A Deep Ensemble-based Wireless Receiver Architecture for Mitigating Adversarial Attacks in Automatic Modulation Classification

Rajeev Sahay, *Student Member, IEEE,* Christopher G. Brinton,
Senior Member, IEEE, and David J. Love, *Fellow, IEEE*

Abstract—Deep learning-based automatic modulation classification (AMC) models are susceptible to adversarial attacks. Such attacks inject specifically crafted wireless interference into transmitted signals to induce erroneous classification predictions. Furthermore, adversarial interference is transferable in black box environments, allowing an adversary to attack multiple deep learning models with a single perturbation crafted for a particular classification model. In this work, we propose a novel wireless receiver architecture to mitigate the effects of adversarial interference in various black box attack environments. We begin by evaluating the architecture uncertainty environment, where we show that adversarial attacks crafted to fool specific AMC DL architectures are not directly transferable to different DL architectures. Next, we consider the domain uncertainty environment, where we show that adversarial attacks crafted on time domain and frequency domain features to not directly transfer to the altering domain. Using these insights, we develop our Assorted Deep Ensemble (ADE) defense, which is an ensemble of deep learning architectures trained on time and frequency domain representations of received signals. Through evaluation on two wireless signal datasets under different sources of uncertainty, we demonstrate that our ADE obtains substantial improvements in AMC classification performance compared with baseline defenses across different adversarial attacks and potencies.

Index Terms—Adversarial attacks, automatic modulation classification, machine learning in communications, wireless security

I. INTRODUCTION

THE recent exponential growth of wireless traffic has resulted in a crowded radio spectrum, which, among other factors, has contributed to reduced mobile efficiency. With the number of devices requiring wireless resources projected to continue increasing, this inefficiency is expected to present large-scale challenges in wireless communications. Automatic modulation classification (AMC), which is a part of cognitive radio technologies, aims to alleviate the inefficiency induced in shared spectrum environments by dynamically extracting meaningful information from massive streams of wireless data. Traditional AMC methods are based on maximum-likelihood

approaches [2]–[6], which consist of deriving statistical decision boundaries using hand-crafted (i.e., manually-defined) features to discern various modulation constellations. More recently, deep learning (DL) has become a popular alternative to maximum-likelihood methods for AMC, since it does not require manual feature engineering to attain high classification performance [7]–[11].

Despite their ability to obtain strong AMC performance, however, deep learning models are highly susceptible to adversarial evasion attacks [12]–[16], which introduce additive wireless interference into radio frequency (RF) signals to induce erroneous behavior on well-trained spectrum sensing models. Adversarial interference signals are specifically crafted to alter the classification decisions of trained DL models using a minimum, and often undetectable, amount of power. Not only do adversarial interference signals inhibit privacy and security in cognitive radios, but they are also more efficient than traditional jamming attacks applied in communication networks [17]. As a result, wireless adversarial interference presents high-risk challenges for the deployment of deep learning models in autonomous signal classification receivers [18].

Adversarial attacks can vary in potency depending on the amount of system knowledge available to the attacker. The most effective attack is crafted in a *white box* threat model, where the adversary has knowledge of both the signal features used for classification as well as the underlying classification model (including its hyper-parameter settings and values). Not only is the availability of such specific information to an adversary rare in the context of wireless communications [19], but it is also not necessary to craft an effective adversarial interference signal. This is due to the transferability property of adversarial attacks between classification models [20]–[22], in that an attack crafted to fool a specific DL classifier can significantly degrade performance on a disparate model trained to perform the same task. Such *black box* attacks are more realistic to consider than *white box* attacks in real-world communication channels, where an adversary operates with limited system information. As a result of the transferability property of adversarial attacks, an adversary can induce large-scale AMC performance degradation, thus reducing spectrum efficiency and compromising secure communication channels.

In this work, we develop a novel wireless receiver architecture capable of mitigating the effects of transferable AMC adversarial interference injected into transmitted signals in black

box environments. We train a series of deep learning-based AMC models, which utilize two different representations of wireless signals as input features: (i) in-phase and quadrature (IQ) time domain signals, and (ii) frequency domain signals. Our methodology incorporates two key insights from this analysis. First, we find that *attacks on DL-based AMC models are not directly transferable between signal domains* (i.e., from the time domain to the frequency domain, and vice versa). Second, we find that *the transferability property varies substantially between DL architectures*. Based on these insights, we propose an ensemble AMC classification methodology that utilizes different signal representations and DL models, which we find significantly mitigates the effects of additive black box adversarial interference instantiated on both time domain and frequency domain feature sets.

