

## Article

# Adversarial Distillation Defense: A Robust and Lightweight Training Framework for Deep Learning-Based Radar Jamming Recognition

Yifan Peng, Xiaowei Hu \*, Yiduo Guo, Weike Feng, Jian Gong, Hongbing Li and Cunqian Feng

Air and Missile Defense College, Air Force Engineering University, Xi'an 710051, China; 15079925827@163.com (Y.P.); guoyiduo111@126.com (Y.G.); fengweike007@163.com (W.F.); gjdoctor@163.com (J.G.); lihongbing077@163.com (H.L.); fengcunqian2026@163.com (C.F.)

\* Correspondence: xwhu002@163.com

## Abstract

Deep learning models have achieved remarkable performance in radar jamming recognition, yet they remain highly vulnerable to adversarial attacks—small, carefully crafted perturbations that cause misclassification—posing a critical threat to intelligent electronic countermeasure systems. Existing adversarial defenses suffer from an inherent accuracy–robustness tradeoff, limited defensive knowledge sources, and poor generalization to unseen attacks, while the additional challenge of model lightweighting for resource-constrained radar platforms remains largely unaddressed. This paper proposes Adversarial Distillation Defense (ADD), a training framework that synergistically integrates adversarial training with knowledge distillation to produce lightweight yet robust jamming recognition models. In ADD, an adversarially pre-trained teacher model simultaneously transfers its classification knowledge on clean samples and defensive knowledge on adversarial samples to a compact student model through four complementary loss terms. Extensive experiments on a simulated dataset comprising seven radar jamming types demonstrate that ADD achieves the strongest defensive performance among the compared defenses under both white-box and black-box attacks across varying perturbation strengths and jamming-to-noise ratios. Feature-space visualization further confirms that ADD enables the student model to maintain well-separated class clusters even under strong adversarial perturbations. These results indicate that ADD offers an effective strategy for building secure and lightweight deep learning models for radar jamming recognition.

**Keywords:** radar jamming recognition; adversarial attack; adversarial training; knowledge distillation

Academic Editors: Ennio Gambi and Gianluca Ciattaglia

Received: 30 April 2026

Revised: 19 June 2026

Accepted: 26 June 2026

Published: 1 July 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 \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

## 1. Introduction

Radar jamming recognition constitutes one of the core capabilities underpinning modern electronic countermeasures and intelligent radar systems. In complex electromagnetic environments, adversarial jammers typically employ a diverse repertoire of active suppression techniques—including dense false target jamming (DFTJ), interrupted sampling repeater jamming (ISRJ), and smart noise jamming—to severely degrade the target detection and tracking performance of radar systems [1]. The accurate and real-time identification of jamming types is a prerequisite for enabling adaptive anti-jamming

decision-making and safeguarding the operational effectiveness of radar systems, and it remains a critical research topic within the domain of cognitive electronic warfare.

Recent advances in deep learning have introduced a new paradigm for radar jamming recognition [2,3]. End-to-end classification models based on deep neural networks (DNNs)—including convolutional neural networks (CNNs) and residual networks (ResNets) [4]—substantially outperform conventional feature engineering approaches owing to their powerful representation learning capabilities. However, DNNs are inherently vulnerable to adversarial attacks [5]: an adversary can superimpose carefully crafted, imperceptibly small perturbations onto the input signal to induce erroneous classifications with high probability. In radar countermeasure scenarios, such attacks may cause jamming-type misidentification and flawed anti-jamming decisions, severely undermining the operational reliability of the radar system. Ensuring the robustness of DNN-based radar jamming recognition under adversarial conditions has therefore become a pressing concern.

As the threat of adversarial attacks has become increasingly prominent, substantial efforts have been devoted to adversarial defense for DNNs [6,7]. Among these, adversarial training (AT) represents the predominant strategy, enhancing model robustness by injecting adversarial examples into the training process [6,8]. Nevertheless, standard adversarial training methods suffer from several intrinsic limitations: first, the defensive knowledge is derived solely from adversarial examples, providing a single and limited source of robustness information; second, the incorporation of large volumes of adversarial examples typically comes at the cost of reduced classification accuracy on clean samples, creating an inherent tension between accuracy and robustness; and third, overfitting to specific attack modalities tends to impair generalization against unseen attack types [9,10]. Furthermore, in practical deployment scenarios where radar systems impose stringent constraints on computational resources and model size, the adversarial training of lightweight models remains insufficiently explored.

Knowledge distillation (KD) [11], as an important framework for model compression and transfer learning, guides a lightweight student model to learn from a high-performance teacher model, thereby significantly enhancing generalization while preserving model compactness. This framework offers valuable insights for addressing the aforementioned challenges. Integrating knowledge distillation with adversarial training holds the promise of jointly optimizing classification accuracy, adversarial robustness, and model efficiency. However, this line of inquiry has yet to be systematically investigated in the context of radar jamming recognition.

To this end, we propose Adversarial Distillation Defense (ADD), a training framework tailored for radar jamming recognition that integrates adversarial training into a knowledge-distillation paradigm through a four-term objective. ADD builds upon knowledge distillation by incorporating an adversarial training mechanism. While knowledge distillation alone can produce lightweight models, such models lack empirical exposure to adversarial examples. The ADD framework addresses this gap by introducing an adversarially trained teacher model that possesses both classification knowledge on clean samples and defensive knowledge on adversarial samples, thereby guiding the student model to achieve strong performance on both fronts simultaneously. The proposed method is validated on a simulated radar jamming dataset.

The main contributions of this paper are summarized as follows:

- We propose ADD, a training framework that integrates adversarial training with knowledge distillation to produce lightweight yet adversarially robust radar jamming recognition models. To the best of our knowledge, this is the first systematic study of adversarial robustness for radar jamming recognition, a one-dimensional time-domain signal classification task;

- We design a four-term training objective in which an adversarially pre-trained, frozen teacher transfers both its clean-sample classification knowledge and its defensive knowledge on adversarial samples to a compact student. This design differs from existing adversarial robustness distillation methods in its distillation target, its distillation metric, and its use of ground-truth supervision, as detailed in Section 2.3;
- We conduct extensive experiments on a simulated dataset of seven radar jamming types, showing that ADD outperforms representative defense baselines under both white-box and black-box attacks across varying perturbation strengths and jamming-to-noise ratios, while remaining lightweight; ablation, gradient-similarity, and feature-space analyses further confirm the synergy among its loss terms.

The remainder of this paper is organized as follows. Section 2 reviews related work on deep learning-based radar jamming recognition, adversarial attacks, and adversarial defenses. Section 3 introduces the preliminaries, including the radar jamming signal model, the recognition task, adversarial attacks, and knowledge distillation. Section 4 presents the proposed ADD framework and its training procedure. Section 5 reports and analyzes the experimental results under various attack settings, including comparisons with existing defenses, ablation studies, and feature-space visualization. Section 6 concludes the paper.

## 2. Related Work

In this section, we first review existing radar jamming recognition methods, then survey the literature on adversarial attacks, and finally discuss current adversarial defense techniques.

### 2.1. Deep Learning-Based Radar Jamming Recognition

Conventional radar jamming recognition methods rely on prior knowledge and handcrafted features, including statistical decision approaches based on likelihood ratio tests and feature extraction methods based on time-frequency transforms. While such methods are viable when jamming types are limited and the electromagnetic environment is simple, their feature engineering processes are complex, heavily dependent on expert knowledge, and exhibit limited generalization to diverse jamming modalities.

The advent of deep learning has fundamentally transformed radar jamming recognition. Shao et al. were among the first to systematically explore CNN-based radar jamming classification frameworks. In [12], they designed a one-dimensional CNN (1D-CNN) for jamming classification under sufficient-sample conditions for pulse compression radar, and introduced a Siamese network-based scheme to address recognition in few-shot scenarios. In [13], they further proposed a deep feature fusion method integrating 1D-CNN and 2D-CNN, significantly improving classification performance through the synergistic utilization of temporal and time-frequency features. These works established the end-to-end deep learning paradigm for radar jamming recognition. Subsequently, architectures such as ResNet, attention mechanisms, and Transformers have been successively introduced into this field. Zhou et al. [14] proposed a CNN-based approach for recognizing compound jamming, while Zhang et al. [15] constructed a robust radar jamming recognition framework under the open-world setting.

These advances demonstrate that DNNs have substantially surpassed traditional methods in jamming recognition accuracy. However, their security and reliability have yet to be thoroughly examined, particularly regarding their robustness under adversarial threats—a critical issue that demands in-depth investigation.

## 2.2. Adversarial Attacks

Adversarial attacks are generally categorized into white-box and black-box attacks, depending on the degree of the attacker's knowledge about the target model [16,17]. In white-box attacks, the attacker has complete access to the model's architecture and parameters, enabling direct exploitation of gradient information from the loss function with respect to the input to construct adversarial perturbations. In black-box attacks, the attacker can only interact with the model through its input-output interface, without access to internal parameters or gradients.

The security threat that adversarial attacks pose to deep learning models was first systematically revealed by Goodfellow et al. [18] in the general visual classification setting. Their proposed Fast Gradient Sign Method (FGSM) achieves efficient attacks with minimal cost by adding a single-step perturbation along the gradient direction of the loss function under an  $\ell_\infty$ -norm constraint, and they attributed the adversarial vulnerability to the linear nature of neural networks. Madry et al. [8] formulated adversarial attacks as a constrained saddle-point optimization problem and proposed Projected Gradient Descent (PGD) as a "universal first-order adversary", constructing the theoretically strongest first-order white-box attack through multi-step iterations with projection back onto the permissible perturbation set. Dong et al. [19] subsequently introduced the Momentum Iterative FGSM (MI-FGSM), incorporating a momentum term into the iterative process to stabilize gradient update directions while substantially improving the cross-model transferability of adversarial examples. These foundational attack methods established the core paradigm in the vision domain and were subsequently widely adapted to radar and wireless signal recognition.

In the radar domain, Huang et al. [20] pioneered adversarial attack research targeting deep learning-based radar high-resolution range profile (HRRP) recognition models, proposing four fine-grained adversarial perturbation generation algorithms based on binary search and multi-step iteration, and designing universal adversarial perturbations (UAPs) with generalization capability. Their experiments validated the effectiveness and transferability of the attacks under both white-box and black-box settings, demonstrating that adversarial perturbations cause more significant degradation to HRRP recognition models than random noise of equivalent power. Gao et al. [21] provided a comprehensive survey on adversarial attack and defense for intelligent radar image recognition, systematically reviewing attack methods and existing defense strategies for both 1D-HRRP and 2D-synthetic aperture radar (SAR) image recognition models. Zhang et al. [22] extended the adversarial attack paradigm from the recognition side to the radar transmitter, introducing adversarial attack methods into anti-recognition waveform (ARW) design for the first time. They formulated ARW design as a constrained optimization problem based on cross-entropy loss, proposed the VWFGA gradient attack algorithm incorporating weighted predictive gradients, gradient variance adjustment, and adaptive step sizes, along with a random packet ensemble (RPE) strategy based on gradient direction cosine similarity, reducing the average recognition accuracy of 17 target recognition models by 32.82% under black-box conditions while maintaining full compatibility with pulse Doppler and synthetic aperture radar signal processing.

