as a min-max game. The surrogate model D attempts to maintain its classification accuracy on clean images while identifying adversarial patterns (maximizing uncertainty), whereas the generator G tries to fool the surrogate with low uncertainty. The complete training procedure is summarized in Algorithm 1.

Algorithm 1: BUM Attack

Input: Training dataset $\mathcal{X}$, Surrogate model f_s, Perturbation budget ϵ, MC samples K, Hyperparameters $\lambda_{atk}, \lambda_{unc}, \kappa$.

Output: Trained Generator G_θ.

Initialize generator G_θ and surrogate f_s;

while *not converged* **do**

  Sample a batch of clean images $(\boldsymbol{x}, y)$ from $\mathcal{X}$;

  `/* -------- Step 1: Update Surrogate (D) -------- */`

  Enable dropout in f_s;

  Generate adversarial examples: $\boldsymbol{x}_{adv} = \text{Clip}_\epsilon(\boldsymbol{x} + G_\theta(\boldsymbol{x}))$;

  Compute uncertainty $\mathcal{L}_{unc}$ on $\boldsymbol{x}_{adv}$ via K MC passes;

  Compute classification loss $\mathcal{L}_{cls}$ on clean $\boldsymbol{x}$;

  Update f_s to minimize $\mathcal{L}_{cls} - \mathcal{L}_{unc}$ (Maintain accuracy but maximize uncertainty on attacks);

  `/* -------- Step 2: Update Generator (G) -------- */`

  Generate adversarial examples: $\boldsymbol{x}_{adv} = \text{Clip}_\epsilon(\boldsymbol{x} + G_\theta(\boldsymbol{x}))$;

  Perform K stochastic forward passes to get logits $\{\boldsymbol{z}_k\}$ and probs $\{\boldsymbol{p}_k\}$;

  Compute average logits $\bar{\boldsymbol{z}}$ and average probs $\bar{\boldsymbol{p}}$;

  Compute Attack Loss: $\mathcal{L}_{atk} = \max(-\kappa, \bar{\boldsymbol{z}}_y - \max_{i \neq y} \bar{\boldsymbol{z}}_i)$;

  Compute Uncertainty Loss: $\mathcal{L}_{unc} = \frac{1}{K} \sum_{k=1}^{K} \|\boldsymbol{p}_k - \bar{\boldsymbol{p}}\|_1$;

  Total Loss: $\mathcal{L}_G = \lambda_{atk} \mathcal{L}_{atk} + \lambda_{unc} \mathcal{L}_{unc}$;

  Update G_θ to minimize $\mathcal{L}_G$;

end

3.5. Network Architectures

The architecture of the generator G_θ is designed to produce pixel-wise perturbations that match the resolution of the input SAR imagery. As shown in our implementation, G_θ adopts an encoder–decoder structure based on Residual Networks (ResNet) [52]. The generator network, comprising three downsampling blocks, six residual blocks, and two upsampling blocks in sequence, produces an adversarial example that preserves the spatial dimensions of the input image. The detailed architectural layout is illustrated in Figure 3.

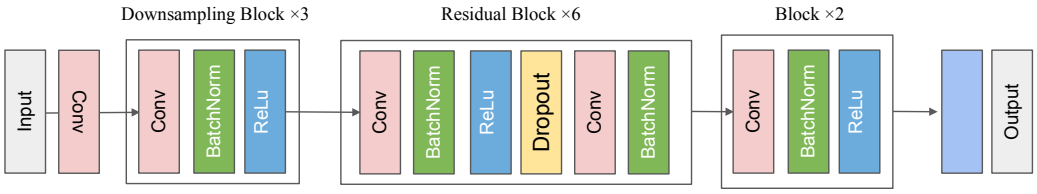


Figure 3. The network architecture of generator.

For the Bayesian surrogate model f_s, we utilize architectures capable of stochastic inference, such as VGG-19 or Vision Transformers (ViT). To enable the Bayesian approximation described in Section 3.3, we inject Dropout layers after the convolutional or attention

blocks. During the training of the generator, these Dropout layers are kept active (even in evaluation mode), serving as the source of epistemic uncertainty.

4. Results

4.1. Datasets

MSTAR dataset. This study leverages the publicly available MSTAR (Moving and Stationary Target Acquisition and Recognition) dataset [44], the predominant benchmark in SAR ATR research. The dataset was collected by Sandia National Laboratories under the joint sponsorship of the U.S. Defense Advanced Research Projects Agency and the Air Force Research Laboratory. It consists of X-band, HH-polarized, high-resolution spotlight SAR imagery with a 0.3 m × 0.3 m resolution. Following the standard SOC protocol, our experiments focus on ten types of military targets (BMP-2, BTR-70, BTR-60, T-72, T-62, 2S1, BRDM-2, D7, ZIL-131, ZSU-23-4). The training set employs images captured at a 17° depression angle, while the test set uses those at 15°, with both covering the full 0–360° azimuth range. The final dataset partition results in 2747 training and 2348 test samples, as summarized in Table 2 and Figure 4.

Table 2. Distribution of the MSTAR dataset.

Class Name	Train Set	Test Set	Total
BMP-2	233	196	429
BTR-70	233	196	429
T-72	232	196	428
BTR-60	256	195	451
2S1	299	196	495
BRDM-2	298	274	572
D7	299	274	573
T-62	299	273	572
ZIL-131	299	274	573
ZSU-234	299	274	573

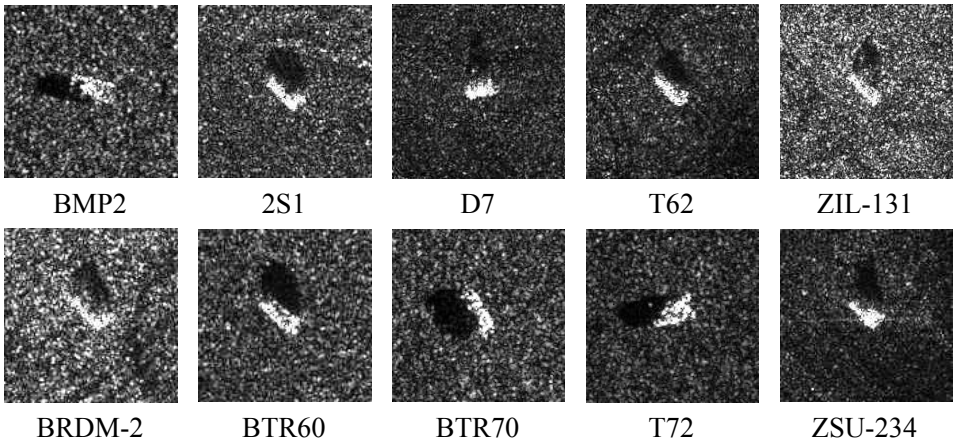


Figure 4. SAR images of the MSTAR dataset.

FUSAR-Ship dataset [45]. This study employs the FUSAR-Ship dataset, a high-resolution benchmark for SAR and AIS data fusion. Comprising over 10,000 image slices (500 × 500 pixels) acquired by China's GF-3 satellite (1m resolution), this collection serves as a resource for ship detection, recognition, and maritime surveillance research. The dataset encompasses diverse maritime environments, vessel types, and operational conditions, supplemented by comprehensive AIS records including vessel identity, position, course, speed, and dimensions. Example images from this dataset are shown in Figure 5. Example

images from this dataset are shown in Figure 5. The specific partitioning of data into training and test subsets for our experiments is provided in Table 3.

Table 3. Distribution of the FUSAR-Ship dataset.

Class Name	Train Set	Test Set	Total
Bulk carrier	191	82	273
Fishing	552	237	789
General cargo	28	13	41
Tanker	173	75	248

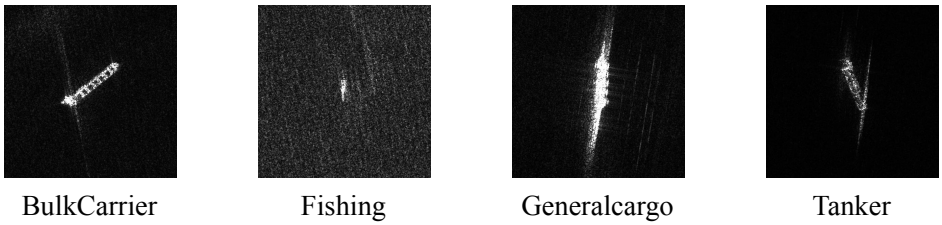


Figure 5. SAR images of the FUSAR dataset.

4.2. Experimental Setup

All experiments were conducted on a single NVIDIA RTX 4070 GPU (16 GB). Input images were standardized to a spatial dimension of 224 × 224 pixels. The model underwent a training regimen of 20 epochs using Stochastic Gradient Descent (SGD) as the optimization algorithm, with an initial learning rate of 0.001. To facilitate a direct comparison with the baseline methods implemented in this work, a uniform set of hyperparameters was adopted for all iterative adversarial attacks: a perturbation budget of $\epsilon = 0.06$ and an iteration count of $T = 10$. Regarding the hyperparameters of BUM, we set the loss weights in Equation (20) to $\lambda_{atk} = 1.0$ and $\lambda_{unc} = 1.0$ based on sensitivity analysis. The confidence margin κ in Equation (18) was set to 20, and the number of stochastic forward passes was set to $K = 4$. To ensure statistical reliability, particularly for generative methods (i.e., M3D and BUM) involving stochastic training processes, we conducted three independent runs with different random seeds. The results reported in this paper represent the mean Attack Success Rate (ASR) of these runs. For deterministic gradient-based attacks, standard evaluation results are reported.

4.3. Evaluation Metric

To quantify the effectiveness of adversarial attack algorithms, this paper employs the Attack Success Rate (ASR) as the core evaluation metric. Specifically, it represents the proportion of adversarial examples (with added perturbations) that cause the classifier to make incorrect predictions, calculated over the subset of original images that are correctly recognized by the model. A higher metric indicates a more potent attack algorithm and a greater vulnerability of the target model to that attack. The calculation formula is as follows:

$$\text{ASR} = \frac{\sum_{i=1}^{N} \mathbb{I}(f(x_i^{adv}) \neq y_i)}{N} \times 100\% \tag{21}$$

where N represents the total number of samples, $f(\cdot)$ is the classifier, x_i^{adv} is the i-th adversarial example, y_i is the ground truth label, and $\mathbb{I}(\cdot)$ is the indicator function. For targeted attacks, $f(x_i^{adv})$ must equal the preset target label y_{target}.

Kurtosis Difference (KurtDiff). Given that the background clutter in SAR images typically follows a heavy-tailed distribution, traditional L_p norm constraints often fail to guarantee the statistical stealthiness of adversarial perturbations. To quantify the degree to which the adversarial attack disrupts the statistical distribution of the original SAR image, we introduce the KurtDiff metric.

First, for a given image $\mathbf{x}$, the sample kurtosis $K(\mathbf{x})$ of its pixel distribution is defined as the ratio of the fourth central moment to the square of the variance:

$$K(\mathbf{x}) = \frac{\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^4}{\left(\frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})^2\right)^2} \tag{22}$$

where N is the total number of pixels in the image, and $\bar{x}$ is the pixel mean. Based on this, KurtDiff is defined as the absolute difference between the kurtosis of the adversarial image and that of the original image:

$$\text{KurtDiff} = |K(\mathbf{x}_{adv}) - K(\mathbf{x})| \tag{23}$$

Since the non-Gaussian nature of SAR images manifests as high kurtosis values, while gradient-based attacks tend to "flatten" the pixel distribution or generate abnormal extreme values, a smaller KurtDiff value (approaching 0) implies that the adversarial example better preserves the heavy-tailed statistical features of the original scene.

4.4. Performance Analysis on Clean Models

To establish a reliable baseline for subsequent adversarial attack evaluations, we comprehensively assessed the classification performance of nine mainstream deep learning models on two benchmark SAR datasets: MSTAR and FUSAR. The selected models cover a wide spectrum of architectures, ranging from classical CNNs (e.g., VGG [53], ResNet [52], RegNet [54]) to lightweight networks (e.g., MobileNetv2 [55], EfficientNet [56]) and emerging Vision Transformers (e.g., ViT [57], Swin Transformer [58], PiT [59]). Table 4 details the input resolution, parameter size, and classification accuracy for each model. Performance on MSTAR: As shown in Table 4, most models achieved a recognition accuracy exceeding 90% on the MSTAR dataset, demonstrating the strong capability of modern deep learning techniques in standard SAR target recognition. Notably, EfficientNet achieved the state-of-the-art (SOTA) classification performance with the highest accuracy of 98.56%. It is worth noting that EfficientNet accomplished this with only 5.29 M parameters, highlighting its superior efficiency in feature extraction compared to the heavy-weight VGG16, which achieved a comparable accuracy of 98.06% but requires a massive parameter count of 138.36 M. Performance on FUSAR and Generalization: However, the evaluation on the FUSAR dataset presents a more challenging scenario. As evidenced by the general decline in accuracy across all models, FUSAR appears to contain more complex background clutter or target variations than MSTAR. Despite this, EfficientNet demonstrated remarkable robustness, maintaining the top ranking with 86.00% accuracy. Similarly, classical CNNs like VGG16 and lightweight models like MobileNet remained competitive, achieving 84.77% and 84.52% respectively, indicating that inductive biases inherent in CNNs (such as translation invariance) offer better generalization in diverse SAR environments. A significant classification performance divergence was observed between CNNs and Transformers on the FUSAR dataset. While CNN-based models like ResNet-18 maintained a respectable accuracy of 82.56%, the Transformer family faced severe difficulties. Swin Transformer and PiT, which performed well on MSTAR (>90%), experienced drastic performance drops on FUSAR, falling to 71.74% and 72.24% respectively. Interestingly, the standard ViT showed a consistent but mediocre classification performance (approximately 82% on both datasets).