Outline and Summary of Contributions: Compared to related work in AMC (Sec. II), we make the following contributions:

1) **Cross-domain signal receiver architecture for AMC** (Sec. III-A, III-B, III-C, and IV-B): We develop a robust AMC module consisting of both IQ-based and frequency-based deep learning models. We show that these models obtain high classification accuracy on two real-world datasets.

2) **Resilience to transferable adversarial attacks between classification architectures** (Sec. III-D and Sec IV-C): We demonstrate our receiver's ability to withstand transferable adversarial interference between deep learning classification architectures trained on the same input signal representation.

3) **Resilience to transferable adversarial interference across domains** (Sec. III-E and Sec. IV-D): For a given type of DL classification architecture, we demonstrate the resilience of frequency domain trained classifiers to time domain instantiated attacks, and vice versa. This analysis leads to the identification of the most robust deep learning architectures suitable for withstanding transferable adversarial attacks.

4) **Black box adversarial interference mitigation via deep ensemble** (Sec. III-F and Sec. IV-E): Using the foregoing properties, we develop a deep ensemble consisting of both time domain and frequency domain-based AMC classifiers trained on a variety of architectures. Our experiments show that this ensemble effectively mitigates evasion attacks regardless of the signal features or classification architecture targeted by the adversary.

II. Related Work

Deep learning has been widely proposed for AMC as it requires little to no feature selection to attain high classification performance on IQ samples. In particular, several studies have demonstrated the success of convolutional neural networks (CNNs) for AMC [8], [9], [23]–[26] using network graphs such as AlexNet [27] and ResNet [28]. In addition to CNNs, recurrent neural networks (RNNs) have also been shown to provide high AMC accuracy [29]–[31]. In this work, we build upon the success of prior AMC deep learning by proposing

a series of models consisting of convolutional, recurrent, both convolutional and recurrent, and dense fully connected layers to construct the classifiers contained in our wireless signal receiver. Moreover, building on prior work [32], [33], we consider how varying the signal domain representation (time or frequency) of the input signal can impact AMC.

Although the susceptibility of deep learning AMC classifiers to evasion attacks has been demonstrated in prior work [12]–[15], [34], relatively few defenses have been proposed to detect [35] or mitigate the effects of adversarial interference [34], [36]. The defense algorithms which have been proposed for AMC DL classifiers – adversarial training [37], [38], Gaussian smoothing [39], [40], and autoencoder pre-training [41] – have each demonstrated degraded performance in black box environments. This is largely due to these defenses being specifically designed to defend white box attacks and being directly adopted in black box environments without special consideration being given to the differing threat model. Our proposed wireless receiver architecture, on the other hand, is designed with the intention of mitigating black box adversarial interference attacks under different knowledge levels of the adversary such as DL architecture uncertainty and classification domain uncertainty. In this regard, and to the best of our knowledge, no work has explored the extent to which adversarial attacks in AMC are transferable between signal domains (although various domains for classification have been investigated [32], [33]).

Contrary to the limited defenses that exist for defending black box AMC adversarial interference, several defenses have been proposed for defending deep learning image classifiers from black box adversarial attacks, with no method generally accepted as a robust solution [42]. Nonetheless, even considering the adoption of image classification defenses for AMC is difficult due to the differing constraints placed on the adversary in each setting (e.g., channel effects and transmit power budget in AMC versus visual perceptibly and targeted pixel attacks in image classification). Therefore, in this work, we develop an ensemble defense specifically tailored to defend AMC models from adversarial attacks when the adversary is constrained by communications-based limitations. Future work may consider the adaptation of our proposed method for AMC in the image classification setting.