## 2.3. Adversarial Defense

Existing adversarial defense methods broadly fall into two categories: detection-based and mitigation-based approaches. Detection-based methods aim to identify and reject adversarial examples during inference, with common strategies including anomaly detection based on statistical features and filtering via neural rejection mechanisms; for example, Zhang et al. [23] combined neural rejection with label smoothing and Gaussian noise injection, achieving high detection accuracy and significantly improving model security under white-box FGSM attacks. Mitigation-based methods predominantly adopt

adversarial training as the principal paradigm. Madry et al. [8] proposed PGD-based adversarial training (PAT) within a saddle-point optimization framework, establishing the most important baseline defense in the field; however, PAT typically sacrifices clean-sample accuracy due to the injection of large volumes of adversarial examples and tends to overfit to specific attack forms, limiting generalization to unseen attacks. To address this tension, Zhang et al. [24] proposed the TRADES framework, which decomposes adversarial error into natural error and boundary error and achieves a principled accuracy–robustness trade-off via a regularization hyperparameter, demonstrating comprehensive advantages over PAT across multiple robustness evaluation benchmarks.

As an effective training mechanism, knowledge distillation (KD) has also been leveraged to endow DNNs with defense capabilities by directing a lightweight student to learn the “soft target” distribution output by a teacher, transferring implicit knowledge while enhancing generalization. Papernot et al. [25] first introduced defensive distillation (DD), employing the same architecture for both teacher and student to reduce the magnitude of network gradients exploitable by adversaries; however, this mechanism relies on gradient masking, which merely “confuses” the adversary rather than eliminating adversarial examples. Subsequent research has expanded KD from the initial response-based (logit alignment) paradigm to feature-based [26] and relation-based [27] paradigms. Gou et al. [28] provided a comprehensive survey of the field, highlighting that extended paradigms such as multi-teacher, online, and self-distillation offer flexible knowledge-transfer solutions, providing important methodological references for the present work.

Several defense efforts have been specifically directed toward radar systems under adversarial attacks. Zhang et al. [29] observed that general-purpose detection methods suffer severe performance degradation against region-constrained perturbations due to the neglect of distributional discrepancies between adversarial and clean samples; they therefore leveraged an energy-based model (EBM) to capture and amplify this energy gap, yielding an energy-based adversarial detector that requires no modification to the pre-trained classifier. For SAR target recognition, Li et al. [30] proposed SAR-AD-BagNet, which integrates adversarial training with the interpretable BagNet architecture and a Bagging ensemble strategy, achieving decision transparency alongside improved robustness. In summary, while adversarial defense research in radar signal recognition has covered multiple technical approaches and achieved encouraging progress in SAR image and HRRP radar target recognition scenarios, none of the aforementioned works has conducted a systematic adversarial robustness study specifically targeting radar jamming recognition, which involves the classification of one-dimensional time-domain signals.

More closely related to our work is the line of adversarial robustness distillation, which transfers robustness from a robust teacher to a compact student. Goldblum et al. [10] proposed Adversarially Robust Distillation (ARD), which trains the student so that its predictions on adversarial inputs match the predictions of an adversarially pre-trained teacher on the corresponding clean inputs, using a temperature-scaled Kullback–Leibler (KL) divergence together with a hard-label cross-entropy term. Zi et al. [9] subsequently introduced Robust Soft Label Adversarial Distillation (RSLAD), which discards the ground-truth hard labels entirely and instead uses the teacher’s soft predictions on clean samples (robust soft labels) to supervise the student on both clean and adversarial examples, while also incorporating these soft labels into the generation of adversarial examples.

Our proposed ADD belongs to this family but adopts a distinct design, summarized in Table 1 and detailed in Section 4. In brief, ADD differs from ARD and RSLAD in three respects. First, on adversarial inputs, ADD aligns the student with the teacher’s response to the same adversarial sample rather than to the clean sample, thereby transferring the teacher’s behavior under attack rather than its clean-input behavior; for an ideally robust teacher the two targets coincide, whereas for a realistic teacher ADD explicitly distills the

under-attack response. Second, ADD performs this alignment directly in the logit space via mean squared error (MSE) rather than in the probability space via KL divergence. Third, ADD retains ground-truth supervision on both clean and adversarial samples rather than replacing it with soft labels—a choice that our ablation in Section 5.2 shows to be essential for this task. Together, these design choices yield the four-term objective introduced in Section 4 and, as demonstrated in Section 5, translate into consistently stronger robustness for lightweight radar jamming recognition models.

**Table 1.** Comparison of ADD with representative adversarial robustness distillation methods.

| Aspect                          | ARD                    | RSLAD                          | ADD (Ours)                   |
|---------------------------------|------------------------|--------------------------------|------------------------------|
| Adv.-branch distillation target | Teacher on clean input | Teacher on clean input         | Teacher on adversarial input |
| Distillation metric             | KL on softmax          | KL on softmax                  | MSE on raw logits            |
| Hard-label supervision          | Yes (clean branch)     | None (replaced by soft labels) | Yes (clean & adv. branches)  |
| Adv. example generation         | Attack the student     | Teacher-guided                 | Attack the student           |
| Number of loss terms            | 2                      | 2                              | 4                            |

### 3. Preliminaries

In this section, we introduce the fundamental concepts that form the basis for understanding the proposed method.

#### 3.1. Radar Jamming Signal Model

When a radar receiver is subjected to jamming, the received echo signal can be modeled as a superposition of the target echo, the jamming signal, and noise. Let  $x(t)$  denote the received echo signal,  $S(t)$  the target echo,  $N(t)$  the noise, and  $J(t)$  the jamming signal. The received signal can then be expressed as

$$x(t) = S(t) + N(t) + J(t) \quad (1)$$

The jamming effectiveness is typically quantified by the jamming-to-noise ratio (JNR), defined as the ratio of the jamming signal power to the noise power:

$$\text{JNR} = 10 \log_{10} \frac{P_j}{P_n} \quad (\text{dB}) \quad (2)$$

Depending on the underlying mechanism, radar jamming signals can be broadly classified into two categories [15]: noise-type jamming and deception-type jamming. Noise-type jamming suppresses the radar receiver by injecting wideband or narrowband noise signals, thereby reducing the signal-to-noise ratio and submerging the target echo within the jamming background. Deception-type jamming intercepts and retransmits the radar signal to generate false signals that closely resemble authentic target echoes, thereby misleading the radar's target detection and tracking decisions. Different jamming types exhibit distinct statistical characteristics and structural patterns across the time domain, frequency domain, and joint time-frequency domain, providing the essential physical basis for discriminative feature extraction by deep learning models [31].

#### 3.2. Radar Jamming Recognition Task

The central task of radar jamming recognition is the automatic classification of received unknown jamming signals, constituting a canonical multi-class classification problem. To accommodate the real-valued processing paradigm of DNNs, the real and imaginary parts of the complex signal sequence are typically extracted and concatenated along the channel dimension to form a real-valued two-channel input  $\mathbf{x} \in \mathbb{R}^{2 \times N}$ . Suppose the

classification scheme encompasses  $C$  jamming categories. A DNN  $f_\theta$  parameterized by  $\theta$  maps the signal  $\mathbf{x}$  to a logit vector over all classes, which is then normalized via the softmax function to yield the posterior probability distribution:

$$\mathbf{p} = \text{softmax}(f_\theta(\mathbf{x})), \quad p_c = \frac{\exp(f_\theta(\mathbf{x})_c)}{\sum_{c'=1}^C \exp(f_\theta(\mathbf{x})_{c'})} \quad (3)$$

where  $p_c$  denotes the predicted probability that the input belongs to class  $c$ , and the final classification decision  $\hat{y}$  is determined by selecting the class with the highest probability:

$$\hat{y} = \arg \max_{c \in \{1, \dots, C\}} p_c \quad (4)$$

Under the supervised learning framework, the cross-entropy loss is adopted as the training objective function:

$$L_{\text{CE}}(\theta; \mathbf{x}, y) = -\log p_y \quad (5)$$

where  $y$  denotes the ground-truth class label of the sample. Given a training dataset  $D = \{(\mathbf{x}_i, y_i)\}_{i=1}^M$  consisting of  $M$  samples, the network parameters are optimized by minimizing the empirical risk:

$$\min_{\theta} \frac{1}{M} \sum_{i=1}^M L_{\text{CE}}(\theta; \mathbf{x}_i, y_i) \quad (6)$$

thereby driving the network parameters to converge toward a solution with favorable generalization performance.

### 3.3. Adversarial Attacks Formulation

An adversarial attack refers to the deliberate injection of carefully crafted, imperceptibly small perturbations into original input samples to generate adversarial examples, thereby inducing erroneous predictions from the DNN [32]. Let  $f_\theta$  denote the target classifier,  $(\mathbf{x}, y)$  a jamming signal sample and its ground-truth label, and  $\delta$  the adversarial perturbation to be optimized. The construction of an adversarial example  $\mathbf{x}^{\text{adv}}$  can be formally stated as the following constrained optimization problem:

$$\mathbf{x}^{\text{adv}} = \mathbf{x} + \delta^*, \quad \delta^* = \arg \max_{\|\delta\|_p \leq \epsilon} L_{\text{CE}}(f_\theta(\mathbf{x} + \delta), y) \quad (7)$$

where  $\delta^*$  denotes the optimal perturbation that maximizes the classification loss  $L_{\text{CE}}$  within the perturbation constraint,  $\epsilon$  is the perturbation magnitude bound, and  $\|\cdot\|_p$  denotes the  $L_p$  norm. The constraint ensures the perceptual similarity between the adversarial example and the original sample, while the loss maximization drives the model output away from the correct class. For radar signals, this work considers the case of  $p=2$ , where the constraint corresponds to bounding the perturbation signal power [33].

### 3.4. Knowledge Distillation

Knowledge distillation enables a student model to learn inter-class similarity structures from the logit outputs of a teacher model, thereby achieving generalization performance approaching that of a much larger model with a substantially more compact architecture [34]. During distillation, let  $\mathbf{z}^T = [z_1^T, z_2^T, \dots, z_C^T]$  and  $\mathbf{z}^S = [z_1^S, z_2^S, \dots, z_C^S]$  denote the logit output vectors of the teacher and student models, respectively, for an input  $\mathbf{x}$ ,

where  $z_c^T$  and  $z_c^S$  are the raw scores for class  $c$ . The distillation loss is formulated by directly constraining the discrepancy between the logit-level outputs using MSE:

$$L_{soft} = \frac{1}{C} \sum_{c=1}^C (z_c^S - z_c^T)^2 \quad (8)$$

By minimizing the distillation loss  $L_{soft}$ , the student model is guided to approximate the teacher's output at the logit level, thereby transferring the knowledge structure encoded within the teacher to the student. In complete distillation training, the soft-label loss works in concert with the hard-label cross-entropy loss: the hard-label loss ensures that the student maintains the correct classification direction, while the soft-label loss guides the student to absorb deeper knowledge structures from the teacher. Both components are indispensable.