This suggests that without extensive pre-training or strong inductive biases, Transformer architectures may struggle to capture the robust features required for complex SAR ship detection tasks compared to their CNN counterparts. These models serve as robust baselines for this study. The classification performance gap observed between MSTAR and FUSAR, along with the varying resilience of different architectures, provides a comprehensive testbed for evaluating the transferability and universality of the proposed adversarial attack method in the following sections.

Table 4. Classification accuracy (%) comparison of different models.

Model	Resolution	Parameters (M)	MSTAR (%)	FUSAR (%)
VGG16	224 × 224	138.36	98.06	84.77
ResNet-18	224 × 224	11.69	95.01	82.56
ResNet-50	224 × 224	25.56	92.49	82.56
MobileNet	224 × 224	3.50	94.68	84.52
RegNet	224 × 224	2.68	91.71	84.03
EfficientNet	224 × 224	5.29	98.56	86.00
ViT	224 × 224	22.05	82.02	82.56
SwinTransformer	224 × 224	28.29	91.30	71.74
PiT	224 × 224	10.62	90.56	72.24

4.5. Black-Box Attack Transferability

We extensively evaluate the transferability of adversarial examples generated by our proposed BUM framework against a diverse set of target models, including CNN-based architectures (VGG16, ResNet-18/50, MobileNet, RegNet, EfficientNet) and Transformer-based architectures (ViT, Swin Transformer, PiT). The experimental results on the MSTAR and FUSAR datasets are presented in Table 5 and Table 6, respectively. In these tables, the rows represent the source models used to generate perturbations, while the columns represent the target black-box models (FGSM [11], BIM [33], MIM [35], PGD [34], NI-FGSM [60], VMI-FGSM [33], DIM [12], TIM [13], DeepFool [21], UAP [22], M3D [50]). The values indicate the ASR, where higher values signify stronger attack performance.

Table 5. Transferability Matrix (ASR%) on MSTAR. The last column shows the average black-box transferability.

Source	Attack	Target Model									Avg.
		VGG16	Res-18	Res-50	Mobile	RegNet	EffNet	ViT	Swin	PiT	
VGG16	FGSM	**88.4** *	36.9	34.8	65.2	56.0	67.6	80.5	59.4	72.4	**59.1**
	BIM	**94.9** *	30.0	35.6	69.4	28.2	69.5	80.4	65.3	72.9	**56.4**
	MIM	**94.3** *	28.9	35.5	68.0	33.7	70.5	80.5	63.9	73.3	**56.8**
	PGD	**94.6** *	29.4	28.1	62.2	60.3	58.6	77.5	62.4	68.8	**55.9**
	NI-FGSM	**94.3** *	28.9	35.7	68.2	33.7	70.4	80.5	63.9	73.4	**56.8**
	VMI-FGSM	**92.6** *	30.7	37.0	68.0	43.2	69.2	77.8	61.0	71.2	**57.3**
	DIM	**93.7** *	32.7	36.0	66.7	37.2	66.3	80.6	65.1	74.2	**57.4**
	TIM	**94.3** *	30.0	36.2	68.5	31.8	71.0	81.0	64.3	74.3	**57.1**
	DeepFool	**100.0** *	0.7	1.1	7.4	0.9	4.3	24.4	18.1	20.9	**9.7**
	UAP	**77.4** *	48.1	44.9	74.0	76.3	71.0	87.5	74.4	85.2	**70.2**
	M3D	**85.1** *	74.8	74.9	74.9	80.1	77.1	92.4	91.7	90.9	82.3
	BUM (Ours)	**85.8** *	75.4	75.6	75.0	81.4	77.1	92.4	91.7	91.5	**82.5**

Table 5. *Cont.*

Source	Attack	Target Model									Avg.
		VGG16	Res-18	Res-50	Mobile	RegNet	EffNet	ViT	Swin	PiT	
Res-18	FGSM	0.2	**92.6** *	88.4	29.5	87.8	21.6	14.2	4.6	14.0	**32.5**
	BIM	0.0	**99.9** *	90.5	23.2	86.6	12.5	5.2	2.4	5.1	**28.2**
	MIM	0.0	**99.9** *	89.1	35.8	86.7	22.7	14.8	5.3	15.9	**33.8**
	PGD	0.0	**100.0** *	90.1	11.1	87.6	6.0	7.0	2.7	6.6	**26.4**
	NI-FGSM	0.0	**99.9** *	89.1	35.6	86.6	22.9	14.6	5.5	15.6	**33.7**
	VMI-FGSM	0.2	**88.0** *	65.3	26.6	80.2	19.5	11.2	4.5	14.4	**27.7**
	DIM	0.2	**99.5** *	94.0	41.5	85.6	30.8	12.5	4.6	14.4	**35.5**
	TIM	5.1	**99.8** *	92.1	71.2	87.0	55.6	62.7	37.3	57.8	**58.6**
	DeepFool	0.0	**99.9** *	5.0	0.2	2.6	0.1	0.1	0.0	0.1	**1.0**
	UAP	0.4	**88.1** *	87.8	5.2	87.7	4.9	11.7	2.7	9.2	**26.2**
	M3D	4.0	**91.5** *	91.3	77.4	90.8	41.8	47.5	33.6	45.9	54.0
	BUM (Ours)	3.9	**89.6** *	88.8	66.8	77.7	58.8	45.3	41.6	53.7	**57.5**
Res-50	FGSM	0.2	89.1	**91.4** *	21.2	87.7	13.2	14.2	4.4	13.3	**30.4**
	BIM	0.1	98.0	**98.6** *	20.0	84.2	8.7	4.8	2.0	5.1	**27.9**
	MIM	0.2	98.2	**98.6** *	31.5	84.6	17.9	12.4	4.3	12.5	**32.7**
	PGD	0.1	97.7	**99.8** *	7.8	87.2	5.5	7.4	2.8	7.3	**27.0**
	NI-FGSM	0.2	98.2	**98.6** *	31.8	84.7	18.1	12.6	4.5	12.6	**32.8**
	VMI-FGSM	0.2	70.4	**91.2** *	19.5	81.3	9.6	15.4	4.7	16.9	**27.2**
	DIM	0.2	94.9	**99.2** *	26.3	86.8	16.4	9.9	3.3	9.1	**30.9**
	TIM	5.1	89.5	**98.5** *	66.3	84.0	48.9	57.4	31.8	52.2	**54.4**
	DeepFool	0.0	0.0	**99.5** *	0.0	0.4	0.0	0.0	0.0	0.0	**0.0**
	UAP	0.2	88.1	**87.8** *	3.9	87.7	3.7	10.9	2.9	7.8	**25.7**
	M3D	2.0	91.5	**91.3** *	82.5	90.9	77.4	44.6	23.9	39.7	56.6
	BUM (Ours)	3.9	89.7	**89.9** *	74.5	85.7	74.4	60.7	37.3	54.9	**60.2**
Mobile	FGSM	2.9	81.8	84.8	**95.0** *	87.7	29.6	35.8	17.4	28.8	**46.1**
	BIM	0.7	46.1	49.2	**100.0** *	72.5	4.9	17.9	5.0	14.1	**26.3**
	MIM	2.3	77.8	82.1	**100.0** *	86.9	20.4	33.9	13.5	28.5	**43.2**
	PGD	0.6	57.9	62.9	**100.0** *	84.1	4.0	17.8	5.2	13.8	**30.8**
	NI-FGSM	2.6	78.0	81.9	**100.0** *	86.7	20.4	34.4	13.8	28.7	**43.3**
	VMI-FGSM	1.4	77.7	81.8	**99.9** *	86.6	12.4	24.5	10.4	20.2	**39.4**
	DIM	5.0	76.3	79.2	**100.0** *	86.5	38.5	50.6	25.3	43.1	**50.6**
	TIM	16.3	30.5	24.7	**100.0** *	29.8	64.6	65.6	45.9	58.1	**41.9**
	DeepFool	0.0	0.0	0.1	**100.0** *	0.2	0.0	0.3	0.1	0.1	**0.1**
	UAP	2.1	55.9	61.8	**56.9** *	80.8	34.1	24.0	22.3	23.4	**38.1**
	M3D	7.2	78.0	74.7	**86.6** *	47.9	87.8	58.6	43.7	56.1	56.8
	BUM (Ours)	3.1	81.6	81.2	**86.9** *	70.8	88.1	66.3	46.7	61.1	**62.4**
RegNet	FGSM	0.1	86.9	87.0	7.0	**87.6** *	3.3	13.2	3.2	12.4	**26.7**
	BIM	0.1	80.2	84.3	2.5	**99.6** *	1.7	5.2	2.1	4.6	**22.6**
	MIM	0.0	86.7	87.4	4.8	**99.4** *	2.8	9.4	2.7	8.6	**25.3**
	PGD	0.1	81.8	84.4	3.4	**100.0** *	1.7	6.7	2.5	5.9	**23.3**
	NI-FGSM	0.0	86.7	87.4	4.9	**99.4** *	2.9	9.4	2.8	8.5	**25.3**
	VMI-FGSM	0.0	63.3	71.9	6.3	**72.8** *	2.2	8.4	2.1	10.2	**20.6**
	DIM	0.1	86.2	86.4	4.1	**95.6** *	2.5	8.0	2.6	7.7	**24.7**
	TIM	4.1	57.3	58.6	53.0	**100.0** *	36.3	48.9	25.3	44.1	**40.9**
	DeepFool	0.0	0.0	0.0	0.0	**100.0** *	0.0	0.0	0.0	0.0	**0.0**
	UAP	0.1	87.5	87.7	4.4	**87.7** *	2.4	8.5	3.5	9.6	**25.5**
	M3D	1.5	91.5	91.2	82.1	**91.2** *	74.0	40.7	21.6	36.9	54.9
	BUM (Ours)	1.7	84.7	85.4	78.2	**92.0** *	74.2	45.8	26.1	48.1	**55.5**