III. Our AMC Methodology

In this section, we outline the wireless channel input-output model we consider for AMC as well as our proposed defense mechanisms to mitigate black box adversarial interference. We begin by describing our system model (Sec. III-A and III-B), followed by the DL classifiers that we consider for AMC (Sec. III-C). Then, we characterize the adversary's attack strategy (Sec. III-D) and define transferability relative to model architectures and signal domains (Sec. III-E). Finally, we present our deep ensemble defense for robustness against black box attacks (Sec. III-F). An overview of our methodology is given in Fig. 1.
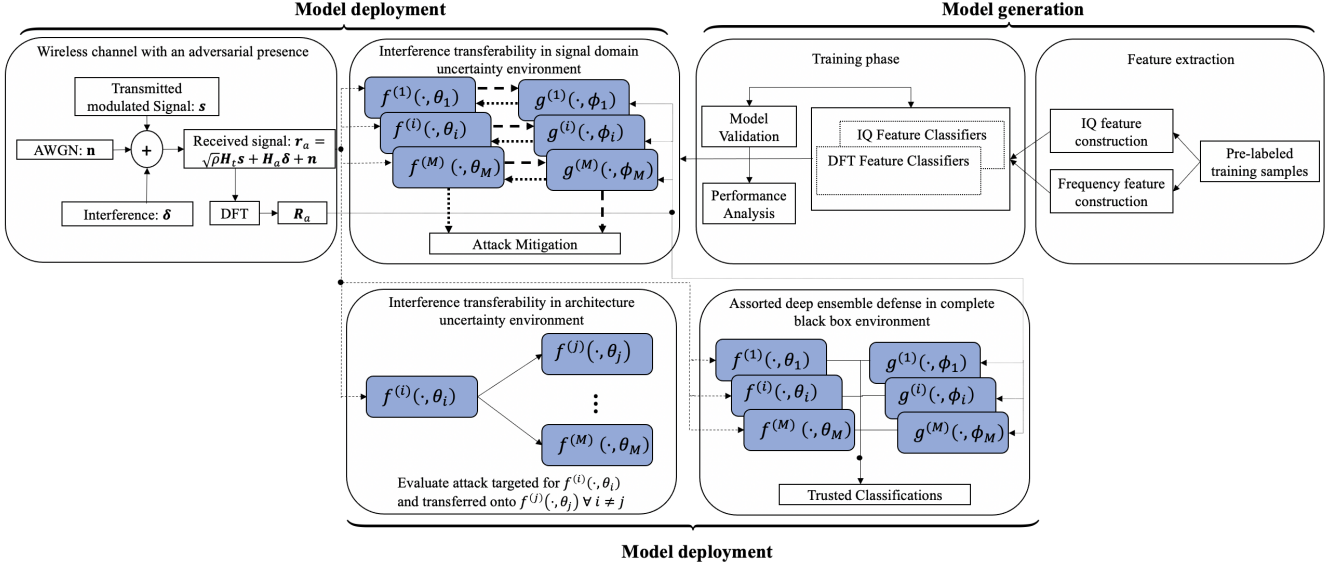
Fig. 1: Our AMC system model with adversarial interference. The shaded blocks correspond to our constructed models deployed in the wireless receiver, and the solid and dotted lines correspond to the adversarial frequency feature-based signal and adversarial IQ feature-based signal, respectively. The models that exhibit the highest degree of attack mitigation in both the signal domain uncertainty environment and the architecture uncertainty environment are utilized in the construction of the assorted deep ensemble (ADE) defense, whose outputs are aggregated to calculate the predicted constellation of the input signal.

A. Signal and Channel Modeling

We consider a wireless channel consisting of a transmitter, which is aiming to send a modulated signal, and a receiver, whose objective is to perform AMC on the received waveform and realize its modulation constellation. In addition, we consider an adversary aiming to inject interference into the transmitted signal to induce misclassification at the receiver. We will denote the channel from the transmitter to the receiver as $\mathbf{h}_t \in \mathbb{C}^\ell$ and the channel from the adversary to the receiver as $\mathbf{h}_a \in \mathbb{C}^\ell$, where $\mathbf{h}_t = [h_t[0], \ldots, h_t[\ell-1]]^T$, $\mathbf{h}_a = [h_a[0], \ldots, h_a[\ell-1]]^T$, and ℓ denotes the length of the received signal's observation window. Both $\mathbf{h}_t$ and $\mathbf{h}_a$ also include radio imperfections such as sample rate offset (SRO), center frequency offset (CFO), and selective fading, none of which are known to the receiver. We further assume that the receiver has no knowledge of the channel model or its distribution. Therefore, the channel model is not directly utilized in the development of our methodology. However, this general setting motivates an AMC solution using a data-driven approach, as we consider in this work, in which the true modulation constellation of the received signal is estimated from a model trained on a collection of pre-existing labeled signals, which capture the effects of the considered wireless channel.