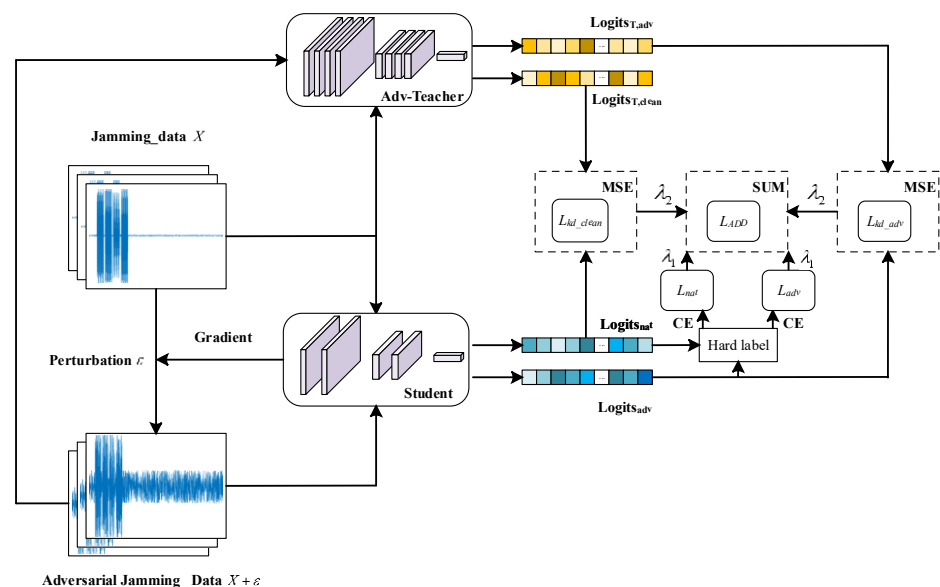
#### 4. Adversarial Distillation Defense Framework

In this section, we present the proposed ADD framework in detail and analyze the underlying principles of its design.

##### 4.1. Framework Design

The ADD framework first requires the training of a high-performance teacher model that exhibits strong performance on both clean and adversarial samples. Adversarial training is therefore employed during the pre-training stage to ensure that the teacher simultaneously possesses superior classification accuracy and robust defense capability.

The overall architecture of ADD is illustrated in Figure 1. The framework comprises two models: a student model, which serves as the lightweight model ultimately deployed for radar jamming recognition, and an adversarially pre-trained teacher model (Adv-Teacher).



**Figure 1.** Overall architecture of the ADD framework.

During student network training, adversarial examples  $X^{adv}$  are first generated by attacking the current student model on clean samples  $X$ . Both clean and adversarial samples are then fed into the student and teacher models to obtain the respective logit outputs. Let  $Logits_{nat}$  and  $Logits_{adv}$  denote the student's logit outputs on clean and adversarial samples,  $Logits_{T,clean}$  and  $Logits_{T,adv}$  the teacher's corresponding logit outputs,  $Y$  the ground-truth label. The ADD framework constructs the following four loss terms based on these outputs.

Clean-sample classification loss  $L_{nat}$ . The cross-entropy loss of the student model on clean samples, which preserves the baseline classification capability for normal jamming signals:

$$L_{nat} = \text{CrossEntropy}(\text{Softmax}(\text{Logits}_{nat}), \mathbf{Y}) \quad (9)$$

Adversarial-sample classification loss  $L_{adv}$ . The cross-entropy between the student's predictions on adversarial samples and the ground-truth labels, driving the student to produce correct classifications even under adversarial perturbations. This term directly embodies the adversarial training mechanism within ADD:

$$L_{adv} = \text{CrossEntropy}(\text{Softmax}(\text{Logits}_{adv}), \mathbf{Y}) \quad (10)$$

Clean-sample distillation loss  $L_{kd\_clean}$ . The MSE between the student's and teacher's logit outputs on clean samples, guiding the student to approximate the teacher's classification knowledge structure on normal inputs:

$$L_{kd\_clean} = \text{MSE}(\text{Logits}_{nat}, \text{Logits}_{T, clean}) \quad (11)$$

Adversarial-sample distillation loss  $L_{kd\_adv}$ . The MSE between the student's and teacher's logit outputs on adversarial samples, guiding the student to acquire the teacher's robust decision boundary knowledge on adversarial inputs. This term transfers the defensive capability accumulated through the teacher's adversarial training to the student via the distillation mechanism:

$$L_{kd\_adv} = \text{MSE}(\text{Logits}_{adv}, \text{Logits}_{T, adv}) \quad (12)$$

The total loss of the ADD training framework is, thus,

$$L_{ADD} = \lambda_1(L_{nat} + L_{adv}) + \lambda_2(L_{kd\_clean} + L_{kd\_adv}) \quad (13)$$

where  $\lambda_1$  and  $\lambda_2$  are the hyperparameters to balance the importance of each part.

#### 4.2. Training Procedure

Algorithm 1 details the complete training procedure of the ADD framework. First, the student model parameters are initialized. In each training epoch, for every mini-batch, adversarial examples are generated by attacking the current student model on clean samples. Both clean and adversarial samples are then forwarded through the student and teacher models to obtain their respective logit outputs. Based on these outputs, the four sub-losses—clean-sample classification loss, adversarial-sample classification loss, clean-sample distillation loss, and adversarial-sample distillation loss—are computed and summed to obtain the total loss  $L_{ADD}$ . The student model parameters are then updated via backpropagation using this total loss. Crucially, the teacher model parameters remain frozen throughout the entire training process; only the student model participates in gradient updates, ensuring that defensive knowledge is transferred unidirectionally from teacher to student and preventing teacher degradation.

**Algorithm 1.** Adversarial Distillation Defense of Student Model

**Input:** Student model  $f_S(\cdot, \theta_S)$ ; Adversarial teacher model  $f_T(\cdot, \theta_T)$ ; Number of training epochs  $k$ ; Learning rate  $\eta$ ; Training set  $\mathbf{D}_{train}$ ; Ground-truth labels  $\mathbf{Y}$ .

**Output:** Trained student model  $f_S(\cdot, \theta_S)$

1. Initialize student model parameters  $\theta_S = \theta_{S,0}$
2. For training epoch 1 to  $k$  do
3.   For each batch of jamming signal data  $\mathbf{X}^i \in \mathbf{D}_{train}, \mathbf{Y}^i \in \mathbf{Y}$  do
4.     Generate adversarial examples  $\mathbf{X}_{adv}^i$
5.      $f_S(\mathbf{X}^i, \theta_S) \xrightarrow{out} \text{Logits}_{nat}$
6.      $f_T(\mathbf{X}^i, \theta_T) \xrightarrow{out} \text{Logits}_{T, clean}$
7.      $f_S(\mathbf{X}_{adv}^i, \theta_S) \xrightarrow{out} \text{Logits}_{adv}$
8.      $f_T(\mathbf{X}_{adv}^i, \theta_T) \xrightarrow{out} \text{Logits}_{T, adv}$
9.      $L_{nat} = \text{CrossEntropy}(\text{Softmax}(\text{Logits}_{nat}), \mathbf{Y})$
10.     $L_{adv} = \text{CrossEntropy}(\text{Softmax}(\text{Logits}_{adv}), \mathbf{Y})$
11.     $L_{kd\_clean} = \text{MSE}(\text{Logits}_{nat}, \text{Logits}_{T, clean})$
12.     $L_{kd\_adv} = \text{MSE}(\text{Logits}_{adv}, \text{Logits}_{T, adv})$
13.     $L_{ADD} = \lambda_1(L_{nat} + L_{adv}) + \lambda_2(L_{kd\_clean} + L_{kd\_adv})$
14.     $\theta_S = \theta_S - \eta \cdot \nabla_{\theta_S} L_{ADD}$
15.   End for
16. End for
17. Return  $f_S(\cdot, \theta_S)$

The integration of adversarial training and knowledge distillation within the ADD framework is not a mere superposition but rather constitutes a synergistic relationship of mutual complementarity and reinforcement, with each loss term playing a distinct role during training. Adversarial training, through  $L_{adv}$ , exerts continuous adversarial pressure on the student, driving it to actively learn correct classifications under perturbed inputs. Knowledge distillation, through  $L_{kd\_adv}$ , provides the student with the teacher's defensive knowledge guidance, compensating for the limited defense capability that arises from insufficient model capacity during standalone adversarial training of lightweight models. Concurrently,  $L_{kd\_clean}$  continuously steers the student toward the teacher's classification knowledge, effectively mitigating the negative impact of adversarial training on clean-sample accuracy, thereby maintaining classification performance while enhancing robustness.  $L_{nat}$ , as the student's original classification loss, helps preserve the model's general-purpose capability and prevents it from being entirely dominated by the teacher model.

Under the ADD framework, the four loss terms are jointly optimized, subjecting the student model to dual constraints of classification supervision and knowledge distillation on both clean and adversarial samples. This enables the collaborative enhancement of classification accuracy and adversarial robustness under a limited parameter budget. Compared with standard adversarial training, ADD additionally introduces a clean-sample pathway and two distillation loss terms beyond the adversarial-sample pathway, yielding a richer source of defensive knowledge and effectively alleviating the trade-off between classification performance and adversarial defense. Furthermore, through the distillation mechanism, a more lightweight model can be trained whose performance is comparable to that of a substantially larger model.

## 5. Experimental Results and Analysis

This section validates the effectiveness of the proposed method through comprehensive experiments. We train and adversarially attack lightweight defense models on a simulated dataset and compare their classification and defense performance under various robustness training frameworks.

### 5.1. Experimental Setup

#### 5.1.1. Jamming Signal Dataset

For the radar jamming recognition task in this study, a simulated radar jamming signal dataset was generated using MATLAB R2021a. The radar received echo signal  $x(t)$  comprises the target echo  $S(t)$ , noise  $N(t)$ , and jamming signal  $J(t)$ . The target echo  $S(t)$  is based on a single-pulse linear frequency-modulated (LFM) signal with the parameters specified in Table 2.

**Table 2.** Target echo signal parameters.

| Parameter          | Value       |
|--------------------|-------------|
| Bandwidth          | 10 MHz      |
| Pulse width        | 20 $\mu$ s  |
| Sampling frequency | 20 MHz      |
| Time delay         | 0–5 $\mu$ s |
| SNR                | 0 dB        |

The jamming signal  $J(t)$ , also based on the LFM signal, encompasses seven jamming types: Dense False Target Jamming (DFTJ), Interrupted Sampling Repeater Jamming (ISRJ), Noise Multiplication Jamming (NMJ), Comb Spectrum Jamming (CSJ), Narrow Pulse Jamming (NPJ), Barrage Jamming (BJ), and Convolution Noise Jamming (CNJ). The principal parameters of the jamming signals are listed in Table 3.

**Table 3.** Jamming signal parameters.

| Parameter                     | Value        |
|-------------------------------|--------------|
| DFTJ: number of false targets | 3/4/5/6      |
| ISRJ: number of repeats       | 1/2/3/4      |
| CSJ: number of spectral lines | 10           |
| NPJ: number of pulses         | 5/6/7/8/9/10 |
| BJ: spot bandwidth            | 20–40 MHz    |
| NMJ: modulation period        | 5/10 $\mu$ s |
| CNJ: convolution-noise length | 10 $\mu$ s   |

When generating the radar echo signals, Gaussian white noise is adopted for  $N(t)$ . For each of the seven jamming types, 10,000 signal samples are generated across 10 JNR levels ranging from −10 dB to 35 dB in 5 dB increments (1000 samples per JNR level), yielding a total of 70,000 signal samples. The dataset is split into 56,000 training samples and 14,000 test samples at a ratio of 4:1, stratified across jamming types and JNR levels.