Table 5. *Cont.*

Source	Attack	Target Model									Avg.
		VGG16	Res-18	Res-50	Mobile	RegNet	EffNet	ViT	Swin	PiT	
EffNet	FGSM	3.5	81.1	84.2	52.6	87.5	**96.8** *	38.7	19.7	31.2	**49.8**
	BIM	1.1	71.0	72.7	23.7	84.5	**99.1** *	16.9	5.4	11.8	**35.9**
	MIM	1.8	75.6	80.0	39.4	86.9	**99.0** *	26.7	12.7	23.0	**43.3**
	PGD	0.9	70.5	71.9	21.2	85.7	**99.0** *	15.1	5.2	13.4	**35.5**
	NI-FGSM	1.8	75.6	80.0	39.4	86.9	**99.0** *	26.8	12.6	23.0	**43.3**
	VMI-FGSM	1.4	75.0	80.0	25.8	86.9	**94.5** *	23.7	8.0	17.3	**39.7**
	DIM	6.7	72.0	77.4	64.9	85.9	**98.8** *	52.8	30.6	41.9	**54.0**
	TIM	40.3	45.1	38.6	83.0	42.9	**98.8** *	76.6	67.0	72.4	**58.2**
	DeepFool	0.2	0.2	0.3	1.1	0.7	**100.0** *	2.6	0.7	1.2	**0.9**
	UAP	2.0	73.7	79.7	37.8	86.1	**83.8** *	24.4	15.0	24.3	**42.9**
	M3D	2.3	72.1	71.1	47.0	60.0	**76.7** *	42.4	27.9	40.2	45.4
	BUM (Ours)	59.7	78.7	73.2	82.0	76.2	**84.9** *	89.7	81.6	84.9	**78.2**
ViT	FGSM	15.1	78.7	81.6	36.6	86.7	23.6	**93.2** *	47.0	68.7	**54.8**
	BIM	5.3	64.2	71.4	26.5	85.5	16.9	**99.9** *	33.1	61.0	**45.5**
	MIM	11.7	74.5	79.2	38.7	86.1	26.5	**99.7** *	48.1	72.0	**54.6**
	PGD	4.3	65.2	72.1	25.3	85.3	13.8	**99.5** *	29.8	58.2	**44.2**
	NI-FGSM	11.7	74.4	79.2	38.8	86.1	26.4	**99.7** *	48.1	72.0	**54.6**
	VMI-FGSM	10.8	74.1	79.2	38.4	86.1	24.6	**99.6** *	47.7	71.0	**54.0**
	DIM	18.0	73.5	70.8	46.1	82.5	35.8	**99.5** *	57.6	74.9	**57.4**
	TIM	51.0	49.2	52.1	72.2	59.9	69.6	**98.9** *	82.7	92.2	**66.1**
	DeepFool	0.1	0.0	0.1	0.3	0.2	0.1	**98.0** *	0.8	1.5	**0.4**
	UAP	27.5	53.9	60.7	32.1	80.2	18.9	**97.3** *	46.3	71.9	**49.0**
	M3D	58.2	76.8	77.4	64.6	85.6	80.3	**86.2** *	80.5	85.1	76.1
	BUM (Ours)	35.3	77.8	75.6	74.3	81.0	73.4	**79.7** *	74.3	77.9	**71.2**
Swin	FGSM	37.5	62.2	60.1	64.3	84.6	49.5	74.5	**98.6** *	66.4	**62.4**
	BIM	22.2	30.8	33.2	34.6	69.2	16.0	69.0	**100.0** *	65.0	**42.5**
	MIM	35.2	40.5	41.7	52.0	79.2	34.4	75.5	**100.0** *	75.4	**54.2**
	PGD	19.3	39.3	39.9	30.2	82.1	12.8	65.8	**100.0** *	62.5	**44.0**
	NI-FGSM	35.2	40.5	41.8	52.0	79.3	34.4	75.5	**100.0** *	75.4	**54.2**
	VMI-FGSM	33.4	41.5	42.0	51.7	79.7	34.2	74.9	**100.0** *	74.2	**54.0**
	DIM	45.5	42.8	43.2	60.4	72.8	43.9	78.3	**100.0** *	78.8	**58.2**
	TIM	61.6	29.1	32.4	70.2	33.7	62.7	84.3	**100.0** *	86.3	**57.5**
	DeepFool	0.5	0.2	0.2	1.0	0.2	0.7	13.4	**97.2** *	6.7	**2.9**
	UAP	19.0	58.1	59.7	39.3	84.7	16.2	82.5	**87.1** *	50.3	**51.2**
	M3D	28.3	73.9	70.8	80.1	85.1	54.7	75.2	**69.6** *	74.7	67.8
	BUM (Ours)	68.9	77.9	79.2	76.2	87.1	81.2	92.4	**89.2** *	88.5	**81.4**
PiT	FGSM	24.9	69.7	70.2	49.1	83.5	29.5	71.6	53.9	**92.3** *	**56.5**
	BIM	11.5	55.4	54.4	24.6	78.5	15.9	79.8	36.4	**100.0** *	**44.6**
	MIM	28.8	70.0	70.6	46.0	83.0	28.5	84.0	59.4	**100.0** *	**58.8**
	PGD	8.7	63.1	65.0	23.0	82.7	12.7	75.6	32.4	**100.0** *	**45.4**
	NI-FGSM	28.8	70.0	70.6	45.9	83.0	28.6	84.0	59.4	**100.0** *	**58.8**
	VMI-FGSM	27.6	69.8	70.0	43.2	82.7	27.5	81.9	57.3	**100.0** *	**57.5**
	DIM	34.3	70.0	66.4	53.2	79.2	35.6	84.1	64.8	**99.9** *	**60.9**
	TIM	51.3	41.6	40.0	69.6	51.3	59.2	90.5	80.3	**99.6** *	**60.5**
	DeepFool	0.2	0.1	0.1	0.7	0.6	0.3	5.2	1.2	**98.8** *	**1.1**
	UAP	10.1	76.6	76.5	43.3	86.9	16.4	54.1	30.1	**84.5** *	**49.3**
	M3D	38.3	72.0	70.8	58.4	74.8	71.4	81.0	68.5	**82.6** *	66.9
	BUM (Ours)	75.4	70.3	71.9	76.3	78.9	68.6	95.1	84.2	**92.3** *	**77.6**

* Diagonal and bold elements indicate white-box attacks. The 'Avg.' column represents the average black-box ASR. Underlined values indicate the best ASR in each column.

Table 6. Transferability Matrix (ASR%) on FUSAR. The last column shows the average black-box transferability.

Source	Attack	Target Model									Avg.
		VGG16	Res-18	Res-50	Mobile	RegNet	EffNet	ViT	Swin	PiT	
VGG16	FGSM	**39.4** *	70.5	38.4	39.5	36.8	40.6	41.7	30.8	26.5	**40.6**
	BIM	**77.7** *	51.2	44.4	47.4	38.9	49.1	53.9	43.1	44.9	**46.6**
	MIM	**75.6** *	83.0	50.9	54.9	47.4	57.1	59.5	63.0	45.6	**57.7**
	PGD	**61.7** *	30.7	36.0	33.4	30.4	38.6	42.0	27.7	29.3	**33.5**
	NI-FGSM	**76.5** *	83.6	51.2	54.6	46.8	57.4	59.8	63.0	44.9	**57.7**
	VMI-FGSM	**55.6** *	63.4	30.1	33.4	23.7	36.0	34.8	29.1	27.2	**34.7**
	DIM	**73.9** *	84.5	50.6	56.4	53.5	59.4	59.5	63.0	51.0	**59.8**
	TIM	**74.8** *	82.7	49.1	49.7	57.0	55.4	59.5	63.0	46.3	**57.9**
	DeepFool	**100.0** *	7.4	12.2	12.8	10.8	6.9	11.3	9.2	13.6	**10.5**
	UAP	**28.7** *	69.6	32.1	34.9	34.2	29.4	28.0	48.3	7.5	**35.5**
	M3D	**38.8** *	59.3	45.3	45.0	55.1	54.5	62.0	55.6	55.1	54.0
	BUM (Ours)	**60.3** *	81.5	62.4	64.6	69.2	75.0	78.9	81.0	77.6	**73.8**
Res-18	FGSM	19.1	**88.1** *	46.7	24.4	36.5	31.4	33.6	21.9	21.1	**29.4**
	BIM	19.1	**100.0** *	94.0	23.5	36.3	19.4	25.3	21.9	21.1	**32.6**
	MIM	19.1	**100.0** *	88.4	24.1	36.5	25.4	32.1	21.9	21.1	**33.6**
	PGD	19.1	**100.0** *	84.5	23.3	36.5	21.7	29.2	21.9	21.1	**32.2**
	NI-FGSM	19.1	**100.0** *	88.7	24.7	36.5	26.0	30.9	21.9	21.1	**33.6**
	VMI-FGSM	19.1	**98.5** *	55.4	21.8	36.3	24.3	27.1	21.9	21.1	**28.4**
	DIM	19.1	**100.0** *	79.5	26.2	36.5	27.7	30.1	21.9	21.1	**32.8**
	TIM	22.6	**100.0** *	58.3	37.8	49.7	48.3	43.8	36.0	29.9	**40.8**
	DeepFool	0.0	**99.7** *	6.9	1.2	0.9	0.0	0.0	0.0	0.0	**1.1**
	UAP	19.1	**82.1** *	83.6	24.7	36.5	30.9	36.3	21.9	21.1	**34.3**
	M3D	21.5	**57.9** *	30.7	23.6	29.1	30.4	38.1	21.9	21.1	27.0
	BUM (Ours)	57.2	**90.7** *	71.7	60.3	52.3	79.6	82.8	85.0	83.8	**71.6**
Res-50	FGSM	19.1	37.5	**97.3** *	21.5	36.5	32.0	35.7	21.9	21.1	**28.2**
	BIM	18.3	75.9	**100.0** *	12.2	36.5	16.3	24.4	21.9	21.1	**28.3**
	MIM	19.1	70.2	**100.0** *	9.9	36.5	26.3	28.9	21.9	21.1	**29.2**
	PGD	19.1	65.2	**100.0** *	15.4	36.5	21.1	25.9	21.9	21.1	**28.3**
	NI-FGSM	19.1	69.3	**100.0** *	8.7	36.5	26.3	30.1	21.9	21.1	**29.1**
	VMI-FGSM	19.1	40.2	**81.5** *	22.1	36.0	28.3	25.6	21.9	21.1	**26.8**
	DIM	19.7	53.3	**100.0** *	18.9	36.5	29.7	35.4	21.9	21.1	**29.6**
	TIM	21.2	74.4	**98.2** *	29.1	36.0	34.9	36.3	13.0	19.4	**33.0**
	DeepFool	0.0	1.5	**99.4** *	0.3	0.6	0.0	0.0	0.0	0.0	**0.3**
	UAP	19.1	36.6	**33.9** *	12.8	36.5	25.1	33.9	21.9	21.1	**25.9**
	M3D	20.1	32.9	**60.9** *	23.1	43.9	22.8	38.1	21.9	21.1	28.0
	BUM (Ours)	60.5	89.0	**80.3** *	67.9	73.8	81.5	87.1	91.2	83.8	**79.4**
Mobile	FGSM	19.7	36.6	33.6	**37.2** *	36.5	39.4	36.3	21.9	21.1	**30.7**
	BIM	16.2	7.4	29.5	**100.0** *	30.4	12.0	20.5	21.9	21.1	**19.9**
	MIM	19.1	34.5	31.6	**100.0** *	35.4	37.7	32.7	21.9	21.1	**29.3**
	PGD	19.4	22.9	31.2	**100.0** *	33.6	19.1	25.0	21.9	21.1	**24.3**
	NI-FGSM	19.1	35.7	31.9	**100.0** *	35.7	38.0	29.8	21.9	21.1	**29.1**
	VMI-FGSM	19.1	34.8	31.6	**59.6** *	36.3	37.4	31.9	21.9	21.1	**29.3**
	DIM	19.4	33.6	29.8	**93.0** *	36.0	38.0	31.9	21.9	21.1	**29.0**
	TIM	21.2	59.5	32.4	**99.7** *	30.4	38.0	38.1	21.9	21.1	**32.8**
	DeepFool	0.0	0.0	0.0	**100.0** *	0.0	0.0	0.0	0.0	0.0	**0.0**
	UAP	19.1	17.9	19.9	**37.5** *	36.0	20.0	24.1	21.9	21.1	**22.5**
	M3D	26.7	39.2	49.3	**27.2** *	43.4	33.1	41.6	36.3	35.8	38.2
	BUM (Ours)	49.5	76.1	62.2	**64.4** *	62.3	72.7	74.4	77.4	75.4	**68.7**