At the transmitter, we denote the transmitted signal as $\mathbf{s} = [s[0], \ldots, s[\ell-1]]$, which is modulated using one of $C = |\mathcal{S}|$ modulation constellations chosen from a set, $\mathcal{S}$, of possible modulation schemes, with each scheme having equal probability of selection. At the receiver, the collected waveform is modeled by

$$\mathbf{r}_t = \sqrt{\rho}\,\mathbf{H}_t\mathbf{s} + \mathbf{n} \tag{1}$$

when the adversary does not instantiate an attack and as

$$\mathbf{r}_a = \mathbf{r}_t + \mathbf{H}_a\boldsymbol{\delta} = \sqrt{\rho}\,\mathbf{H}_t\mathbf{s} + \mathbf{H}_a\boldsymbol{\delta} + \mathbf{n} \tag{2}$$

when the adversary launches an adversarial interference signal, which is denoted by $\boldsymbol{\delta} \in \mathbb{C}^\ell$, whose potency (i.e., effectiveness in inducing misclassification) is dependent on the adversary's power budget, denoted as P_T. In both (1) and (2), $\mathbf{r}_t = [r_t[0], \ldots, r_t[\ell-1]]^T$ and $\mathbf{r}_a = [r_a[0], \ldots, r_a[\ell-1]]^T$ denote the received signal in the absence and presence of adversarial interference, respectively, $\mathbf{H}_t = \mathrm{diag}\{h_t[0], \ldots, h_t[\ell-1]\} \in \mathbb{C}^{\ell \times \ell}$, $\mathbf{H}_a = \mathrm{diag}\{h_a[0], \ldots, h_a[\ell-1]\} \in \mathbb{C}^{\ell \times \ell}$, $\mathbf{n} \in \mathbb{C}^\ell$ represents complex additive white Gaussian noise (AWGN) distributed as $\mathcal{CN}(0, 1)$, and ρ denotes the signal to noise ratio (SNR), which is known at the receiver. Note that although $\mathbf{r}_t = \mathbf{r}_a$ when $\boldsymbol{\delta} = 0$, we define both signals separately to characterize the construction of $\boldsymbol{\delta}$ throughout this work.

B. Signal Domain Transform

At the receiver, we model $\mathbf{r}_t = [r_t[0], \ldots, r_t[\ell-1]]^T$ using both (i) its in-phase and quadrature (IQ) components in the time domain and (ii) its frequency components obtained from the discrete Fourier transform (DFT) of $\mathbf{r}_t$. Specifically, the pth component of the DFT of $\mathbf{r}_t$ is given by

$$R_t[p] = \sum_{k=0}^{\ell-1} r_t[k]e^{-\frac{j2\pi}{\ell}pk}, \quad p = 0, \ldots, \ell-1, \tag{3}$$

where $\mathbf{R}_t = [R_t[0], \ldots, R_t[\ell-1]]^T$ contains all frequency components of $\mathbf{r}_t$. Here, we are interested in comparing the effects of $\boldsymbol{\delta}$ when an attack is instantiated on an AMC model trained on one domain (i.e., $\mathbf{r}_a$ or $\mathbf{R}_a$) and transferred to an AMC model trained on the other (i.e., $\mathbf{R}_a$ or $\mathbf{r}_a$, respectively).

Although both signal representations are complex (i.e., $\mathbf{r}_t, \mathbf{R}_t \in \mathbb{C}^\ell$), we represent all signals as two-dimensional

reals, using the real and imaginary components for the first and second dimension, respectively, in order to utilize all signal components during classification. Thus, we represent all time and frequency domain features as real-valued matrices (i.e., $\mathbf{r}_t, \mathbf{R}_t \in \mathbb{R}^{\ell \times 2}$).