Each jamming signal datum is a complex-valued vector of dimension  $1 \times 2048$ . For compatibility with subsequent processing, the real and imaginary parts are separated and stored as a  $2 \times 2048$  real-valued matrix, where the first row corresponds to the real part and the second row to the imaginary part of the echo signal.

### 5.1.2. Model Architectures

Since the ADD framework aims to construct lightweight models for radar jamming recognition, the selected architectures are required to possess signal classification capability while maintaining a small parameter footprint. To this end, we adopt three student models: (i) 1D-CNN, a ResNet-based architecture [35] that is among the most widely used models for signal classification; (ii) ResNet8, the simplest variant in the ResNet family, adapted with one-dimensional convolutions to accommodate signal sequence inputs; and (iii) MobileNet [36], a lightweight model designed for edge devices. For the adversarial teacher (Adv-Teacher), we employ VGG16, a high-capacity model with relatively strong performance.

As summarized in Table 4, the three student models are markedly more lightweight than the teacher and are well suited to resource-constrained radar platforms.

**Table 4.** Computational cost of the teacher and the three student models.

| Model     | Role    | Params (M) | FLOPs (M) | Size (MB) | GPU lat (ms) | CPU lat (ms) |
|-----------|---------|------------|-----------|-----------|--------------|--------------|
| VGG16     | Teacher | 24.58      | 12618.5   | 93.78     | 5.11         | 81.75        |
| 1D-CNN    | Student | 0.23       | 152.5     | 0.86      | 3.81         | 8.24         |
| MobileNet | Student | 3.19       | 654.6     | 12.17     | 3.40         | 13.77        |
| ResNet8   | Student | 0.03       | 17.5      | 0.10      | 0.78         | 3.25         |

### 5.1.3. Adversarial Attack Methods

To validate the effectiveness of the proposed method, we employ multiple adversarial attack methods to assess the robustness of models trained under the ADD framework. Specifically, MI-FGSM [19], PGD [8], and AutoAttack (AA) [37] are adopted as white-box attack methods, while the Square attack [38] serves as the black-box attack method. These attacks are among the most commonly used and potent adversarial attack methods for deep learning models. All adversarial perturbations in this work are bounded in the  $L_2$  norm, consistent with the power-constrained threat model of Section 3.3.

For AutoAttack we use the standard version [37]—the ensemble of four parameter-free attacks: untargated APGD on the cross-entropy loss (APGD-CE), targeted APGD on the DLR loss (APGD-T), targeted FAB (FAB-T), and the query-based black-box Square attack—with APGD run for 100 iterations and Square allotted 3000 queries, all at the same perturbation budget as the other attacks. Because this ensemble combines strong gradient-based (white-box) and gradient-free (black-box) components, AutoAttack provides an attack-agnostic robustness estimate and serves as our primary white-box evaluation.

### 5.1.4. Baseline Defense Methods

To demonstrate the superiority of the proposed ADD framework, it is compared against several established adversarial defense methods, including PGD-based adversarial training (PAT) [8], DD [25], and TRADES [24]. These methods represent powerful and widely adopted adversarial defense strategies.

### 5.1.5. Implementation Details

All experiments are conducted on a workstation equipped with an Intel Core i9-9900KF CPU (3.60 GHz) (Intel, Santa Clara, CA, USA) and an NVIDIA GeForce RTX 2080 Ti GPU (NVIDIA, Santa Clara, CA, USA). Adversarial example generation and model training are performed on a single GPU, and all models are implemented using the PyTorch (2.1.2+cu118) framework. For initial training of both student and teacher models, the batch size is set to 128, the Adam optimizer is employed with an initial learning rate of 0.001, and a step-decay learning rate schedule (StepLR) is applied with a decay factor of 0.8 every 10 epochs. Cross-entropy is used as the loss function, and the random seed is fixed at 111 to ensure reproducibility. For the initial (standard) training of the student and

teacher models, training is run for a fixed schedule of 100 epochs, by which point the training loss has converged. No early stopping or model selection is performed on the test set: the test set is held out and used exclusively for final evaluation, and all hyperparameters are fixed in advance and shared across all methods to ensure a fair comparison. The teacher model (VGG16) is adversarially pre-trained with standard PGD-based adversarial training [8], using PGD-generated adversarial examples with perturbation budget  $\epsilon = 0.06$ , step size  $\eta = 0.03$ , and iteration count  $t = 5$ ; all other training settings follow those described above for the student models. The pre-trained teacher achieves 100% clean accuracy and 90.57% robust accuracy under PGD ( $\epsilon = 0.10$ ). Identical perturbation parameters are applied to PAT and TRADES for fairness. For DD, the temperature parameter is set to 30. The training duration for both the ADD framework and all competing defense methods is 200 epochs.

## 5.2. Design Validation of ADD

### 5.2.1. Ablation on Loss Weights and Distillation Form

Before validating the overall effectiveness of ADD, we examine two key design choices in the ADD objective of Equation (13): the relative weights  $\lambda_1$  and  $\lambda_2$  assigned to the classification-loss group ( $L_{nat} + L_{adv}$ ) and the distillation-loss group ( $L_{kd\_clean} + L_{kd\_adv}$ ), respectively, and the specific form of the distillation loss. For the distillation terms, we compare the logit-level MSE adopted by ADD with a temperature-scaled Kullback–Leibler (KL) divergence as a control, in which the teacher and student logits are softened by a temperature  $\tau$  into probability distributions and the KL divergence between them is used as the per-branch distillation loss. As ARD [10] reports that adversarial robustness distillation is largely insensitive to the temperature, with no significant effect on the student’s clean or robust accuracy, we fix  $\tau = 30$  throughout.

All experiments in this subsection use the 1D-CNN student and the adversarially pre-trained VGG16 teacher, with the training settings of Section 5.1.5. We first detail how the attack test set is constructed, since the same procedure is used—unless otherwise noted—in all subsequent robustness evaluations. After a model is trained, it is evaluated under a white-box PGD attack; to form the attack test set, we first screen the samples that the model classifies correctly on the original test set and then randomly select 10,000 of them using a fixed random seed. Screening for correctly classified samples ensures that the accuracy degradation under attack reflects the effect of the perturbation rather than pre-existing misclassifications, while the size of 10,000 keeps the test set sufficiently large to minimize the influence of data selection. The PGD attack is applied with a perturbation budget  $\epsilon = 0.10$ , step size  $\eta = 0.03$ , and  $t = 5$  iterations. For each configuration of  $(\lambda_1, \lambda_2)$ , the resulting clean accuracy and robust accuracy are reported in Table 5.

**Table 5.** Effect of loss weights and distillation form.

| $\lambda_1$ | $\lambda_2$ | MSE Clean (%) | MSE PGD (%) | KL Clean (%) | KL PGD (%) |
|-------------|-------------|---------------|-------------|--------------|------------|
| 0.1         | 0.9         | 96.75         | 40.29       | 97.08        | 41.83      |
| 1           | 1           | 100           | 48.94       | 100          | 50.22      |
| 1           | 0.1         | 100           | 74.51       | 100          | 75.36      |
| 2           | 0.1         | 100           | 77.42       | 100          | 76.91      |
| 5           | 0.1         | 100           | 76.03       | 100          | 75.77      |
| 10          | 0.1         | 100           | 74.71       | 100          | 74.65      |

We first analyze the effect of the loss weights. The clean accuracy is almost insensitive to the weight configuration, remaining at full accuracy for the great majority of settings and decreasing only slightly when the classification-loss group is substantially down-

weighted; this indicates that adjusting the weights does not sacrifice standard classification performance. The robust accuracy, in contrast, is governed mainly by the relative weight of the two loss groups: when the distillation-loss group is weighted too heavily, the robust accuracy drops sharply, whereas once it is assigned a moderately small weight, the robust accuracy stabilizes on a high plateau over a wide range of classification-loss weights, fluctuating only slightly and reaching its maximum at  $\lambda_1 = 2$ ,  $\lambda_2 = 0.1$ . These results show that, provided the distillation-loss group is given a moderately small weight, ADD is insensitive to the precise weight values; we therefore adopt  $\lambda_1 = 2$  and  $\lambda_2 = 0.1$  as the default configuration. Unless otherwise specified, all subsequent experiments adopt this default configuration ( $\lambda_1 = 2$ ,  $\lambda_2 = 0.1$ ) with logit-level MSE as the distillation loss; that is, the ADD model evaluated in Sections 5.2.2–5.5 is the configuration with the highest robust accuracy identified in this study.

We next analyze the effect of the distillation loss form. As shown in Table 4, the MSE and KL distillation losses perform comparably overall, with very small differences between them and essentially identical clean accuracy. More importantly, the highest robust accuracy across all configurations is attained with MSE, and throughout the high-robustness region—the configurations with a small distillation weight and  $\lambda_1 \geq 2$ —MSE is consistently slightly superior to KL. This behavior is consistent with the known intrinsic relationship between the two losses: in the high-temperature limit, the gradient of the KL distillation loss with respect to the student logits reduces to that of the logit-level MSE [11]. Kim et al. [39] further show that the KL loss interpolates between logit matching at large temperatures and label matching at small temperatures, that logit matching is generally correlated with improved generalization, and that directly regressing the logits via MSE attains performance comparable to—or better than—KL while dispensing with the temperature hyperparameter. The knowledge transferred by direct logit regression is therefore essentially equivalent to that conveyed by softmax-based KL divergence. Taken together, MSE both eliminates the temperature hyperparameter required by KL and attains the highest robustness at the optimal configuration. We accordingly adopt logit-level MSE as the default distillation loss in ADD: it achieves robustness comparable to, or better than, KL with fewer hyperparameters to tune, which is better aligned with our goal of lightweight deployment.

### 5.2.2. Overall Effectiveness of ADD

We first conduct a preliminary experiment to validate the effectiveness of the proposed ADD method by comparing six training strategies for the student model: (i) standard training on clean data (Original) using  $L_{nat}$ ; (ii) knowledge distillation (KD) using  $L_{nat}$  and  $L_{kd\_clean}$ ; (iii) adversarial training (AT) using  $L_{nat}$  and  $L_{adv}$ ; (iv) teacher-guided distillation (TD) using  $L_{kd\_clean}$  and  $L_{kd\_adv}$  exclusively; (v) adversarial training with teacher guidance (ATT) using  $L_{nat}$ ,  $L_{adv}$ , and  $L_{kd\_clean}$ ; and (vi) the proposed ADD method with all four loss terms. All ablation models are trained for the same 200 epochs schedule as the main experiments (specified in Section 5.1.5), with all other hyperparameters kept consistent.

After training, all three student models are evaluated under the PGD attack. For each model, the attack test set is constructed and the PGD attack configured exactly as described in Section 5.2.1, and the classification accuracy under attack is computed for each training strategy. The results are reported in Table 6.

**Table 6.** Validation of ADD effectiveness.

| Student Model | Training Method Accuracy (%) |       |       |       |       |       |
|---------------|------------------------------|-------|-------|-------|-------|-------|
|               | Original                     | KD    | AT    | TD    | ATT   | ADD   |
| 1D-CNN        | 0.2                          | 34.67 | 68.36 | 23.45 | 68.52 | 74.51 |
| MobileNet     | 0.92                         | 22.94 | 78.76 | 24.37 | 79.81 | 88.1  |
| ResNet8       | 7.3                          | 23.46 | 55.35 | 22.85 | 56.25 | 67.97 |

Table 5 reports the PGD-attack accuracy of each student model under the six training strategies. Under standard training (Original), all three students collapse almost completely, retaining only a trivial fraction of their clean-sample accuracy. These near-zero accuracies confirm that undefended DNNs are entirely unable to withstand PGD perturbations. Introducing clean-sample knowledge distillation alone (KD) yields only a modest gain, indicating that distillation from a naturally trained teacher offers mild regularization but no genuine adversarial defense, since the teacher itself has no knowledge of adversarial samples to transfer.