Table 6. *Cont.*

Source	Attack	Target Model									Avg.
		VGG16	Res-18	Res-50	Mobile	RegNet	EffNet	ViT	Swin	PiT	
RegNet	FGSM	19.1	35.4	33.9	22.4	**39.2** *	25.4	34.2	21.9	21.1	**26.7**
	BIM	13.3	23.2	31.9	16.3	**100.0** *	8.3	21.4	21.9	21.1	**19.7**
	MIM	19.1	31.9	33.6	20.1	**100.0** *	17.4	26.5	21.9	21.1	**23.9**
	PGD	19.1	27.1	33.6	19.2	**100.0** *	16.3	26.2	21.9	21.1	**23.1**
	NI-FGSM	19.1	32.1	33.6	22.4	**100.0** *	18.0	26.8	21.9	21.1	**24.4**
	VMI-FGSM	19.1	25.3	32.1	20.3	**90.9** *	21.7	25.3	21.9	21.1	**23.4**
	DIM	19.1	30.9	33.9	20.3	**92.4** *	19.7	28.0	21.9	21.1	**24.4**
	TIM	19.1	33.0	44.6	24.1	**100.0** *	18.3	35.4	20.5	20.8	**27.0**
	DeepFool	0.0	0.0	0.0	0.3	**100.0** *	0.0	0.0	0.0	0.0	**0.0**
	UAP	19.1	30.6	32.1	13.7	**81.6** *	24.0	24.4	21.9	21.1	**23.4**
	M3D	21.8	26.5	27.5	23.4	**47.6** *	22.9	38.1	21.9	21.1	25.4
	BUM (Ours)	28.4	52.3	50.7	27.5	**68.9** *	35.3	41.8	41.8	41.8	**40.0**
EffNet	FGSM	19.4	36.9	30.9	23.0	35.7	**95.4** *	33.3	21.9	21.1	**27.8**
	BIM	16.2	11.9	28.9	13.7	30.7	**99.7** *	18.8	21.6	21.1	**20.3**
	MIM	19.1	33.9	32.1	23.8	36.3	**100.0** *	28.0	21.9	21.1	**27.0**
	PGD	19.4	23.5	31.5	18.6	34.2	**100.0** *	24.7	21.9	21.1	**24.4**
	NI-FGSM	19.1	34.8	32.1	24.1	36.3	**99.7** *	28.0	21.9	21.1	**27.2**
	VMI-FGSM	19.1	29.5	29.8	25.9	35.7	**71.1** *	32.7	21.9	21.1	**27.0**
	DIM	19.1	36.6	34.8	27.9	35.7	**99.4** *	30.9	21.9	21.1	**28.5**
	TIM	20.3	70.2	35.7	30.5	30.7	**99.7** *	37.2	24.0	21.4	**33.8**
	DeepFool	0.0	0.0	0.0	0.9	0.0	**100.0** *	0.0	0.0	0.0	**0.1**
	UAP	19.1	20.8	28.6	32.0	36.5	**79.1** *	24.1	21.9	21.1	**25.5**
	M3D	22.2	23.4	25.0	23.7	38.6	**21.8** *	38.1	21.9	21.1	26.8
	BUM (Ours)	32.4	51.1	44.4	31.0	44.6	**64.2** *	47.5	49.1	47.8	**43.5**
ViT	FGSM	27.0	57.7	45.2	34.9	41.2	51.1	**56.9** *	43.8	43.2	**43.0**
	BIM	28.4	44.6	43.8	32.0	43.9	38.9	**98.5** *	44.9	46.6	**40.4**
	MIM	29.6	51.5	47.0	38.7	45.0	46.9	**98.2** *	45.6	47.6	**44.0**
	PGD	22.9	34.5	39.3	28.2	39.8	26.0	**98.5** *	28.4	31.0	**31.3**
	NI-FGSM	29.6	51.5	47.0	38.1	45.0	46.9	**98.2** *	45.6	47.6	**43.9**
	VMI-FGSM	26.4	51.5	45.5	36.9	42.1	48.9	**96.1** *	45.9	49.0	**43.3**
	DIM	32.2	64.0	50.0	45.1	57.9	54.3	**90.2** *	57.5	53.4	**51.8**
	TIM	33.6	49.4	47.0	42.1	49.4	58.6	**94.9** *	48.6	47.6	**47.1**
	DeepFool	0.6	3.6	6.2	0.6	10.2	0.6	**99.1** *	1.4	9.2	**4.0**
	UAP	37.4	75.3	57.1	39.5	59.1	64.0	**74.4** *	71.2	70.1	**59.2**
	M3D	46.1	77.4	58.3	54.9	64.6	71.4	**75.0** *	72.6	72.8	64.8
	BUM (Ours)	51.3	78.3	60.5	59.4	66.8	73.3	**76.9** *	77.8	77.0	**68.1**
Swin	FGSM	33.3	72.6	44.9	38.7	35.1	43.1	69.9	**75.3** *	66.7	**50.5**
	BIM	43.5	90.8	70.8	37.5	64.9	73.4	88.4	**92.1** *	89.1	**69.8**
	MIM	42.3	78.3	61.6	50.0	58.2	65.4	85.4	**90.1** *	85.0	**65.8**
	PGD	42.3	86.0	66.4	36.9	56.7	66.6	85.4	**90.4** *	86.4	**65.8**
	NI-FGSM	42.3	78.3	61.6	50.0	58.2	65.4	85.4	**90.1** *	85.0	**65.8**
	VMI-FGSM	20.9	83.9	52.4	38.4	37.7	54.3	73.8	**89.0** *	72.8	**54.3**
	DIM	45.5	78.3	59.5	54.6	71.9	66.9	89.0	**92.1** *	88.1	**69.2**
	TIM	56.2	78.6	65.8	60.8	66.4	68.6	94.3	**94.5** *	93.9	**73.1**
	DeepFool	6.1	66.4	30.4	8.7	14.6	14.0	27.4	**67.1** *	50.7	**27.3**
	UAP	19.1	35.4	33.3	9.0	27.8	15.1	31.6	**21.9** *	21.1	**24.1**
	M3D	22.3	24.4	25.0	23.3	39.5	23.1	38.1	**21.9** *	21.1	27.1
	BUM (Ours)	52.3	78.4	61.7	59.5	67.3	73.7	77.1	**77.9** *	77.4	**68.4**

Table 6. *Cont.*

Source	Attack	Target Model									Avg.
		VGG16	Res-18	Res-50	Mobile	RegNet	EffNet	ViT	Swin	PiT	
PiT	FGSM	48.4	75.3	54.5	51.4	60.5	65.4	85.7	87.0	**87.8** *	**66.0**
	BIM	43.2	72.3	60.1	50.3	49.7	58.0	76.5	79.5	**92.5** *	**61.2**
	MIM	47.2	74.1	61.6	49.7	67.2	58.3	80.1	84.9	**93.2** *	**65.4**
	PGD	44.1	72.0	62.2	44.8	64.9	58.9	78.3	81.2	**92.2** *	**63.3**
	NI-FGSM	47.2	74.1	61.6	50.3	67.2	58.3	80.1	84.9	**93.2** *	**65.5**
	VMI-FGSM	45.8	73.8	61.6	48.3	63.4	57.1	77.4	83.2	**92.2** *	**63.8**
	DIM	44.6	74.1	50.9	48.8	52.9	57.1	81.8	86.6	**93.2** *	**62.1**
	TIM	50.1	75.9	57.1	55.8	60.5	64.6	86.0	89.7	**93.9** *	**67.5**
	DeepFool	22.0	61.0	33.3	23.3	40.1	32.0	40.5	59.2	**86.4** *	**38.9**
	UAP	19.4	32.4	33.9	21.5	36.5	25.1	38.1	21.9	**21.1** *	**28.6**
	M3D	20.3	24.6	23.8	23.3	39.4	20.9	38.1	21.9	**21.1** *	26.5
	BUM (Ours)	55.9	83.4	64.7	63.1	71.6	77.1	82.1	82.8	**81.4** *	**72.6**

* Diagonal and bold elements indicate white-box attacks. The 'Avg.' column represents the average black-box ASR. Underlined values indicate the best ASR performance in each column.

Superiority over SOTA Methods. Quantitative results on the MSTAR dataset (Table 5) demonstrate that our proposed BUM method consistently achieves the highest average black-box ASR across all source model configurations, establishing a new state-of-the-art for transferability. For instance, when EfficientNet is utilized as the source model, BUM attains an impressive average black-box ASR of 78.2%, significantly outperforming the second-best method, TIM (58.2%), and the generative baseline, M3D (45.4%). The improvement is even more pronounced when compared to classic gradient-based iterative attacks such as MIM (43.3%) and PGD (35.5%), where BUM nearly doubles the transferability. A key observation is the overfitting phenomenon prevalent in traditional methods. While attacks like PGD and BIM achieve near 100% ASR in white-box settings (diagonal elements), their performance collapses drastically when transferred to unknown models (e.g., PGD drops from 100% on ResNet-18 to merely 2.7% on Swin Transformer). In stark contrast, BUM maintains a exceptionally narrow gap between white-box and black-box performance. For example, with VGG16 as the source, BUM achieves an average black-box ASR of 82.5%, which is remarkably close to its white-box ASR of 85.8%, proving that minimizing Bayesian uncertainty effectively regularizes the perturbation, preventing it from overfitting to the source model's specific decision boundary artifacts.

Cross-Architecture Transferability. A significant highlight of BUM is its robust ability to bridge the generalization gap between CNNs and ViTs, a notoriously difficult challenge in adversarial transferability. As illustrated in Table 5, when adversarial examples are crafted on a CNN-based source (VGG16), BUM achieves a striking ASR of 92.4% against ViT and 91.7% against Swin Transformer. This cross-architecture efficacy is bidirectional. When using a Transformer-based source (PiT) to attack CNN targets, BUM continues to dominate, achieving 95.1% on ViT and 78.9% on RegNet, whereas M3D only reaches 81.0% and 74.8%, and TIM struggles at 90.5% and 51.3%. These results strongly suggest that the perturbations optimized via Bayesian Uncertainty Minimization capture high-level, structure-aware semantic features that are shared across both convolutional and attention mechanisms, rather than relying on high-frequency noise patterns that are specific to the local receptive fields of CNNs.

The results on the FUSAR dataset (Table 6) further validate the generalizability and robustness of our proposed approach. FUSAR presents a significantly more challenging scenario than MSTAR due to its complex sea surface backgrounds and diverse target variations.

Dominance in Black-box Transferability. Despite the increased difficulty, BUM consistently dominates the transferability benchmarks across nearly all source models. (1). VGG16

as Source: When using VGG16 as the surrogate, BUM achieves a remarkable average black-box ASR of 73.8%, which is double the ASR of standard baselines like FGSM (40.6%) and PGD (33.5%), and significantly outperforms the strongest competitor, DIM (59.8%). (2). ResNet-18 as Source: With ResNet-18 as the source, BUM maintains a commanding lead with an average ASR of 71.6%. It is worth noting that while TIM shows competitive ASR (40.8%), it still lags behind BUM by a large margin of over 20 percentage points. (3). Challenging Transfers: Even in the notoriously difficult transfer scenario from ResNet-18 to MobileNet, BUM maintains a robust ASR of 60.3%, whereas traditional iterative attacks like PGD and BIM fail catastrophically, dropping to 23.3% and 23.5%, respectively. We also observe that the choice of the surrogate model architecture influences attack ASR, yet BUM consistently amplifies the transferability regardless of the architecture. (1). Transformer-based Surrogates: Consistent with recent findings that self-attention mechanisms promote shape-biased representations, Transformer-based surrogates generally yield higher transferability. For instance, using PiT as the source, BUM achieves an exceptional average ASR of 72.6%, surpassing TIM (67.5%) and DIM (62.1%). (2). CNN-based Surrogates: While CNNs typically generate more texture-biased perturbations that are harder to transfer, BUM effectively mitigates this limitation. On ResNet-50, BUM elevates the average ASR to 79.4%, a dramatic improvement over the baseline PGD (28.3%) and MIM (29.2%). This consistent superiority across different data distributions and model architectures confirms that modeling the epistemic uncertainty of the surrogate model is a fundamental and effective strategy for enhancing adversarial transferability in SAR ATR tasks.

To comprehensively evaluate the stealthiness of the generated adversarial examples, specifically in the context of SAR imagery, we employ the KurtDiff metric. Unlike optical images, SAR images are characterized by a heavy-tailed backscattering distribution. A low KurtDiff indicates that the adversarial perturbations preserve these intrinsic statistical properties, making them harder to detect by statistical defense mechanisms. Table 7 presents the quantitative results on the MSTAR dataset.

Table 7. Kurtosis Difference comparison on MSTAR. Lower values indicate better statistical stealthiness.

Attack Method	VGG16	Res-18	Res-50	Mobile	RegNet	EffNet	ViT	Swin	PiT
FGSM	7.04	6.67	6.89	7.13	7.09	7.17	6.81	7.15	6.06
BIM	5.39	3.97	2.66	1.70	3.64	4.27	3.71	2.54	1.91
MIM	6.29	5.24	4.49	5.06	5.75	5.69	5.33	4.61	4.23
PGD	5.01	5.09	4.47	3.37	4.38	4.79	4.79	3.31	3.30
NI-FGSM	6.29	5.24	4.49	5.05	5.75	5.69	5.33	4.61	4.23
VMI-FGSM	6.30	3.68	2.94	5.16	5.30	5.69	5.46	4.69	4.41
DIM	6.28	4.88	4.90	5.43	5.66	5.90	5.17	4.87	4.44
TIM	6.16	3.21	2.85	5.04	5.12	5.64	4.51	4.59	3.42
DeepFool	0.18	0.02	0.01	0.00	0.00	0.03	0.03	0.04	0.02
UAP	5.94	6.88	6.91	4.71	6.69	7.29	5.93	6.60	4.57
M3D	6.21	1.29	1.48	3.60	1.10	7.25	4.74	2.31	5.47
BUM (Ours)	5.91	5.34	5.72	2.40	5.49	3.75	2.84	3.39	3.49

As shown in Table 7, DeepFool consistently achieves the lowest KurtDiff values (e.g., 0.01 on ResNet-50 and 0.03 on ViT). This is expected, as DeepFool optimizes for the minimal Euclidean distance to the decision boundary, resulting in infinitesimal perturbations. However, referring back to the ASR results (Table 5), DeepFool exhibits near-zero black-box transferability. This implies that while DeepFool is statistically invisible, it creates "fragile" examples that overfit the source model. Conversely, single-step attacks like FGSM and universal attacks like UAP yield the highest KurtDiff values (often exceeding 6.0 or 7.0). These

methods introduce significant statistical anomalies, disrupting the heavy-tailed distribution of SAR clutter, which renders them easily detectable despite their computational efficiency.