C. Deep Learning Architectures

At the receiver, we consider four distinct deep learning architectures for AMC. Each considered classifier is trained on a set of modulated data signals, $\mathcal{X} \subset \mathbb{R}^{\ell \times 2}$, where each input, $\mathbf{r}_t \in \mathcal{X}$, belongs to one of C modulation constellations and is constructed using time domain IQ features. In general, we denote a time domain trained deep learning classifier, parameterized by θ_i, as $f^{(i)}(\cdot, \theta_i) : \mathcal{X} \to \mathbb{R}^C$, where $f^{(i)}(\cdot, \theta_i)$ for $i = 1, \ldots, 4$ refers to one of four deep learning architectures used to construct the classifier along with its corresponding parameters θ_i. The trained classifier assigns each input $\mathbf{r}_t \in \mathcal{X}$ a label denoted by $\hat{\mathcal{C}}(\mathbf{r}_t, \theta_i) = \text{argmax}_k \, f_k^{(i)}(\mathbf{r}_t, \theta_i)$, where $f_k^{(i)}(\mathbf{r}_t, \theta_i)$ is the vector of predicted classification probabilities, assigned by the ith model, of $\mathbf{r}_t$ being modulated according to the kth constellation for $k = 1, \ldots, C$. Similarly, we denote the ith deep learning classifier trained using the DFT of the input signal, $\mathbf{R}_t$, parameterized by ϕ_i, as $g^{(i)}(\cdot, \phi_i) : \mathbb{R}^{\ell \times 2} \to \mathbb{R}^C$, which is trained to perform the same classification task as $f^{(i)}(\cdot, \theta_i)$ but using the frequency features of $\mathbf{r}_t$ to comprise the input signal. Note that θ_i denotes the parameters of a time domain trained classifier whereas ϕ_i denotes the parameters of a frequency domain trained classifier.

We analyze the classification performance and the efficacy of our proposed defense on four common AMC deep learning architectures: the fully connected neural network (FCNN), the convolutional neural network (CNN), the recurrent neural network (RNN), and the convolutional recurrent neural network (CRNN). For each model, we apply the ReLU non-linearity activation function in its hidden layers, given by $\nu(a) = \max\{0, a\}$, and a C-unit softmax activation function at the output layer given by

$$\nu(\mathbf{a})_k = \frac{e^{a_k}}{\sum_{j=1}^{C} e^{a_j}}, \quad (4)$$

where $k = 1, \ldots, C$ for input vector $\mathbf{a}$. This normalization allows a probabilistic interpretation of the model's output predictions.

FCNN: FCNNs consist of multiple layers, which are comprised of individual units. Each unit contains a set of trainable weights, whose dimensionality is equal to the number of units in the preceding layer, and the number of units in each layer is an adjustable hyper-parameter. The output of a single unit, u, in a particular layer is given by

$$\nu\left(\sum_i w_i^{(u)} a_i + b^{(u)} \right), \quad (5)$$

where $\nu(\cdot)$ is the activation function, $\mathbf{w} = [w_1^{(u)}, \ldots, w_n^{(u)}]$ is the weight vector for unit u estimated from the training data, $\mathbf{a} = [a_1, \ldots, a_n]$ is the vector containing the outputs from the previous layer, and $b^{(u)}$ is a threshold bias for unit

u. Our FCNN consists of three hidden layers with 256, 128, and 128 units, respectively, and each hidden layer applies a 20% dropout rate during training.

CNN: CNNs consist of one or more convolutional layers, which extract spatially correlated patterns from their inputs. Each convolutional layer is comprised of a set of $L \times W$ filters, denoted as $\mathbf{m} \in \mathbb{R}^{L \times W}$. The output of the pth convolutional unit in a particular layer (termed a feature map) is given by

$$y_p[j, k] = \nu\left(\sum_{l=0}^{L-1} \sum_{w=0}^{W-1} x[j+l, k+w] m[l^{(p)}, w^{(w)}] \right), \quad (6)$$

where the two dimensional input, $\mathbf{x}$, and the pth filter, with each kernel index denoted by $m[l^{(p)}, w^{(w)}]$, are cross-correlated and passed through an activation function, $\nu(\cdot)$, to produce the pth feature map, $\mathbf{y}$, indexed at j and k. Our CNN is comprised of two convolutional layers consisting of 256 2×5 and 64 1×3 feature maps (each with 20% dropout), respectively, and a 128-unit ReLU fully connected layer.