AT produces a qualitative leap across all three students, confirming the established role of adversarial training as a strong baseline defense. In contrast, teacher-guided distillation (TD)—which relies exclusively on the clean-sample and adversarial-sample distillation losses without any ground-truth supervision—barely exceeds KD and falls far short of AT. The pronounced gap between AT and TD demonstrates that teacher guidance alone is insufficient for adversarial defense: without direct supervision from the ground-truth label on adversarial samples, the student cannot form decision boundaries that correctly classify adversarial examples. The adversarial classification loss is therefore the indispensable foundation of any effective defense within this framework.

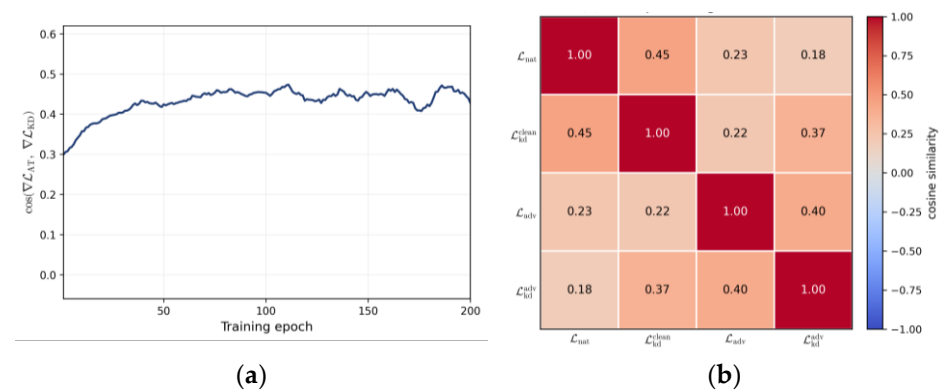
Augmenting AT with clean-sample teacher distillation (ATT) yields only marginal improvements over AT across the three students, indicating that clean-sample knowledge transfer contributes little additional robustness once adversarial training is already in place. The decisive improvement arises from the complete ADD framework, which additionally incorporates the adversarial-sample distillation loss. ADD consistently outperforms both AT and ATT by a substantial and stable margin on every student model. This pronounced gap from ATT to ADD isolates the unique contribution of transferring the teacher’s defensive knowledge on adversarial samples, demonstrating that the four loss terms together produce a synergy that none of the simpler combinations can attain.

Examining the results across architectures, MobileNet achieves the highest accuracy under every defense-augmented strategy, followed by 1D-CNN, with ResNet8 consistently trailing. Overall, ADD simultaneously achieves the best robustness on every student model tested, comprehensively validating the effectiveness of the proposed method and confirming that all four loss terms are necessary to unlock its full defensive potential.

### 5.2.3. Gradient Similarity Analysis

The ablation in Section 5.2.2 establishes that the four loss terms of ADD are jointly necessary. To further probe the underlying optimization mechanism of ADD and to substantiate the qualitative argument of Section 4.2, we examine the geometric relationship among the gradients that the four loss terms induce on the student model. On the 1D-CNN student, we compute the gradient of each loss term with respect to the student parameters,  $g_k = \nabla_{\theta_S} L_k$ , and measure the pairwise cosine similarity between these gradients. We further aggregate the terms into an adversarial-training objective ( $L_{nat} + L_{adv}$ ) and a knowledge-distillation objective ( $L_{kd\_clean} + L_{kd\_adv}$ ), and track the alignment of their gradients over the course of training. The weight of each term follows Equation (13), and the teacher parameters are frozen throughout.

As shown in Figure 2a, the AT and KD objectives remain positively aligned throughout training, with their cosine similarity stable between approximately 0.3 and 0.45 over the entire training process; the two paradigms therefore never pull the student in mutually conflicting directions. At the same time, this plateau lies well below 1, indicating that each objective retains a substantial independent gradient component and that neither paradigm is redundant with respect to the other. Figure 2b confirms this at the level of the individual loss terms: all pairwise cosine similarities are positive, and none approaches 1. The four terms are thus mutually compatible yet non-redundant—precisely the signature of synergy in gradient space rather than a mere superposition of independent objectives. Together with the performance gains in Section 5.2.2, this geometric evidence indicates that the improvement achieved by ADD arises from a genuine synergy among its loss terms.

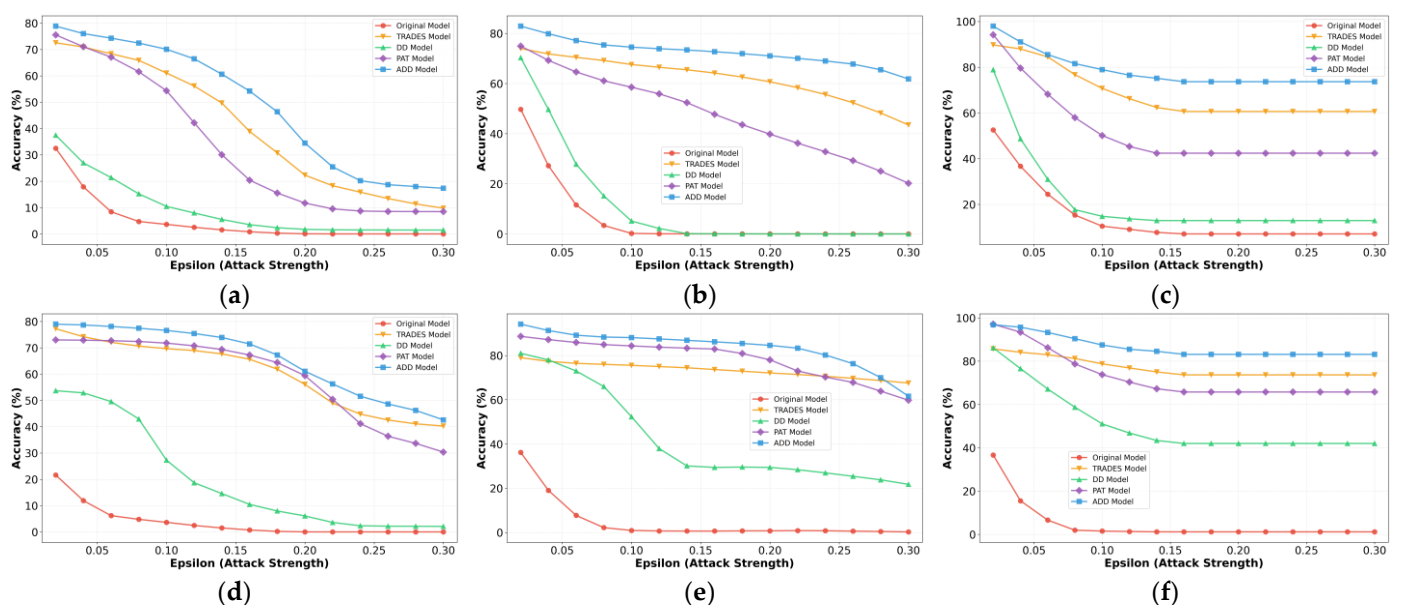


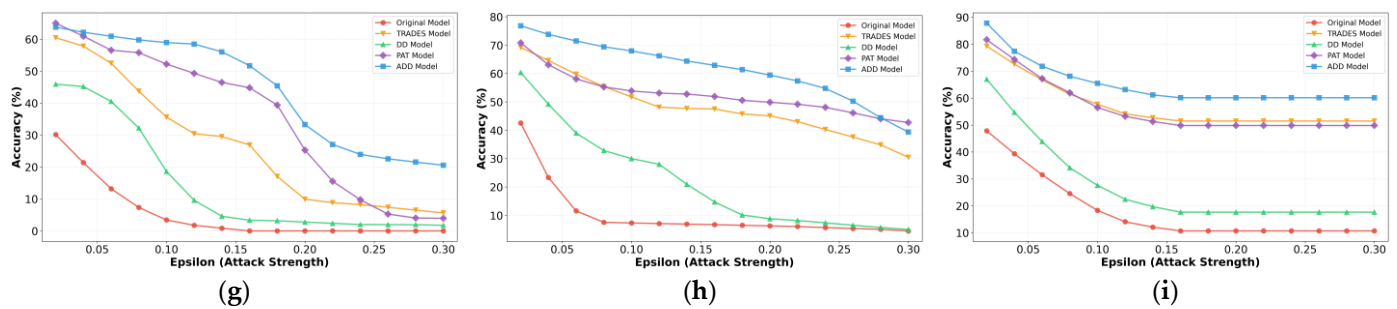
**Figure 2.** Gradient similarity analysis of the ADD loss terms on the 1D-CNN student. (a) Gradient alignment of AT and KD. (b) Mean pairwise gradient cosine.

### 5.3. Comparison with Existing Defense Methods

#### 5.3.1. Defense Performance Under Varying Attack Strengths

This subsection evaluates the defense performance of ADD under different attack intensities and compares it with competing defense frameworks. Following a procedure similar to Section 5.2.1, we randomly select 10,000 correctly classified samples per student model to form the attack test set, and then apply MI-FGSM [19], PGD [8], and AA [37] attacks at varying perturbation strengths. These three attacks are chosen because they all feature a tunable perturbation budget hyperparameter  $\epsilon$ . The results are presented in Figure 3.





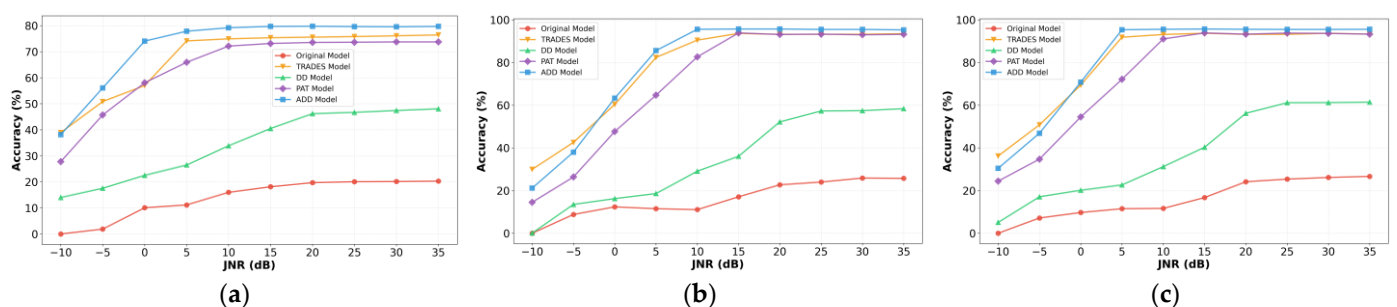
**Figure 3.** Defense performance comparison under varying attack strengths for three student models and three attack methods. (a) AA attack on 1D-CNN. (b) PGD attack on 1D-CNN. (c) MI-FGSM attack on 1D-CNN. (d) AA attack on MobileNet. (e) PGD attack on MobileNet. (f) MI-FGSM attack on MobileNet. (g) AA attack on ResNet8. (h) PGD attack on ResNet8. (i) MI-FGSM attack on ResNet8.