Our proposed BUM method demonstrates a robust performance in maintaining statistical consistency. BUM achieves a remarkably low KurtDiff of 2.84, significantly outperforming traditional iterative methods like PGD (4.79) and MIM (5.33). This suggests that the uncertainty-minimized perturbations generated by BUM align well with the global semantic structure of SAR images, rather than introducing high-frequency statistical noise. Across CNN models (e.g., MobileNet, EffNet), BUM maintains KurtDiff values (e.g., 2.40 on MobileNet) that are comparable to or lower than standard iterative baselines (PGD, BIM), while avoiding the statistical degradation seen in FGSM.

The Stealthiness-Transferability Trade-off. A critical insight emerges when combining the results from Table 5 (ASR) and Table 7 (KurtDiff). Ideally, an adversarial attack should maximize ASR while minimizing KurtDiff. M3D shows instability; while it achieves a low KurtDiff on ResNet-18 (1.29), it suffers from high statistical distortion on VGG16 (6.21) and EffNet (7.25). BUM strikes a superior balance. Although BUM does not always yield the absolute lowest KurtDiff compared to the non-transferable DeepFool, it achieves state-of-the-art transferability (as detailed in the ASR analysis) while maintaining acceptable statistical consistency. For instance, on the ResNet-50 source model, BUM achieves a massive improvement in transferability compared to PGD, with only a marginal increase in KurtDiff (5.72 vs. 4.46). This indicates that BUM prioritizes perturbing statistically significant features (scattering centers) that are critical for model generalization, rather than adding imperceptible but useless noise.

In summary, BUM generates adversarial examples that are not only highly transferable across different architectures but also statistically deceptive. It successfully navigates the trade-off between aggressiveness and stealthiness, avoiding the statistical anomalies common in simple attacks (FGSM) while delivering significantly stronger attack capability than minimal-perturbation methods (DeepFool).

The statistical consistency analysis on the FUSAR dataset (Table 8) corroborates the findings from MSTAR, though the absolute KurtDiff values are notably higher across all methods due to the complex, non-Gaussian sea surface clutter inherent to FUSAR.

Table 8. Kurtosis Difference comparison on FUSAR. Lower values indicate better statistical stealthiness.

Attack Method	VGG16	Res-18	Res-50	Mobile	RegNet	EffNet	ViT	Swin	PiT
FGSM	138.34	162.55	163.05	162.89	162.86	162.92	117.18	74.12	47.60
BIM	121.45	128.63	121.13	92.17	110.83	97.13	91.95	89.40	96.05
MIM	140.84	151.60	151.50	150.13	149.73	149.31	122.72	78.03	59.14
PGD	133.96	136.85	133.77	121.90	130.00	126.08	123.01	86.02	85.77
NI-FGSM	140.81	151.60	151.51	150.06	149.71	149.31	122.71	78.03	59.14
VMI-FGSM	107.40	141.10	140.31	143.46	140.54	141.85	121.40	125.52	67.90
DIM	139.77	149.22	150.27	150.37	150.81	150.14	107.84	44.27	51.31
TIM	142.94	146.68	149.32	149.58	147.44	145.63	95.21	58.32	46.27
DeepFool	35.02	0.62	0.36	0.00	0.02	0.29	5.07	36.66	28.03
UAP	149.29	159.28	159.65	137.53	157.06	145.61	59.21	161.81	150.48
M3D	76.90	97.48	54.87	127.87	57.62	48.50	103.32	110.91	39.97
BUM (Ours)	61.62	52.29	43.54	44.70	51.84	50.74	73.43	57.05	61.14

Despite this increased environmental complexity, BUM maintains superior statistical stealthiness among transfer-based attacks. (1). Comparison with Baselines: When using ResNet-50 as the source model, BUM achieves a KurtDiff of 43.54, which is significantly lower than the generative baseline M3D (54.87) and standard iterative methods like PGD (133.77) and MIM (151.50). (2). Consistency: While DeepFool continues to show the lowest

statistical deviation (e.g., 0.36 on ResNet-50), its inability to transfer limits its practical utility. Conversely, BUM successfully suppresses the high-frequency statistical anomalies typical of FGSM and UAP (which often exceed 140), ensuring that the generated perturbations remain statistically indistinguishable from natural sea clutter while delivering the high transferability reported in Table 6.

To provide qualitative insights into the attack mechanisms, we visualized the adversarial examples and their corresponding perturbations on both the MSTAR and FUSAR datasets. Figures 6–9 illustrate the original images, the adversarial examples generated by various methods, and the magnified perturbation patterns.

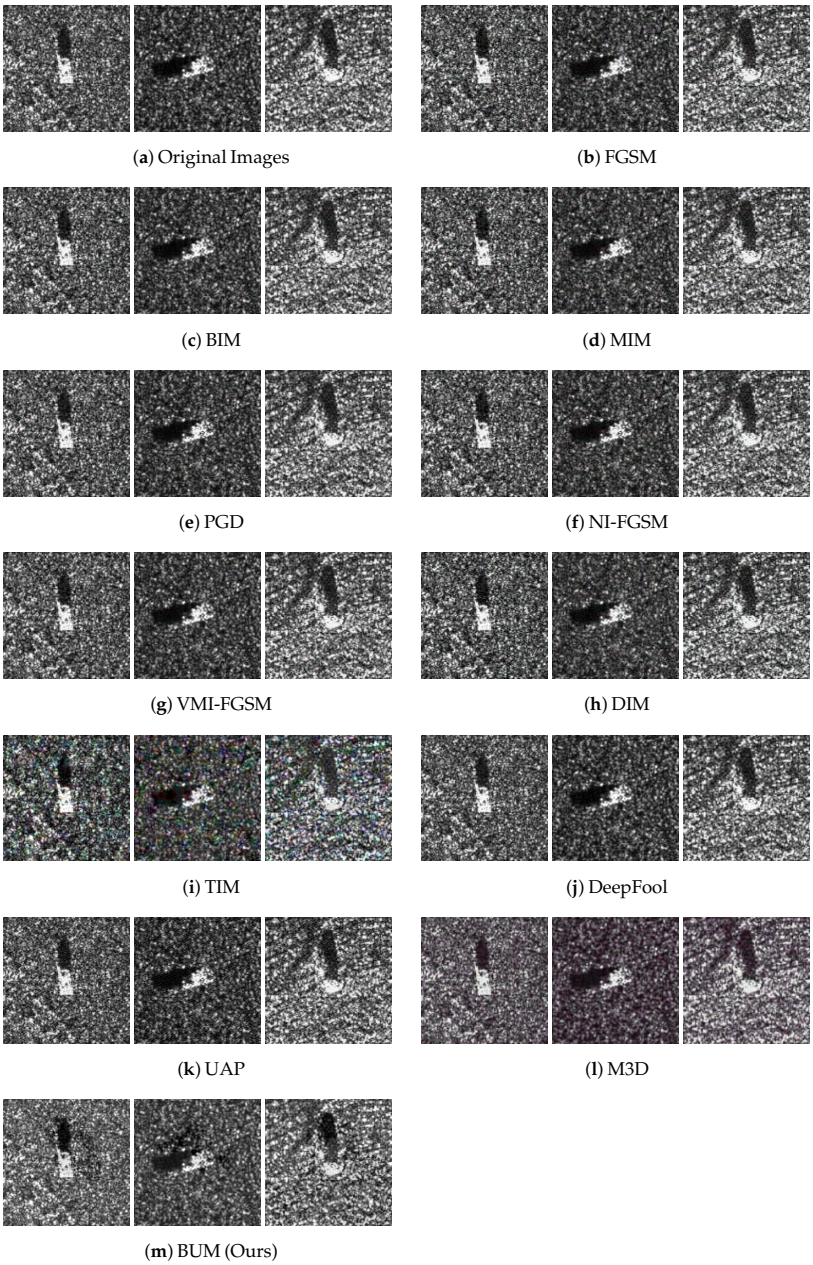


Figure 6. Visual comparison of adversarial examples on the MSTAR dataset. The figure displays: (**a**) Original images; (**b**–**m**) Adversarial examples generated by different attack methods.

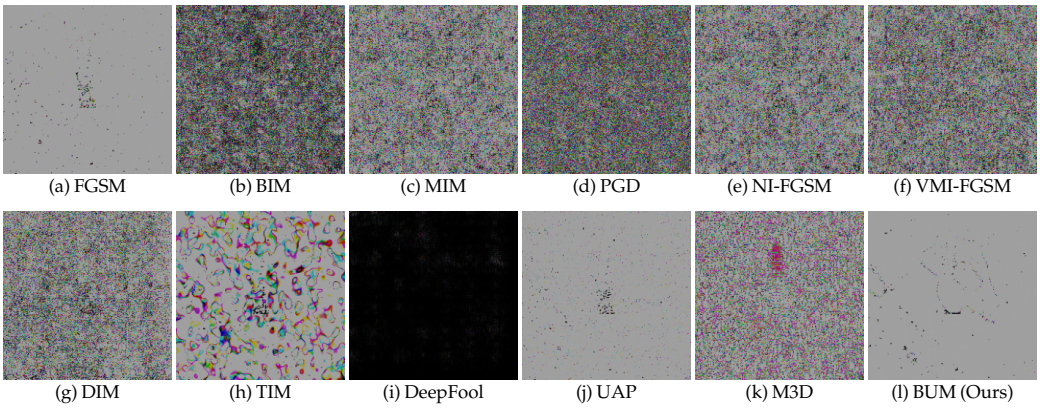


Figure 7. Visualization of perturbations generated by different attacks on the MSTAR dataset. The original images and adversarial examples are displayed in the first column of Figure 6. All perturbations are magnified (10×) for better visibility.

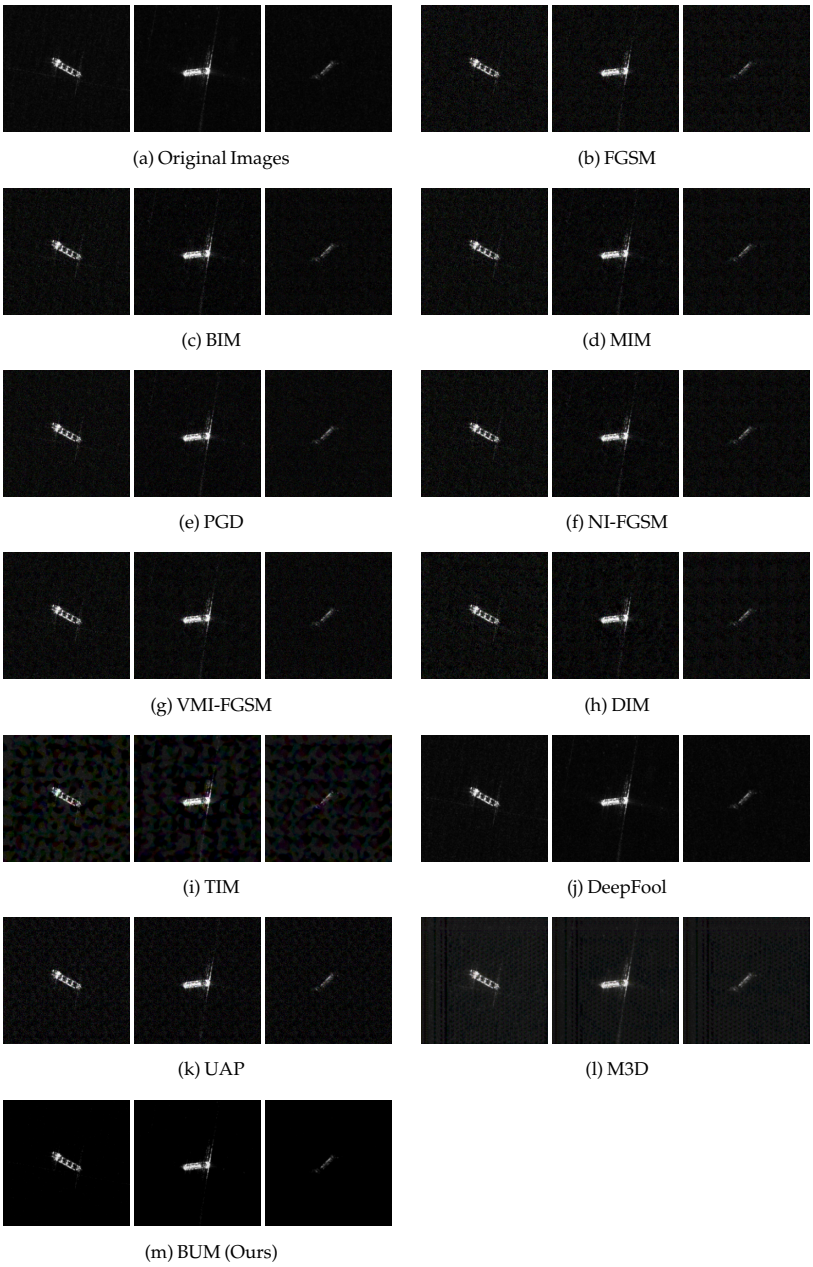


Figure 8. Visual comparison of adversarial examples on the FUSAR dataset.