RNN: RNNs implement feedback layers, which extract temporally correlated patterns from their inputs. Long-short-term-memory (LSTM) cells [43] extend recurrence to create memory in neural networks by introducing three gates for learning. *Input gates* prevent irrelevant features from entering the recurrent layer while *forget gates* eliminate irrelevant features altogether. *Output gates* produce the LSTM layer output, which is inputted into the subsequent network layer. The gates are used to recursively calculate the internal state of the cell, denoted by $\mathbf{z}_c^{(t)}$ at time t for cell c, at a specific recursive iteration, called a time instance, which is then used to calculate the cell output given by

$$\mathbf{q}^{(t)} = \tanh(\mathbf{z}_c^{(t)}) \nu(\mathbf{p}^{(t)}), \quad (7)$$

where $\mathbf{p}^{(t)}$ is the parameter obtained from the output gate and $\nu(\cdot)$ is the logistic sigmoid function given by $\nu(p_i^{(t)}) = 1/(1 + e^{-p_i^{(t)}})$ for the ith element in $\mathbf{p}^{(t)}$. Our RNN is comprised of a 75-unit LSTM layer followed by a 128-unit ReLU fully connected layer.

CRNN: Lastly, we consider a CRNN, which captures both spatial (convolutional) and temporal (recurrent) correlations in the input sequence. Our CRNN is consists of two convolutional layers (containing 256 2×5 and 128 1×4 feature maps, respectively) followed by a 128-unit LSTM layer and a 64-unit ReLU fully connected layer.

Training Details: Each AMC classifier contained in the wireless receiver is trained using the Adam optimizer [44] with a batch size of 64 and a dynamic learning rate scheduler. Furthermore, an early stopping criterion with a patience of 50 epochs was used as the stopping condition for training on each model to achieve convergence (i.e., the model training is terminated when the loss on a validation set has not decreased in the last 50 successive epochs). Each model was trained using the categorical cross entropy cost function, which, for the time domain feature-based models, is given by

$$\mathcal{L}_n(\mathbf{r}_{t,n}, \mathbf{y}_n, \theta) = \sum_{j=1}^{C} y_j \log(\hat{y}_j) \quad (8)$$

for the nth raining sample $\mathbf{r}_{t,n}$, and

$$\mathcal{L} = -\frac{1}{N}\sum_{n=1}^{N}\mathcal{L}_n, \tag{9}$$

over the entire training set containing N samples. Note that N denotes the number of samples in the training set whereas ℓ denotes the length of the received signal's observation window. Here, $\mathbf{y}_n$ is the one-hot encoded label of the nth signal (i.e., $y_j = 1$ if the ground truth label of the sample is modulation class j and $y_j = 0$ otherwise), and $\hat{y}_j$ is the confidence assigned by the classifier, parameterized by θ, that the given input is modulated according to constellation j. The categorical cross entropy cost for the frequency domain feature-based models is calculated similarly to (8) and (9), but with $\mathbf{R}_{t,n}$ and ϕ replacing the $\mathbf{r}_{t,n}$ and θ, respectively, in (8).

D. Adversarial Interference

Adversarial interference (i.e., $\boldsymbol{\delta} \in \mathbb{R}^{\ell \times 2}$) is specifically crafted to induce erroneous modulation constellation predictions at the receiver. Several methods exist to craft adversarial interference signals and, therefore, several designs of $\boldsymbol{\delta}$ may effectively induce misclassification (i.e., disparate and unique constructions for $\boldsymbol{\delta}$ may fulfill an adversary's objective). For a general classifier trained on IQ features, adversarial interference is crafted by solving

$$\min_{\boldsymbol{\delta}} \quad ||\boldsymbol{\delta}||_2 \tag{10a}$$

$$\text{s. t.} \quad \hat{\mathcal{C}}(\mathbf{r}_t, \theta) \neq \hat{\mathcal{C}}(\mathbf{r}_t + \mathbf{H}_a\boldsymbol{\delta}, \theta), \tag{10b}$$