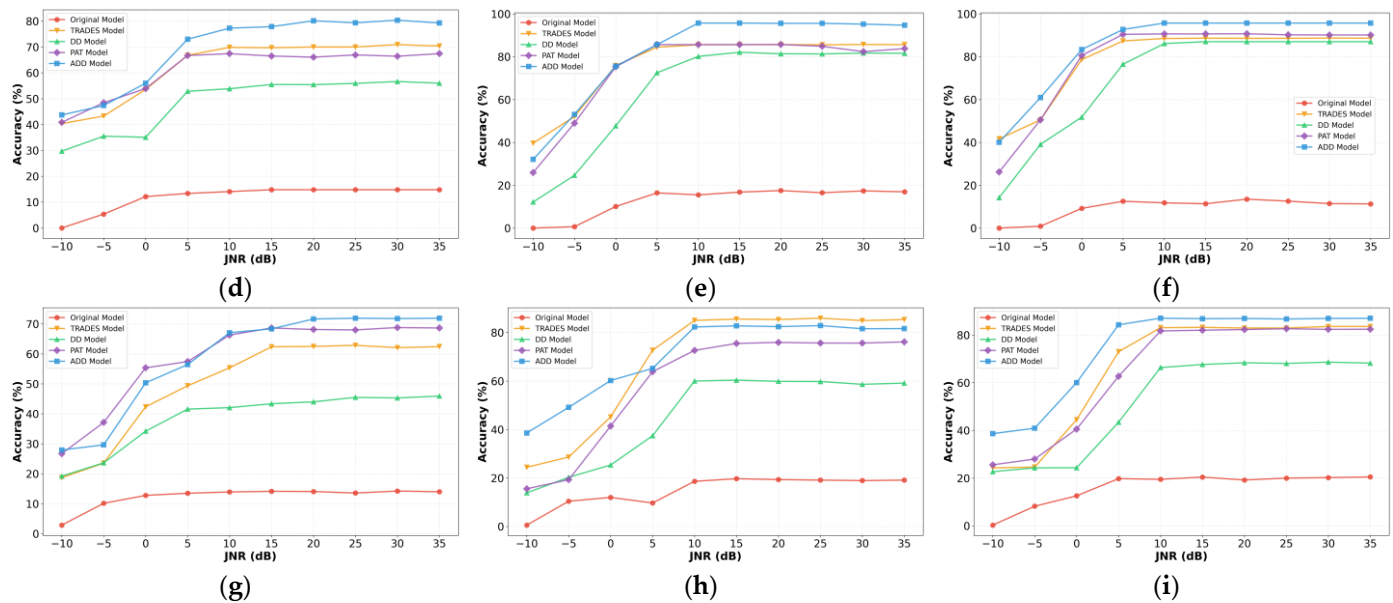
As the perturbation budget  $\epsilon$  increases, the classification accuracy of all methods declines, but ADD consistently exhibits the slowest degradation and maintains the highest accuracy over the entire range, demonstrating superior robustness on all three student models. PAT and TRADES afford effective protection at small budgets yet degrade more rapidly as  $\epsilon$  grows, whereas DD—relying solely on distillation without adversarial-example priors—and the undefended Original model collapse the fastest. Across architectures, MobileNet is the most resilient and ResNet8 the most susceptible, with 1D-CNN in between; the only case in which ADD does not strictly lead is ResNet8 at the largest budgets, where PAT becomes marginally competitive.

Comparing the three attacks reveals the conventional ordering of attack strength: AutoAttack is the strongest, yielding the lowest accuracies; PGD is intermediate; and the momentum-based MI-FGSM is the weakest, leaving the highest accuracies. This ordering—a strong ensemble attack (AutoAttack) dominating a single transfer-oriented momentum attack (MI-FGSM)—is exactly what is expected of a genuinely robust model and is an important indication that the robustness of ADD does not arise from gradient masking or obfuscation. ADD retains the best robustness under all three attacks on all three models, further corroborating the effectiveness of the proposed framework.

### 5.3.2. Defense Performance Under Varying JNR Levels

We further compare ADD with existing robustness training methods across different JNR levels. At each JNR level, 1000 correctly classified samples are randomly selected (covering the majority of correctly classified samples at that level) to form the attack test set. The results are shown in Figure 4.



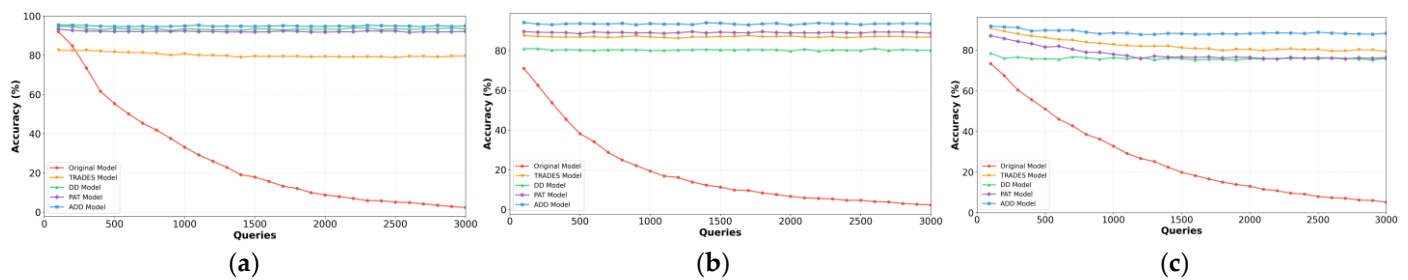


**Figure 4.** Defense performance comparison under varying JNR levels for three student models and three attack methods. (a) AA attack on 1D-CNN. (b) PGD attack on 1D-CNN. (c) MI-FGSM attack on 1D-CNN. (d) AA attack on MobileNet. (e) PGD attack on MobileNet. (f) MI-FGSM attack on MobileNet. (g) AA attack on ResNet8. (h) PGD attack on ResNet8. (i) MI-FGSM attack on ResNet8.

As shown in Figure 4, models trained with ADD achieve the highest classification accuracy across virtually all JNR levels, attack methods, and architectures. Accuracy increases with JNR and saturates toward the high-JNR end, where the strong defenses (ADD, PAT, TRADES) converge toward a common ceiling; in the more challenging low-to-moderate JNR regime, ADD separates most clearly from the competing methods. On the compact ResNet8 student, ADD remains the most accurate at every JNR level, but its margin over PAT and TRADES narrows at high JNR as the curves saturate. PAT and TRADES again provide solid robustness, whereas DD trails substantially and the undefended model remains near the bottom. Consistent with Section 5.3.1, the architecture ordering is MobileNet > 1D-CNN > ResNet8, and the attack-strength ordering is AutoAttack > PGD > MI-FGSM across JNR levels. Overall, ADD delivers a consistent robustness advantage over existing methods across the full JNR range, most pronounced at low-to-moderate JNR.

### 5.3.3. Defense Performance Against Black-Box Attacks

The Square attack [38] is a query-based black-box method that generates adversarial examples through input-output queries, differing fundamentally from the gradient-based methods evaluated above. We therefore evaluate the adversarial defense methods under Square attack with  $\epsilon = 0.3$  and a query budget of 3000, monitoring the accuracy trajectory throughout the querying process. As before, 10,000 correctly classified samples are randomly selected per model for the attack test set. The results are depicted in Figure 5; Figure 5a–c show the accuracy evolution for the three student models.



**Figure 5.** Defense performance under Square black-box attack. (a) Square attack on 1D-CNN. (b) Square attack on MobileNet. (c) Square attack on ResNet8.

Figure 5 shows the accuracy of each method under the black-box Square attack as the query budget increases to 3000. The undefended Original model degrades steeply and is driven down to near-zero accuracy, confirming that the black-box attack is effective against an unprotected classifier. In sharp contrast, all defended models—and ADD in particular—remain almost flat at high accuracy throughout the query budget, with only marginal degradation. ADD attains the best black-box robustness on all three students: it retains over 90% accuracy on the 1D-CNN and MobileNet students, where its lead over the other defenses is largest, and remains the most robust on the compact ResNet8 student as well.

This black-box evaluation also serves as a sanity check against gradient masking. A defense whose apparent robustness stems from obfuscated or masked gradients is characteristically broken more easily by gradient-free attacks than by gradient-based ones, since the former do not rely on the masked gradient signal. We observe the opposite: the gradient-free Square attack leaves the defended models near their clean accuracy, whereas the gradient-based white-box attacks of Sections 5.3.1–5.3.2 reduce their accuracy substantially. The fact that gradient-based attacks are strictly stronger than the gradient-free attack indicates that the gradients of the ADD-trained models are genuinely informative for the adversary, and hence that the robustness of ADD reflects a real improvement of the decision boundary rather than gradient masking. This conclusion is reinforced by the white-box results themselves, in which the strongest attack is the AutoAttack ensemble—whose standard configuration already incorporates a black-box (Square) component—exactly as expected for a non-obfuscated model.

#### 5.3.4. Classification Performance on Clean Samples

The preceding subsections established the robustness advantage of ADD under both white-box and black-box attacks. A central motivation of ADD, however, is to enhance robustness without sacrificing clean-sample accuracy, thereby alleviating the accuracy–robustness tradeoff. To verify this, we evaluate the clean (unattacked) classification accuracy of the three student models trained with each method; the results are reported in Table 7.

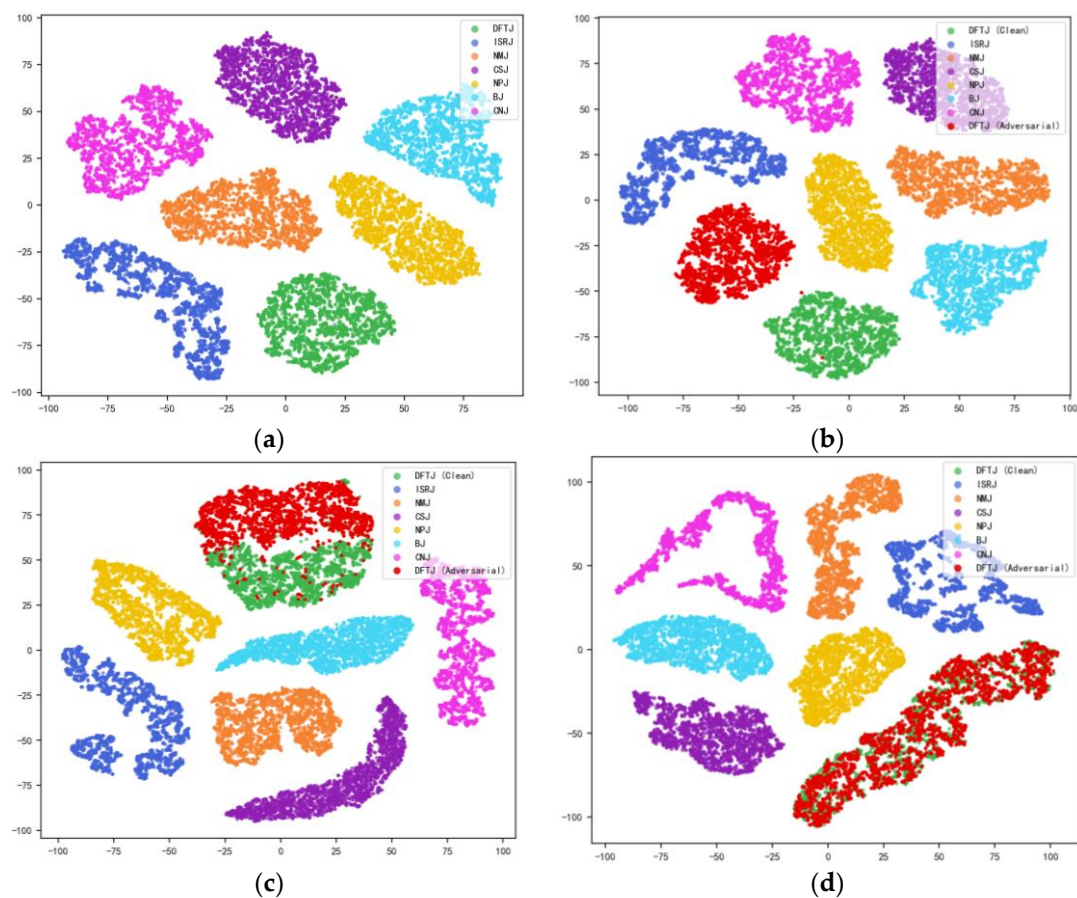
**Table 7.** Clean classification accuracy of the three student models under different defense methods.