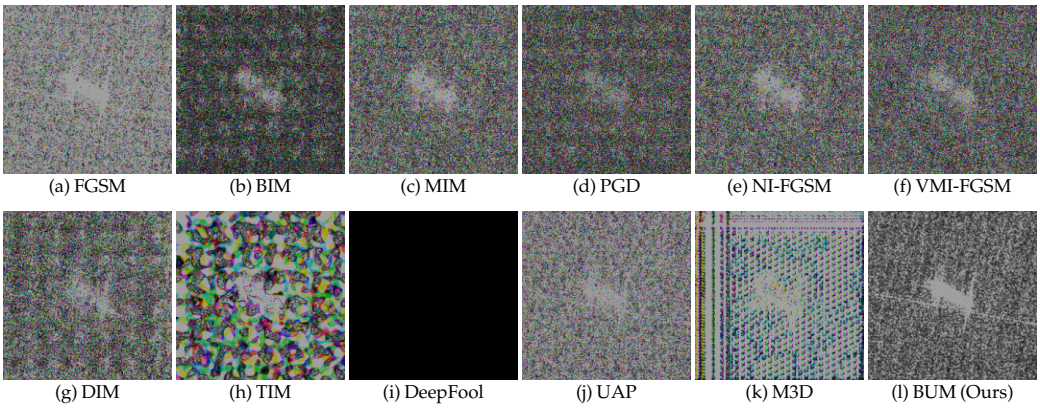


Figure 9. Visualization of perturbations generated by different attacks on the FUSAR dataset. The original images and adversarial examples are displayed in the first column of Figure 8. All perturbations are magnified (10×) for better visibility.

Visual Imperceptibility on MSTAR. Figure 6 displays the adversarial examples generated on the MSTAR dataset. Standard gradient-based attacks like FGSM and VMI-FGSM introduce noticeable "salt-and-pepper" noise across the entire image, disrupting the background clutter significantly. The M3D method overlays a fixed, grid-like pattern, which is visually distinct from the natural SAR speckle noise. TIM tends to generate perturbations with blocky artifacts (likely due to the generator's upsampling layers), which can degrade image quality. In contrast, the adversarial examples generated by BUM retain the high-fidelity structural details of the target vehicle while maintaining the natural appearance of the background speckle. The perturbations are seamlessly integrated into the scattering centers of the target, making the attack visually indistinguishable from natural SAR imaging variations. The results on the FUSAR dataset (Figure 8m) further corroborate these findings.

Analysis of Perturbation Patterns. The magnified perturbation maps in Figures 7 and 9 reveal the fundamental differences in how each method constructs an attack: (1). Iterative methods like PGD and DIM generate noise that is uniformly distributed across the image (both target and background). This indicates that these attacks rely on overfitting to high-frequency background noise, which explains their poor transferability to other models. (2). DeepFool: Although DeepFool generates the faintest perturbations (almost invisible in Figure 7i), its lack of structural focus results in negligible transferability, as discussed in before. (3). BUM's Semantic Focus: The perturbation map of BUM (Figures 7 and 9l) is distinctively sparse and structure-aware. Unlike the chaotic noise of PGD, BUM's perturbations are concentrated around the target's key scattering centers and edges. By minimizing Bayesian uncertainty, BUM identifies and perturbs the most "semantically sensitive" pixels—features that are shared across different DNN architectures. This localized, feature-centric perturbation strategy ensures that the adversarial example remains stealthy even in simple backgrounds where global noise would be easily spotted by human observers. This explains why BUM achieves SOTA transferability: it attacks the object structure rather than the model-specific decision boundary.

Visualizations confirm that BUM does not simply add random noise; instead, it intelligently modifies the semantic features of the SAR targets. This structure-aware perturbation strategy not only ensures high visual quality (stealthiness) but also guarantees that the attack features remain effective when transferred to unknown black-box models.

4.6. Ablation Study

4.6.1. Impact of Loss Components

To verify the necessity of each component in our BUM framework, we conducted an ablation study by training generators with different loss configurations. Table 9 summarizes the average black-box ASR on both datasets. When training with only $\mathcal{L}_{unc}$ (i.e., without explicit adversarial guidance), the ASR drops significantly (e.g., ResNet-18 on FUSAR drops from 74.45% to 36.76%). This indicates that uncertainty minimization alone acts as a regularizer; it cannot effectively drive the input across the decision boundary without the directional guidance provided by the attack loss. When removing $\mathcal{L}_{unc}$ (equivalent to the baseline M3D), the method suffers from overfitting. By re-introducing uncertainty minimization, the transferability improves consistently across all models. This improvement is particularly pronounced for ResNet-18 on FUSAR, where the ASR increases by nearly 9% (65.57% $\rightarrow$ 74.45%). The combination of both losses achieves the best average black-box ASR performance. This confirms our hypothesis: $\mathcal{L}_{atk}$ ensures the perturbation is aggressive enough to fool the source model, while $\mathcal{L}_{unc}$ forces the perturbation into low-uncertainty regions, ensuring these adversarial features are robust and transferable to unknown target architectures.

Table 9. Ablation study analyzing the individual and combined contributions of Attack Loss ($\mathcal{L}_{atk}$) and Epistemic Uncertainty Minimization ($\mathcal{L}_{unc}$) to the average black-box ASR.

Source Model	Components		Avg. Black-Box ASR (%)	
	Attack Loss	Uncertainty	MSTAR	FUSAR
VGG16	✓	×	82.96	78.78
	×	✓	49.99	67.26
	✓	✓	83.16	81.39
ResNet-18	✓	×	57.82	65.57
	×	✓	44.94	36.76
	✓	✓	64.30	74.45
PiT	✓	×	78.52	78.38
	×	✓	60.07	36.15
	✓	✓	81.58	80.90

Note: ✓ indicates the component is included, while × indicates it is removed.

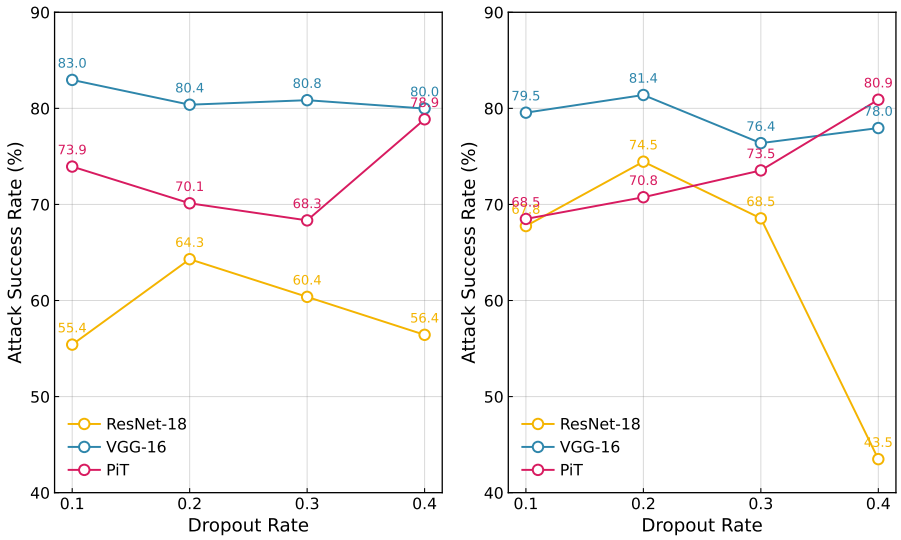
Impact of Loss Weights (λ_{atk} and λ_{unc}). The hyperparameters λ_{atk} and λ_{unc} in Equation (20) balance the trade-off between attack aggressiveness and generalization capability. To investigate their impact, we fixed $\lambda_{atk} = 1.0$ and varied the uncertainty weight $\lambda_{unc} \in \{0.5, 1.0, 1.5\}$ on the MSTAR dataset. The results are summarized in Table 10. For CNN-based surrogates (VGG16 and ResNet-18), we observe a distinct "inverted U-shape" trend. The average black-box ASR performance peaks at $\lambda_{unc} = 1.0$. Specifically, for ResNet-18, reducing λ_{unc} to 0.5 leads to a sharp drop in ASR (from 59.8% to 38.5%), indicating that insufficient uncertainty regularization causes the generator to overfit the surrogate's decision boundary. Conversely, increasing λ_{unc} to 1.5 also degrades average black-box ASR performance (43.6%), as the optimization may overly prioritize variance reduction at the expense of the attack success objective. Interestingly, the Transformer-based model (PiT) exhibits a slightly different behavior, where average black-box ASR performance marginally improves as λ_{unc} increases (reaching 71.2% at $\lambda_{unc} = 1.5$). This suggests that Transformers, with their global attention mechanisms, may benefit from stronger regularization constraints. However, considering the trade-off across different architectures and the significant stability observed in CNNs, we adopted $\lambda_{atk} = 1.0$ and $\lambda_{unc} = 1.0$ as the default setting for all experiments to ensure general applicability.

Table 10. Sensitivity analysis of the hyperparameter λ_{unc} in the total loss function ($\mathcal{L}_{total} = \lambda_{atk}\mathcal{L}_{atk} + \lambda_{unc}\mathcal{L}_{unc}$). We fixed $\lambda_{atk} = 1$ and varied λ_{unc}. The results report the Average Black-box ASR (%).

Source Model	λ_{atk}	λ_{unc}	Avg. Black-Box ASR (%)
VGG16	1.0	0.5	81.2
	1.0	1.0	83.1
	1.0	1.5	81.0
ResNet-18	1.0	0.5	38.5
	1.0	1.0	59.8
	1.0	1.5	43.6
PiT	1.0	0.5	69.1
	1.0	1.0	70.1
	1.0	1.5	71.2

4.6.2. Sensitivity Analysis of Dropout Rate

The dropout rate p serves as a critical hyperparameter in our BUM framework, controlling the stochastic intensity of the Monte Carlo Dropout mechanism. It determines the diversity of the sampled sub-models and, consequently, the estimation of epistemic uncertainty. To investigate its influence, we evaluated the transferability of adversarial examples generated with varying dropout rates (e.g., ranging from 0.1 to 0.4) across two datasets. The results are summarized in Figure 10.



(**a**) (**b**)

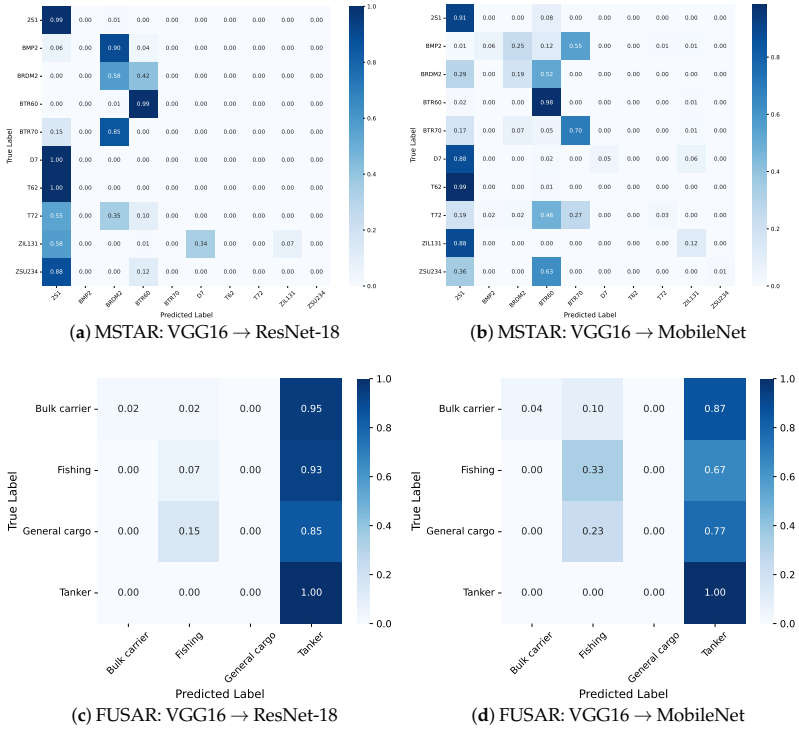
Figure 10. Sensitivity analysis of the dropout rate p on adversarial transferability. The curves illustrate the Average Black-box ASR (%) of adversarial examples generated by BUM using different source models (VGG16, ResNet-18, PiT). (**a**) ASR performance variations on the MSTAR. (**b**) ASR performance variations on the FUSAR.