$$||\boldsymbol{\delta}||_2^2 \leq P_T, \tag{10c}$$

$$\mathbf{r}_t + \mathbf{H}_a\boldsymbol{\delta} \in \mathbb{R}^{\ell \times 2}, \tag{10d}$$

where $||\cdot||_2$ refers to the l_2 norm. The constraint given in (10b) attempts to induce misclassification with respect to the parameters of one particular targeted model (trained on time domain features) while simultaneously using the least amount of power possible to evade detection caused by higher powered adversarial interference [35], [45], thus restricting the power budget to P_T in (10c). Finally, (10d) ensures that the perturbed sample, $\mathbf{r}_a$, remains in the same space as $\mathbf{r}_t$. An adversary may also choose to instantiate an attack on a classifier trained on frequency features; in this case, the crafted perturbation is given by replacing $\mathbf{r}_t$ with $\mathbf{R}_t$ in (10) while utilizing the classifier parametertized by ϕ instead of θ.

Note that $\boldsymbol{\delta}$ can be constrained using other l_p norms such as $p = 0$, $p = 1$, or $p = \infty$, but $p = 2$ is a natural choice to consider in the domain of wireless communications as it directly corresponds to the perturbation power. Furthermore, the best solution to (10) is not necessarily realized when $||\boldsymbol{\delta}||_2$ is minimized or when $||\boldsymbol{\delta}||_2^2 = P_T$, as the primary objective of the adversary is to induce misclassification on $\mathbf{r}_a$. In addition, a solution to (10) is not always guaranteed to exist, and in such cases, the additive perturbation may not necessarily result in $\mathbf{r}_a$ being misclassified at the receiver.

In a real-world wireless communication channel, the adversary's knowledge for constructing an attack is limited, which prevents it from solving (10) directly. Our focus is on black box threat models in which the adversary has some or no knowledge about the classification method at the receiver. In this capacity, we consider three distinct knowledge levels for the adversary: (i) *architecture uncertainty*, where the adversary is aware of the features being used for classification (i.e., IQ vs DFT features) but unaware of the DL architecture used for classification; (ii) *signal domain uncertainty*, where the adversary is aware of the DL classification architecture but unaware of the features used to comprise the input signal for classification; and (iii) *overall uncertainty*, where the adversary is unaware of both the classification architecture and signal features used for classification at the receiver.

Furthermore, due to the black box threat model, the optimization in (10) is formulated as an *untargeted* adversarial attack, where the adversary aims to induce misclassification without regard to a particular incorrect constellation assigned at the receiver. Since prior work has shown that targeted adversarial examples rarely transfer with their targeted labels [20], our considered black box threat model would treat both targeted and untargeted attacks crafted on a surrogate model as untargeted attacks on the underlying target classifier. Therefore, we evaluate the resilience of our wireless receiver on untargeted attacks only.

In this work, we consider three methods for crafting adversarial perturbations: the gradient-based fast gradient sign method (FGSM) [46], the gradient-based basic iterative method (BIM) [47], and the optimization-based Carlini and Wagner (CW) l_2 attack [48]. Under the FGSM attack, the adversary exhausts its total power budget on a single step attack, whereas under the BIM and CW attacks, the adversary iteratively uses a fraction of its attack budget, resulting in a more potent attack, in comparison to the FGSM perturbation, at the cost of higher computational overhead. In the two considered gradient-based attacks, the cost of $\mathbf{r}_{a,n}$ is first linearized as $\mathcal{L}_n(\mathbf{r}_{t,n} + \boldsymbol{\delta}_n, \mathbf{y}_n, \theta) \approx \mathcal{L}_n(\mathbf{r}_{t,n}, \mathbf{y}_n, \theta) + (\mathbf{H}_a\boldsymbol{\delta})^T\mathcal{L}_n(\mathbf{r}_{t,n}, \mathbf{y}_n, \theta)$, which is maximized by setting $\mathbf{H}_a\boldsymbol{\delta} = \epsilon\mathcal{L}_n(\mathbf{r}_{t,n}, \mathbf{y}_n, \theta)$ where ϵ is a scaling factor to satisfy the adversary's power constraint. The optimization-based CW attack, on the other hand, optimizes (10a) subject to the constraint $\hat{\mathcal{C}}(\mathbf{r}_t + \mathbf{H}_a\boldsymbol{\delta}, \theta) = t$, where t is the second highest constellation prediction of $\mathbf{r}_t + \mathbf{H}_a\boldsymbol{\delta}$ made by the classifier paramertized by θ. This is solved by replacing the highly non-linear optimization constraint with an objective function, denoted by $\iota(\cdot)$, and reformulating the initial optimization as