| Method                | Clean Accuracy (%) |           |         |
|-----------------------|--------------------|-----------|---------|
|                       | 1D-CNN             | MobileNet | ResNet8 |
| Original (no defense) | 100                | 100       | 100     |
| PAT                   | 98.4               | 98.1      | 99.0    |
| DD                    | 100                | 99.5      | 100     |
| TRADES                | 99.2               | 99.5      | 99.7    |
| ADD                   | 100                | 100       | 100     |

As shown in Table 6, standard training (Original) and ADD both attain 100% clean accuracy on all three architectures. In contrast, standard adversarial training (PAT) reduces clean accuracy to 98.1–99.0%, and TRADES to 99.2–99.7%, reflecting the clean-accuracy cost typically incurred by adversarial training. Defensive distillation (DD) largely preserves clean accuracy (99.5–100%) but, as shown in Sections 5.3.1–5.3.3, offers little robustness against adversarial attacks. ADD is therefore the only defense that fully preserves the clean accuracy of standard training across all three students, while simultaneously achieving the highest robust accuracy. These results confirm that ADD enhances adversarial robustness without an accuracy penalty: its clean-sample pathway—comprising the clean-sample classification and clean-sample distillation losses—preserves clean accuracy, in contrast to standard adversarial training, which trades clean accuracy for robustness. ADD thereby alleviates the accuracy–robustness tradeoff.

#### 5.4. Feature-Space Visualization

To further elucidate the advantages of the ADD framework over other adversarial defense models (with PAT as the representative baseline), we visualize the output features of each sample using t-distributed stochastic neighbor embedding (t-SNE) [40], projecting them into a two-dimensional space. The 1D-CNN model is used for this analysis. For visual clarity, we select test samples with JNR  $\geq 10$  dB that are correctly classified by the normally trained model. In the experiment, PGD attacks with  $\varepsilon = 0.15$  are applied exclusively to the DFTJ class, while samples from all other classes remain unperturbed. This is an illustrative single-class setting; comprehensive multi-class robustness results are reported in Sections 5.2 and 5.3. The results are shown in Figure 6.



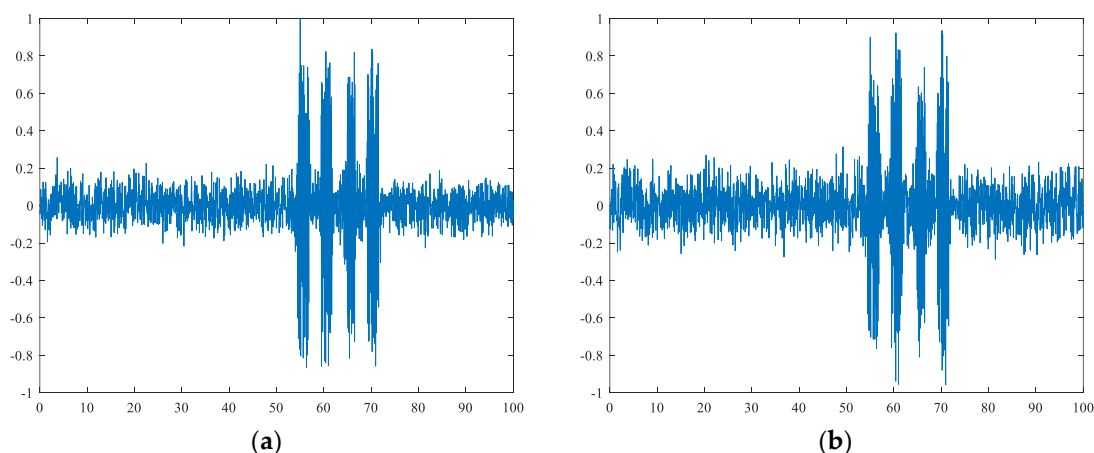
**Figure 6.** Feature-space visualization via t-SNE. (a) Original. (b) Original\_adv. (c) PAT. (d) ADD.

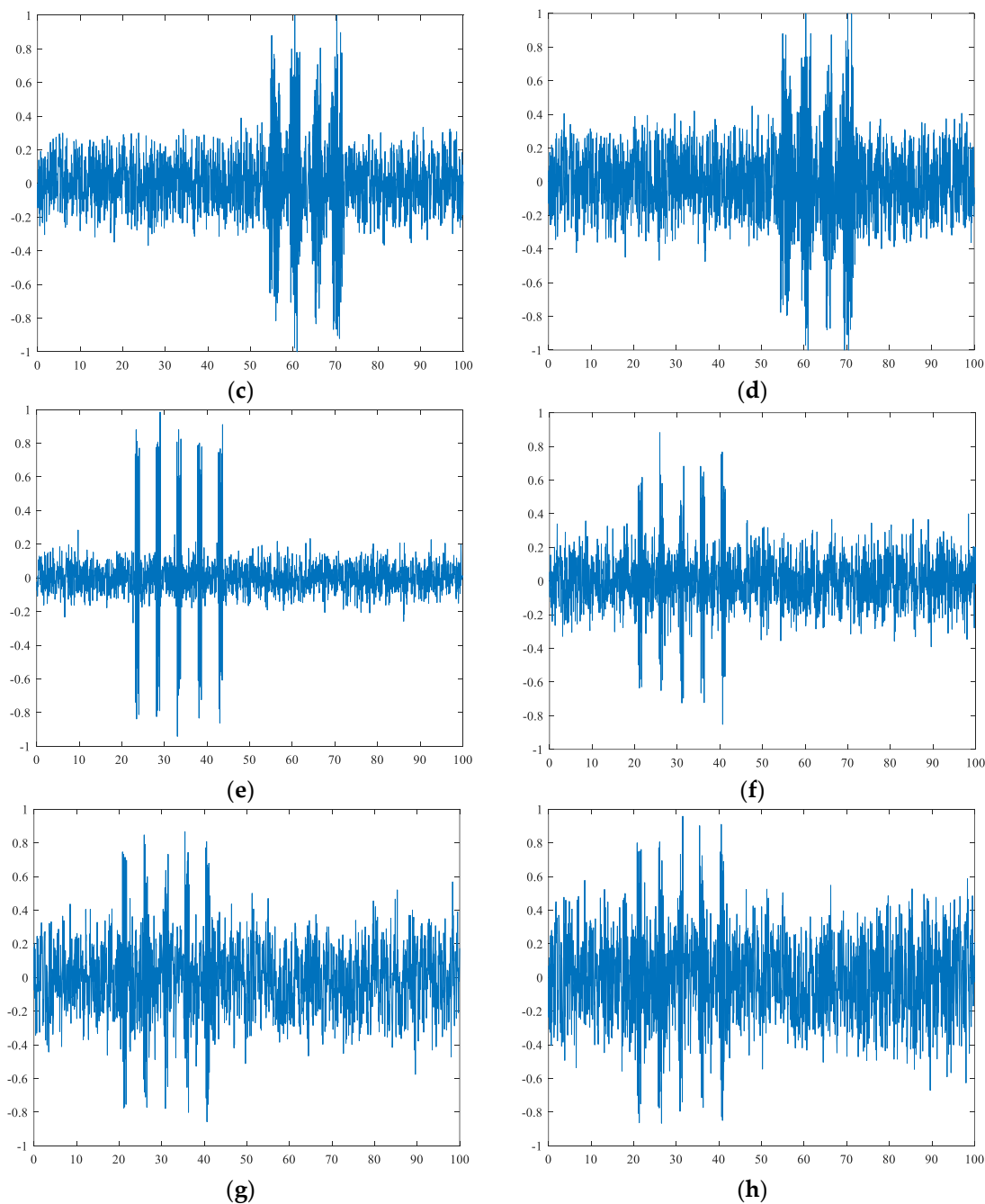
Figure 6a shows all original samples input to the standard model; Figure 6b superimposes the attacked DFTJ samples (marked in red) onto the standard model's original feature space; Figure 6c presents the attacked DFTJ samples alongside original samples for the PAT-trained model; and Figure 6d shows the corresponding visualization for the ADD-trained model. It is evident that the PGD attack disrupts the original distributional characteristics of DFTJ samples, causing the attacked samples to form a separate cluster. While PAT mitigates this effect to some extent, a substantial number of adversarial examples remain segregated as an independent cluster. In contrast, under the ADD framework, the majority of adversarial examples are co-located with the original samples within the same cluster, demonstrating markedly stronger defense capability against adversarial attacks.

### 5.5. Perturbation Covertness Analysis

A meaningful adversarial attack should produce perturbations that are small enough to remain covert—statistically masked by the receiver noise—while still achieving significant attack effectiveness. When conducting adversarial attacks on ADD-trained models, the magnitude of the generated perturbation is governed by  $\epsilon$ ; however, a more intuitive metric from the signal perspective is desirable. To distinguish adversarial perturbations from random noise, we adopt two supplementary metrics as proposed in [41]: the perturbation-to-noise ratio (PNR) and the perturbation-to-signal ratio (PSR). PNR is the ratio of perturbation power to noise power, and PSR is the ratio of perturbation power to signal power, with the relationship  $\text{PNR} = \text{PSR} \times \text{SNR}$  (or in decibels:  $\text{PNR}[\text{dB}] = \text{PSR}[\text{dB}] + \text{SNR}[\text{dB}]$ ). In this work, we define PJR as the ratio of perturbation power to jamming signal power, yielding  $\text{PNR}[\text{dB}] = \text{PJR}[\text{dB}] + \text{JNR}[\text{dB}]$  in the logarithmic domain.