As shown in the results, a moderate dropout rate typically yields the highest transferability. For instance, on the FUSAR dataset, ResNet-18 achieves its peak ASR of 74.45% at the second sampling point, while VGG16 also peaks at 81.39%. This confirms that introducing an appropriate level of stochasticity helps the generator escape the local overfitting of the source model. However, excessive dropout leads to a sharp decline in ASR performance. A notable example is ResNet-18 on FUSAR, where the ASR drops significantly to 43.49% at the highest dropout rate tested. This suggests that when too many neurons are deactivated, the semantic features extracted by the source model become severely fragmented, resulting in uninformative gradients that fail to guide the generator effectively.

Interestingly, VGG16 demonstrates higher resilience to dropout rate variations on the MSTAR dataset, maintaining a stable ASR around 80% across all settings. This may be attributed to the dense parameterization of VGG, which makes it robust to structural pruning. The behavior of the Vision Transformer (PiT) diverges from that of CNNs, reflecting the architectural differences in feature processing. PiT often requires or tolerates higher dropout rates to maximize transferability. This phenomenon aligns with the property of self-attention mechanisms, which possess a global receptive field and are inherently more robust to local occlusions (simulated by dropout) than the local convolution operations in CNNs. Therefore, a stronger perturbation in the model structure (higher p) is needed to effectively estimate uncertainty boundaries for Transformers. In summary, the choice of dropout rate involves a trade-off between ensemble diversity and gradient validity. Too little dropout fails to regularize the attack (overfitting), while too much dropout destroys the gradient signal (underfitting). Based on these empirical results, we empirically set the dropout rate to 0.1 for CNNs and 0.4 for Transformers in our main experiments to ensure optimal transferability.

4.6.3. Confusion Matrix

To address the variability in attack ASR performance across different categories, we visualize the confusion matrices of the target models (ResNet-18 and MobileNet) when attacked by adversarial examples generated from a VGG16 source model (Figure 11). A striking phenomenon of "Pseudo-targeted" bias is observed in the black-box transfer scenarios. Although our BUM attack is theoretically untargeted (aiming only to minimize the probability of the ground-truth class), the generated perturbations do not mislead the target model randomly. Instead, they exhibit a strong preference to shift predictions towards specific "sink" categories.



(**a**) MSTAR: VGG16 → ResNet-18 (**b**) MSTAR: VGG16 → MobileNet

(**c**) FUSAR: VGG16 → ResNet-18 (**d**) FUSAR: VGG16 → MobileNet

Figure 11. Confusion matrices of black-box transfer attacks using VGG16 as the source model under the BUM framework. The top row (**a**,**b**) displays results on the MSTAR dataset, while the bottom row (**c**,**d**) corresponds to the FUSAR dataset. The vertical axis represents the true labels, and the horizontal axis represents the predicted labels.

On the FUSAR dataset (Figure 11c,d): There is a prominent collapse towards the "Tanker" class. Regardless of whether the original image is a Bulk Carrier, Fishing, or General Cargo ship, the adversarial examples are overwhelmingly misclassified as Tanker, with transfer success rates reaching up to 100% in the "Tanker" column. This suggests that the Tanker class possesses robust, high-confidence scattering features (likely due to its large size and strong corner reflectors) that act as a global attractor in the deep feature space. When the uncertainty of the original class is maximized by BUM, the model gravitates towards this most salient category. On the MSTAR dataset (Figure 11a,b): A similar trend is observed, where classes such as 2S1 and BRDM2 act as the dominant attractors. For instance, in the attack against ResNet-18, the D7 and T62 targets are almost entirely misclassified as 2S1.

4.6.4. Sensitivity Analysis of Kappa

The parameter κ in the attack loss function (Equation (18)) controls the margin of separation between the true class and the most likely incorrect class. A higher κ encourages the generator to produce adversarial examples with higher confidence, theoretically leading to stronger attacks. However, an excessively large κ may make the optimization landscape difficult to traverse or lead to overfitting the surrogate model. To determine the optimal value, we conducted a sensitivity analysis on the MSTAR dataset by varying $\kappa \in \{10, 20, 30, 40\}$. The Average Black-box ASRs for different source models are plotted in Figure 12. As shown in Figure 12, the transferability generally peaks at $\kappa = 20$. For instance, ResNet-18 achieves its highest ASR of 57.4% at $\kappa = 20$, compared to 54.4% at $\kappa = 10$ and a significant drop to 46.0% at $\kappa = 30$. Similarly, the Transformer-based PiT model reaches a local maximum of 67.6% at $\kappa = 20$. VGG-16 exhibits relative stability across different κ values, maintaining a high ASR between 78% and 80%. However, $\kappa = 20$ (79.1%) remains a highly competitive setting, marginally lower than the peak at $\kappa = 30$ (80.0%) but offering better consistency across other architectures. We observe that simply increasing κ does not guarantee better transferability. For example, when κ increases from 20 to 30, the Average Black-box ASRs of ResNet-18 and PiT declines. This suggests that enforcing an overly large margin might force the generator to focus on model-specific high-frequency features to satisfy the strict logit constraint, thereby reducing the generalization capability of the adversarial perturbations to unknown black-box models. Based on these empirical results, we set $\kappa = 20$ as the default value for all experiments to ensure stable and high transferability across diverse source architectures.

4.6.5. Computational Efficiency Analysis

We further evaluate the computational cost of the proposed method. A concern might be that the K stochastic forward passes used during training introduce significant overhead. However, empirical results demonstrate that BUM is computationally efficient compared to the baseline M3D. Training Time: On the same hardware (NVIDIA RTX 4070), training BUM requires approximately 170.4 s per epoch, whereas M3D requires 561.0 s per epoch. BUM is roughly 3.3× faster than M3D. This efficiency stems from the fact that M3D necessitates the concurrent training of two distinct surrogate models to estimate model discrepancy, leading to a complex min-max optimization loop. In contrast, BUM employs a single surrogate model with MC Dropout, significantly reducing the parameter update overhead despite the multiple forward passes. In addition, it is important to clarify that the MC Dropout mechanism is only employed during the training phase to regularize the generator. During the inference (attack generation) phase, the generator G_θ operates in a deterministic mode without stochastic sampling. Consequently, the generation speed of BUM is identical to standard GAN-based methods (e.g., AdvGAN), making it suitable for real-time applications.

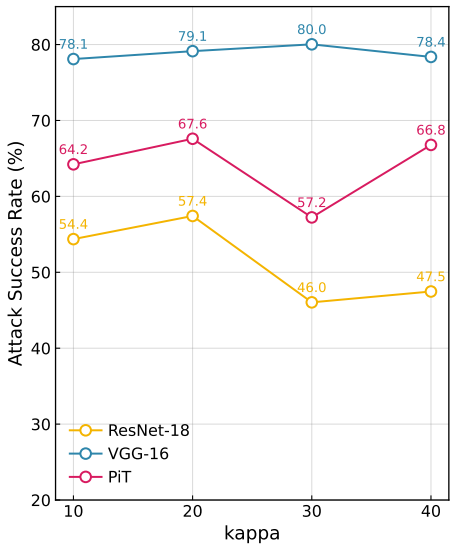


Figure 12. Sensitivity analysis of the confidence parameter κ on the MSTAR dataset. The curves illustrate the Average Black-box ASR (%) of adversarial examples generated by different source models (ResNet-18, VGG-16, and PiT) as κ varies. The results demonstrate that $\kappa = 20$ consistently yields optimal or near-optimal transferability across diverse architectures.

5. Discussion

Interpreting Transferability via Uncertainty Minimization. The primary hypothesis of this study was that the poor transferability of traditional gradient-based attacks (e.g., FGSM, PGD) stems from their tendency to overfit the idiosyncratic decision boundaries of the surrogate model. Our experimental results on MSTAR and FUSAR strongly corroborate this hypothesis. While iterative methods achieve near-perfect white-box success, their ASR performance degrades sharply in black-box settings. In contrast, the proposed BUM framework maintains a high correlation between white-box and black-box ASR performance.

Bridging the Gap Between CNNs and Transformers. A significant finding of this work is the bidirectional transferability between CNNs and ViTs, a challenge that has persisted in the adversarial learning community. Previous literature suggests that CNNs rely heavily on high-frequency textural features, while ViTs prioritize low-frequency global shapes, leading to a "feature misalignment" that hampers transferability. However, our results show that BUM-generated examples from a VGG16 source can fool a Swin Transformer with over 90% success rate. This implies that by minimizing uncertainty, BUM suppresses the model-specific high-frequency noise (which varies greatly between CNNs and ViTs) and instead focuses the perturbation energy on semantic features—specifically, the strong scattering centers of the target vehicles in SAR imagery. These structural features are fundamental to object recognition and are utilized by both convolutional and attention mechanisms, thereby explaining the high cross-architecture success.

The demonstrated vulnerability of modern SAR ATR models—even those with vastly different architectures—has profound implications for the deployment of reliable defense systems. Relying solely on architectural diversity (e.g., using ViTs to defend against CNN-generated attacks) is insufficient against uncertainty-aware attacks like BUM. Future research directions should focus on two aspects: (1). Adversarial Defense: Developing defense mechanisms that explicitly maximize the decision boundary margin or incorporate uncertainty estimation during training to resist such generalized attacks. (2). Physical World Attacks: While our experiments were conducted in the digital domain, extending BUM to account for physical constraints (e.g., manufacturing adversarial reflectors or coatings) remains an open challenge. The structural focus of BUM makes it a promising

candidate for physical implementation, as it targets robust semantic features rather than fragile pixel noise.

6. Conclusions

In this paper, we proposed BUM, a robust generative adversarial attack framework that enhances black-box transferability in SAR ATR by minimizing the epistemic uncertainty of a Bayesian approximation of the surrogate model. This approach effectively mitigates the overfitting to deterministic decision boundaries common in traditional attacks. Comprehensive evaluations on the MSTAR and FUSAR datasets validate that BUM consistently achieves superior transferability compared to existing methods. In conclusion, BUM reveals a critical security vulnerability in current SAR ATR systems: the shared semantic fragility across different deep learning architectures. Our work underscores the importance of uncertainty-aware robustness evaluation and provides a strong baseline for future research in developing more resilient SAR recognition models.

Future work. While BUM demonstrates strong transferability in the digital domain, future research should extend this framework to physical-world attacks against SAR systems, considering complex environmental factors and sensor noises. Additionally, developing uncertainty-aware defense strategies that can detect and reject such high-entropy adversarial examples is a critical direction. Finally, validating the cross-domain transferability of BUM on more complex, large-scale remote sensing datasets beyond MSTAR and FUSAR remains an open avenue for exploration.

Author Contributions: Conceptualization, H.W.; Methodology, H.W.; Software, H.W.; Validation, H.W.; Formal analysis, H.W.; Investigation, H.W.; Resources, Y.L.; Data curation, H.W.; Writing—original draft preparation, H.W.; Writing—review and editing, H.W. and Z.X.; Visualization, H.W.; Supervision, Z.X., F.Y. and T.Z.; Project administration, Y.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by the Beijing Nova Program under Grant 20240484615.

Data Availability Statement: Publicly available datasets were analyzed in this study. The MSTAR data can be found here: https://www.sdms.afrl.af.mil/index.php?collection=mstar (accessed on 19 February 2026). The FUSAR-Ship data can be found here: https://emwlab.fudan.edu.cn/67/05/c20899a222981/page.htm (accessed on 19 February 2026).

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

DNN	Deep Neural Network
ATR	Automatic Target Recognition
MC Dropout	Monte Carlo Dropout
BUM	Bayesian Uncertainty Minimization
CNN	Convolutional Neural Network
GNN	Graph Neural Network
ViT	Vision Transformer
SGD	Stochastic Gradient Descent
ASR	Attack Success Rate
SOTA	State-of-the-Art

References

1. Zhang, M.; An, J.; Yu, D.H.; Yang, L.D.; Wu, L.; Lu, X.Q. Convolutional Neural Network with Attention Mechanism for SAR Automatic Target Recognition. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 4004205. [CrossRef]
2. Lyu, Y.; Wang, Z.; Wang, P.; Li, S.; Tan, H.; Chen, K.; Zhao, L.; Sun, X. Scattering Information and Meta-learning Based SAR Images Interpretation for Aircraft Target Recognition. *J. Radars* **2022**, *11*, 652–665. [CrossRef]
3. Chen, S.; Wang, H. SAR target recognition based on deep learning. In Proceedings of the DSAA 2014—2014 IEEE International Conference on Data Science and Advanced Analytics, Shanghai, China, 30 October–1 November 2014. [CrossRef]
4. Schleher, D.C. Electronic Warfare in the Information Age. In *Review of the Air Force Academy No 3 (27) 2014* ; Artech House on Demand: Boston, MA, USA, 1999; Volume 3.
5. Zhou, F.; Zhao, B.; Tao, M.; Bai, X.; Chen, B.; Sun, G. A large scene deceptive jamming method for space-borne SAR. *IEEE Trans. Geosci. Remote Sens.* **2013**, *51*, 4486–4495. [CrossRef]
6. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. In Proceedings of the 2nd International Conference on Learning Representations, ICLR 2014—Conference Track Proceedings, Banff, AB, Canada, 14–16 April 2014.
7. Huang, T.; Zhang, Q.; Liu, J.; Hou, R.; Wang, X.; Li, Y. Adversarial attacks on deep-learning-based SAR image target recognition. *J. Netw. Comput. Appl.* **2020**, *162*, 102632. [CrossRef]
8. Sun, H.; Chen, J.; Lei, L.; Ji, K.; Kuang, G. Adversarial robustness of deep convolutional neural network-based image recognition models: A review. *J. Radars* **2021**, *10*, 571–594. [CrossRef]
9. Xunzhang, G.; Zhiwei, Z.; Mei, L.; Zhenghui, G.; Xiang, L. Intelligent radar image recognition countermeasures: A review. *J. Radars* **2023**, *12*, 696–712.
10. Ruan, H.; Cui, J.; Mao, X.; Ren, J.; Luo, B.; Cao, H.; Li, H. A Survey of Adversarial Attacks on SAR Target Recognition: From Digital Domain to Physical Domain. *J. Radars* **2024**, *13*, 1298–1326. [CrossRef]
11. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015—Conference Track Proceedings, San Diego, CA, USA, 7–9 May 2015.
12. Xie, C.; Zhang, Z.; Zhou, Y.; Bai, S.; Wang, J.; Ren, Z.; Yuille, A.L. Improving transferability of adversarial examples with input diversity. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; Volume 2019. [CrossRef]
13. Dong, Y.; Pang, T.; Su, H.; Zhu, J. Evading defenses to transferable adversarial examples by translation-invariant attacks. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; Volume 2019. [CrossRef]
14. Long, Y.; Zhang, Q.; Zeng, B.; Gao, L.; Liu, X.; Zhang, J.; Song, J. Frequency Domain Model Augmentation for Adversarial Attack. In *Proceedings of the Lecture Notes in Computer Science*; Lecture Notes in Computer Science (LNCS); Springer: Cham, Switzerland, 2022; Volume 13664. [CrossRef]
15. Gao, L.; Zhang, Q.; Song, J.; Liu, X.; Shen, H.T. Patch-Wise Attack for Fooling Deep Neural Network. In *Proceedings of the Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*; Lecture Notes in Computer Science (LNCS); Springer: Cham, Switzerland, 2020; Volume 12373. [CrossRef]
16. Carlini, N.; Wagner, D. Towards Evaluating the Robustness of Neural Networks. In Proceedings of the IEEE Symposium on Security and Privacy, San Jose, CA, USA, 22–26 May 2017. [CrossRef]
17. Zhang, J.; Qian, W.; Cao, J.; Xu, D. LP-BFGS attack: An adversarial attack based on the Hessian with limited pixels. *Comput. Secur.* **2024**, *140*, 103746. [CrossRef]
18. Sun, H.; Xu, Y.; Kuang, G.; Chen, J. Adversarial robustness evaluation of deep convolutional neural network based SAR ATR algorithm. In Proceedings of the International Geoscience and Remote Sensing Symposium (IGARSS), Brussels, Belgium, 11–16 July 2021. [CrossRef]
19. Du, C.; Huo, C.; Zhang, L.; Chen, B.; Yuan, Y. Fast CW: A Fast Adversarial Attack Algorithm to Fool SAR Target Recognition with Deep Convolutional Neural Networks. *IEEE Geosci. Remote Sens. Lett.* **2022**, *19*, 4010005. [CrossRef]
20. Xu, Y.; Sun, H.; Lei, L.; Ji, K.; Kuang, G. The Research for the Robustness of SAR Ship Identification Based on Adversarial Example. *J. Signal Process.* **2020**, *36*, 1965–1978. [CrossRef]
21. Moosavi-Dezfooli, S.M.; Fawzi, A.; Frossard, P. DeepFool: A Simple and Accurate Method to Fool Deep Neural Networks. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; Volume 2016. [CrossRef]
22. Moosavi-Dezfooli, S.M.; Fawzi, A.; Fawzi, O.; Frossard, P. Universal adversarial perturbations. In Proceedings of the 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, 21–26 July 2017; Volume 2017. [CrossRef]
23. Akhtar, N.; Liu, J.; Mian, A. Defense Against Universal Adversarial Perturbations. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018. [CrossRef]

24. Shafahi, A.; Najibi, M.; Xu, Z.; Dickerson, J.; Davis, L.S.; Goldstein, T. Universal adversarial training. *Proc. AAAI Conf. Artif. Intell.* **2020**, *34*, 5636–5643. [CrossRef]
25. Khrulkov, V.; Oseledets, I. Art of Singular Vectors and Universal Adversarial Perturbations. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018. [CrossRef]
26. Mopuri, K.R.; Ganeshan, A.; Babu, R.V. Generalizable Data-Free Objective for Crafting Universal Adversarial Perturbations. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 2452–2465. [CrossRef]
27. Hayes, J.; Danezis, G. Learning universal adversarial perturbations with generative models. In Proceedings of the 2018 IEEE Symposium on Security and Privacy Workshops, SPW 2018, San Francisco, CA, USA, 24 May 2018. [CrossRef]
28. Mopuri, K.R.; Ojha, U.; Garg, U.; Babu, R.V. NAG: Network for Adversary Generation. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018. [CrossRef]
29. Jandial, S.; Mangla, P.; Varshney, S.; Balasubramanian, V. AdvGAN++: Harnessing latent layers for adversary generation. In Proceedings of the 2019 International Conference on Computer Vision Workshop, ICCVW 2019, Seoul, Republic of Korea, 27–28 October 2019. [CrossRef]
30. Zhao, Z.; Dua, D.; Singh, S. Generating natural adversarial examples. In Proceedings of the 6th International Conference on Learning Representations, ICLR 2018—Conference Track Proceedings, Vancouver, BC, Canada, 30 April–3 May 2018.
31. Liu, X.; Hsieh, C.J. Rob-GAN: Generator, discriminator, and adversarial attacker. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; Volume 2019. [CrossRef]
32. Deb, D.; Zhang, J.; Jain, A.K. AdvFaces: Adversarial face synthesis. In Proceedings of the IJCB 2020—IEEE/IAPR International Joint Conference on Biometrics, Houston, TX, USA, 28 September–1 October 2020. [CrossRef]
33. Wang, J. Adversarial Examples in Physical World. In Proceedings of the IJCAI International Joint Conference on Artificial Intelligence, Montreal, QC, Canada, 19–27 August 2021. [CrossRef]
34. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards deep learning models resistant to adversarial attacks. In Proceedings of the 6th International Conference on Learning Representations, ICLR 2018—Conference Track Proceedings, Vancouver, BC, Canada, 30 April–3 May 2018.
35. Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; Li, J. Boosting Adversarial Attacks with Momentum. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018. [CrossRef]
36. Pang, L.; Wang, L.; Zhang, Y.; Li, H. Adversarial Examples of SAR Images for Deep Learning based Automatic Target Recognition. In Proceedings of the 2021 6th International Conference on Signal and Image Processing, ICSIP 2021, Nanjing, China, 22–24 October 2021. [CrossRef]
37. Ye, T.; Kannan, R.; Prasanna, V.K.; Busart, C. Adversarial attack on GNN-based SAR image classifier. In Proceedings of the Artificial Intelligence and Machine Learning for Multi-Domain Operations Applications V, Orlando, FL, USA, 30 April–5 May 2023. [CrossRef]
38. Gan, F.; Wo, Y. Boosting the Transferability of Adversarial Examples Through Gradient Aggregation. *IEEE Trans. Inf. Forensics Secur.* **2025**, *20*, 5563–5576. [CrossRef]
39. Wang, J.; Chen, Z.; Jiang, K.; Yang, D.; Hong, L.; Guo, P.; Guo, H.; Zhang, W. Boosting the transferability of adversarial attacks with global momentum initialization. *Expert Syst. Appl.* **2024**, *255*, 124757. [CrossRef]
40. Li, H.L.; Chen, S.W. General Polarimetric Correlation Pattern: A Visualization and Characterization Tool for Target Joint-Domain Scattering Mechanisms Investigation. *IEEE Trans. Geosci. Remote Sens.* **2025**, *64*, 5200417. [CrossRef]
41. Li, H.L.; Chen, S.W. Polyhedral corner reflectors multi-domain joint characterization with fully polarimetric radar. *IEEE Trans. Antennas Propag.* **2025**, *73*, 10679–10693. [CrossRef]
42. Li, H.L.; Liu, S.W.; Chen, S.W. PolSAR ship characterization and robust detection at different grazing angles with polarimetric roll-invariant features. *IEEE Trans. Geosci. Remote Sens.* **2024**, *62*, 5225818. [CrossRef]
43. Zhang, T.; Zhang, X.; Li, J.; Xu, X.; Wang, B.; Zhan, X.; Xu, Y.; Ke, X.; Zeng, T.; Su, H.; et al. SAR ship detection dataset (SSDD): Official release and comprehensive data analysis. *Remote Sens.* **2021**, *13*, 3690. [CrossRef]
44. Keydel, E.R.; Lee, S.W.; Moore, J.T. MSTAR extended operating conditions: A tutorial. In Proceedings of the SPIE—The International Society for Optical Engineering, Denver, CO, USA, 4–9 August 1996; Volume 2757.
45. Hou, X.; Ao, W.; Song, Q.; Lai, J.; Wang, H.; Xu, F. FUSAR-Ship: Building a high-resolution SAR-AIS matchup dataset of Gaofen-3 for ship detection and recognition. *Sci. China Inf. Sci.* **2020**, *63*, 140303. [CrossRef]
46. Peng, B.; Peng, B.; Zhou, J.; Xie, J.; Liu, L. Scattering model guided adversarial examples for SAR target recognition: Attack and defense. *IEEE Trans. Geosci. Remote Sens.* **2022**, *60*, 5236217. [CrossRef]
47. Qin, W.; Long, B.; Wang, F. SCMA: A Scattering Center Model Attack on CNN-SAR Target Recognition. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 1–5. [CrossRef]
48. Zhou, J.; Feng, S.; Sun, H.; Zhang, L.; Kuang, G. Attributed Scattering Center Guided Adversarial Attack for DCNN SAR Target Recognition. *IEEE Geosci. Remote Sens. Lett.* **2023**, *20*, 4001805. [CrossRef]

49. Peng, B.; Peng, B.; Xia, J.; Liu, T.; Liu, Y.; Liu, L. Towards assessing the synthetic-to-measured adversarial vulnerability of SAR ATR. *ISPRS J. Photogramm. Remote Sens.* **2024**, *214*, 119–134. [CrossRef]
50. Zhao, A.; Chu, T.; Liu, Y.; Li, W.; Li, J.; Duan, L. Minimizing maximum model discrepancy for transferable black-box targeted attacks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 18–22 June 2023; pp. 8153–8162.
51. Gal, Y.; Islam, R.; Ghahramani, Z. Deep bayesian active learning with image data. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 1183–1192.
52. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
53. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. In Proceedings of the 3rd International Conference on Learning Representations, ICLR 2015—Conference Track Proceedings, San Diego, CA, USA, 7–9 May 2015.
54. Xu, J.; Pan, Y.; Pan, X.; Hoi, S.; Yi, Z.; Xu, Z. RegNet: Self-Regulated Network for Image Classification. *IEEE Trans. Neural Netw. Learn. Syst.* **2023**, *34*, 9562–9567. [CrossRef] [PubMed]
55. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018. [CrossRef]
56. Tan, M.; Le, Q.V. EfficientNet: Rethinking model scaling for convolutional neural networks. In Proceedings of the 36th International Conference on Machine Learning, ICML 2019, Long Beach, CA, USA, 9–15 June 2019; Volume 2019.
57. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16x16 words: Transformers for image recognition at scale. In Proceedings of the ICLR 2021—9th International Conference on Learning Representations, Virtual Event, 3–7 May 2021.
58. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer using Shifted Windows. In Proceedings of the IEEE International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021. [CrossRef]
59. Heo, B.; Yun, S.; Han, D.; Chun, S.; Choe, J.; Oh, S.J. Rethinking Spatial Dimensions of Vision Transformers. In Proceedings of the IEEE International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021. [CrossRef]
60. Lin, J.; Song, C.; He, K.; Wang, L.; Hopcroft, J.E. Nesterov accelerated gradient and scale invariance for adversarial attacks. In Proceedings of the 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.