Figure 7 presents illustrative examples of adversarial attacks with varying perturbation levels and their corresponding PNR values. Figure 7a–d show the real-part time-domain waveforms of an ISRJ signal ( $\text{JNR} = 10 \text{ dB}$ ) before and after PGD attack (1D-CNN model) at  $\epsilon = 0.1, 0.2$ , and  $0.3$ , respectively. Figure 7e–h provide analogous results for a NPJ signal. Following Sadeghi and Larsson [41], we regard a perturbation as covert—i.e., statistically undetectable by power-based means—when its PNR does not exceed unity ( $\text{PNR} \leq 0 \text{ dB}$ ), that is, when its power is at or below the noise floor and is masked by the ambient noise, producing no distinguishable power signature at the receiver. We emphasize that this is a power/detectability criterion defined relative to the receiver noise, not a perceptual threshold borrowed from the image domain. For the  $10 \text{ dB}$  JNR jamming signals under PGD attack, the perturbation power stays at or below the noise level ( $\text{PNR} \leq 0 \text{ dB}$ ) for  $\epsilon$  up to approximately  $0.3$ , confirming that the perturbations used in our experiments remain at or below the noise floor and are therefore difficult to separate from natural noise by power-based detection.





**Figure 7.** Examples of jamming signals after adversarial attack with varying perturbation levels. (a) ISRJ: No Attack, JNR = 10 dB. (b) ISRJ:  $\varepsilon = 0.1$ , PNR = −8.438 dB. (c) ISRJ:  $\varepsilon = 0.2$ , PNR = −3.192 dB. (d) ISRJ:  $\varepsilon = 0.3$ , PNR = −0.239 dB. (e) NPJ: No Attack, JNR = 10 dB. (f) NPJ:  $\varepsilon = 0.1$ , PNR = −7.934 dB. (g) NPJ:  $\varepsilon = 0.2$ , PNR = −2.867 dB. (h) NPJ:  $\varepsilon = 0.3$ , PNR = −0.106 dB.

## 6. Conclusions

To counter the adversarial attack threat facing radar jamming recognition models, this paper proposes a novel training framework—Adversarial Distillation Defense (ADD). Grounded in knowledge distillation theory, ADD organically integrates adversarial training by constructing a teacher model endowed with adversarial example priors and leveraging it to guide the student model toward learning more robust feature representations during adversarial training. Systematic experimental evaluations are conducted on multiple representative radar jamming recognition architectures using a simulated dataset under diverse adversarial attack scenarios. The results demonstrate that the ADD framework significantly enhances model robustness under adversarial conditions and yields

clear advantages over existing mainstream defense methods. ADD therefore provides an effective strategy for building lightweight and adversarially robust radar jamming recognition models. We note, however, that the present evaluation is based on a simulated dataset and does not yet account for real-world hardware and channel effects; validating ADD on measured radar data under realistic operational conditions is an important direction for future work and a key step toward its practical deployment.

**Author Contributions:** Conceptualization, Y.P. and X.H.; methodology, Y.P. and X.H.; software, X.H.; validation, X.H.; formal analysis, Y.G.; investigation, Y.P. and X.H.; resources, J.G.; writing—original draft preparation, Y.P. and X.H.; writing—review and editing, Y.P., X.H., Y.G., W.F., H.L. and J.G.; visualization, Y.P.; supervision, Y.P.; project administration, C.F.; funding acquisition, X.H. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by the Innovation Capability Support Program of Shaanxi under Grant No. 2025ZC-KJXX-81, and the Research Program Project of Youth Innovation Team of Shaanxi Provincial Education Department under Grant No. 24JP221.

**Data Availability Statement:** The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

**Acknowledgments:** The authors would like to thank an editor who handled the manuscript and the anonymous reviewers for their valuable comments and suggestions. The authors also acknowledge the support from the Youth Innovation Team of Shaanxi Universities.

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

## References

1. Hao, G.; Bu, L.; Lu, M.; Liu, H.; Liu, G.; Guo, J. Radar jamming signal recognition algorithm based on multi-feature fusion. *Digit. Signal Process.* **2025**, *158*, 104950.
2. Wang, Z.; Guo, Z.; Shu, G.; Li, N. Radar jamming recognition: Models, methods, and prospects. *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.* **2024**, *18*, 3315–3343.
3. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444.
4. Li, P.; Yang, J.; Lin, J. Radar compound jamming recognition based on image segmentation and fused attention residual network. *Sensors* **2025**, *25*, 2124.
5. Akhtar, N.; Mian, A. Threat of adversarial attacks on deep learning in computer vision: A survey. *IEEE Access* **2018**, *6*, 14410–14430.
6. Kokalj-Filipovic, S.; Miller, R.; Vanhoy, G. Adversarial examples in RF deep learning: Detection and physical robustness. In Proceedings of the 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP), Ottawa, ON, Canada, 11–14 November 2019; pp. 1–5.
7. Xu, D.; Yang, H.; Gu, C.; Chen, Z.; Xuan, Q.; Yang, X. Adversarial examples detection of radio signals based on multifeature fusion. *IEEE Trans. Circuits Syst. II Express Briefs* **2021**, *68*, 3607–3611.
8. Madry, A.; Makelov, A.; Schmidt, L.; Tsipras, D.; Vladu, A. Towards deep learning models resistant to adversarial attacks. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
9. Zi, B.; Zhao, S.; Ma, X.; Jiang, Y.-G. Revisiting adversarial robustness distillation: Robust soft labels make student better. In Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), Montreal, QC, Canada, 10–17 October 2021; pp. 16443–16452.
10. Goldblum, M.; Fowl, L.; Feizi, S.; Goldstein, T. Adversarially robust distillation. In Proceedings of the AAAI Conference on Artificial Intelligence (AAAI), New York, NY, USA, 7–12 February 2020; pp. 3996–4003.
11. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.
12. Shao, G.; Chen, Y.; Wei, Y. Convolutional neural network-based radar jamming signal classification with sufficient and limited samples. *IEEE Access* **2020**, *8*, 80588–80598.
13. Shao, G.; Chen, Y.; Wei, Y. Deep fusion for radar jamming signal classification based on CNN. *IEEE Access* **2020**, *8*, 117236–117244.
14. Zhou, H.; Wang, L.; Guo, Z. Recognition of radar compound jamming based on convolutional neural network. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, *59*, 7380–7394.

15. Zhang, Y.; Zhao, Z.; Bu, Y. Radar active jamming recognition under open world setting. *Remote Sens.* **2023**, *15*, 4107.
16. Moosavi-Dezfooli, S.-M.; Fawzi, A.; Frossard, P. Deepfool: A simple and accurate method to fool deep neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 2574–2582.
17. Brendel, W.; Rauber, J.; Bethge, M. Decision-based adversarial attacks: Reliable attacks against black-box machine learning models. In Proceedings of the International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
18. Goodfellow, I.J.; Shlens, J.; Szegedy, C. Explaining and harnessing adversarial examples. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
19. Dong, Y.; Liao, F.; Pang, T.; Su, H.; Zhu, J.; Hu, X.; Li, J. Boosting adversarial attacks with momentum. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–22 June 2018; pp. 9185–9193.
20. Huang, T.; Chen, Y.; Yao, B.; Yang, B.; Wang, X.; Li, Y. Adversarial attacks on deep-learning-based radar range profile target recognition. *Inf. Sci.* **2020**, *531*, 159–176.
21. Gao, X.; Zhang, Z.; Liu, M.; Gong, Z.; Li, X. Intelligent radar image recognition countermeasures: A review. *J. Radars* **2023**, *12*, 696–712.
22. Zhang, R.; Li, Y.; Liu, J. Transferable anti-intelligence recognition radar waveform design based on adversarial attacks. *IEEE Trans. Aerosp. Electron. Syst.* **2024**, *61*, 3798–3812.
23. Zhang, L.; Lambotharan, S.; Zheng, G.; Liao, G.; Demontis, A.; Roli, F. A hybrid training-time and run-time defense against adversarial attacks in modulation classification. *IEEE Wirel. Commun. Lett.* **2022**, *11*, 1161–1165.
24. Zhang, H.; Yu, Y.; Jiao, J.; Xing, E.; El Ghaoui, L.; Jordan, M. Theoretically principled trade-off between robustness and accuracy. In Proceedings of the International Conference on Machine Learning (ICML), Long Beach, CA, USA, 9–15 June 2019; pp. 7472–7482.
25. Papernot, N.; McDaniel, P.; Wu, X.; Jha, S.; Swami, A. Distillation as a defense to adversarial perturbations against deep neural networks. In Proceedings of the IEEE Symposium on Security and Privacy (SP), San Jose, CA, USA, 22–26 May 2016; pp. 582–597.
26. Romero, A.; Ballas, N.; Kahou, S.E.; Chassang, A.; Gatta, C.; Bengio, Y. FitNets: Hints for thin deep nets. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015.
27. Zagoruyko, S.; Komodakis, N. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. In Proceedings of the International Conference on Learning Representations (ICLR), Toulon, France, 24–26 April 2017.
28. Gou, J.; Yu, B.; Maybank, S.J.; Tao, D. Knowledge distillation: A survey. *Int. J. Comput. Vis.* **2021**, *129*, 1789–1819.
29. Zhang, Z.; Gao, X.; Liu, S.; Peng, B.; Wang, Y. Energy-based adversarial example detection for SAR images. *Remote Sens.* **2022**, *14*, 5168.
30. Li, P.; Hu, X.; Feng, C.; Shi, X.; Guo, Y.; Feng, W. SAR-AD-BagNet: An interpretable model for SAR image recognition based on adversarial defense. *IEEE Geosci. Remote Sens. Lett.* **2022**, *20*, 4000505.
31. Lv, Q.; Quan, Y.; Feng, W.; Sha, M.; Dong, S.; Xing, M. Radar deception jamming recognition based on weighted ensemble CNN with transfer learning. *IEEE Trans. Geosci. Remote Sens.* **2021**, *60*, 5107511.
32. Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I.; Fergus, R. Intriguing properties of neural networks. In Proceedings of the International Conference on Learning Representations (ICLR), Banff, AB, Canada, 14–16 April 2014.
33. Zhou, J.; Peng, B.; Peng, B. Adversarial attacks on radar target recognition based on deep learning. In Proceedings of the IEEE International Geoscience and Remote Sensing Symposium (IGARSS), Kuala Lumpur, Malaysia, 17–22 July 2022; pp. 2646–2649.
34. Maroto, J.; Ortiz-Jiménez, G.; Frossard, P. On the benefits of knowledge distillation for adversarial robustness. *arXiv* **2022**, arXiv:2203.07159.
35. 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 (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
36. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861.
37. Croce, F.; Hein, M. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In Proceedings of the International Conference on Machine Learning (ICML), Virtual, 13–18 July 2020; pp. 2206–2216.
38. Andriushchenko, M.; Croce, F.; Flammarion, N.; Hein, M. Square attack: A query-efficient black-box adversarial attack via random search. In Proceedings of the European Conference on Computer Vision (ECCV), Glasgow, UK, 23–28 August 2020; pp. 484–501.
39. Kim, T.; Oh, J.; Kim, N.; Cho, S.; Yun, S.-Y. Comparing Kullback–Leibler divergence and mean squared error loss in knowledge distillation. In Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), Montreal, QC, Canada, 19–27 August 2021; pp. 2628–2635.

40. Van der Maaten, L.; Hinton, G. Visualizing data using t-SNE. *J. Mach. Learn. Res.* **2008**, *9*, 2579–2605.
41. Sadeghi, M.; Larsson, E.G. Adversarial attacks on deep-learning based radio signal classification. *IEEE Wirel. Commun. Lett.* **2018**, *8*, 213–216.

**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.