$$\min_{\boldsymbol{\delta}} \quad ||\boldsymbol{\delta}||_2 + c \cdot \iota(\mathbf{r}_t + \mathbf{H}_a\boldsymbol{\delta}), \tag{11}$$

which is solved using stochastic gradient descent (SGD) and the smallest possible value for $c > 0$ that results in $\iota(\mathbf{r}_t + \mathbf{H}_a\boldsymbol{\delta}) \leq 0$.

FGSM: In this case, for a time domain attack the adversary adds an l_2-bounded perturbation, given by

$$\boldsymbol{\delta} = \sqrt{P_T}\frac{\nabla_{\mathbf{r}_t}\mathcal{L}_n(\mathbf{r}_{t,n}, \mathbf{y}_n, \theta)}{||\nabla_{\mathbf{r}_t}\mathcal{L}_n(\mathbf{r}_{t,n}, \mathbf{y}_n, \theta)||_2}, \tag{12}$$

to the transmitted signal, $\mathbf{r}_{t,n}$, in a single step exhausting the power budget, P_T. Formally, the nth perturbed received signal

is given by

$$\mathbf{r}_{\text{a, n}} = \mathbf{r}_{\text{t},n} + \mathbf{H}_{\text{a}} \sqrt{P_T} \frac{\nabla_{\mathbf{r}_{\text{t}}} \mathcal{L}_n(\mathbf{r}_{\text{t},n}, \mathbf{y}_n, \theta)}{||\nabla_{\mathbf{r}_{\text{t}}} \mathcal{L}_n(\mathbf{r}_{\text{t},n}, \mathbf{y}_n, \theta)||_2}, \qquad (13)$$

where $\mathcal{L}$ refers to the cost function of $f(\cdot, \theta)$ in (8). Similarly, for a frequency domain attack, the additive perturbation is given by $\boldsymbol{\delta} = \sqrt{P_T} \frac{\nabla_{\mathbf{R}_{\text{t}}} \mathcal{L}_n(\mathbf{R}_{\text{t},n}, \mathbf{y}_n, \phi)}{||\nabla_{\mathbf{R}_{\text{t}}} \mathcal{L}_n(\mathbf{R}_{\text{t},n}, \mathbf{y}_n, \phi)||_2}$ resulting in the nth perturbed received signal being

$$\mathbf{R}_{\text{a},n} = \mathbf{R}_{\text{t},n} + \mathbf{H}_{\text{a}} \sqrt{P_T} \frac{\nabla_{\mathbf{R}_{\text{t}}} \mathcal{L}_n(\mathbf{R}_{\text{t},n}, \mathbf{y}_n, \phi)}{||\nabla_{\mathbf{R}_{\text{t}}} \mathcal{L}_n(\mathbf{R}_{\text{t},n}, \mathbf{y}_n, \phi)||_2}. \qquad (14)$$

Adding a perturbation in the direction of the cost function's gradient behaves as performing a step of gradient ascent, thus aiming to increase the classification error on the perturbed sample.

BIM: The BIM is an iterative extension of the FGSM. Specifically, in each iteration, a fraction of the total power budget, $\alpha < P_T$, is added to the perturbation, and the optimal direction of attack (the direction of the gradient) is recalculated. Formally, the total perturbation, $\boldsymbol{\Delta} \in \mathbb{R}^{\ell \times 2}$, is initialized to zero (i.e., $\boldsymbol{\Delta}_n^{(0)} = \mathbf{0}$), and the perturbation on iteration $k + 1$ on the nth sample in the time domain is calculated according to

$$\boldsymbol{\Delta}_n^{(k+1)} = \boldsymbol{\Delta}_n^{(k)} + \sqrt{\alpha} \frac{\nabla_{\mathbf{r}_{\text{t}}} \mathcal{L}_n(\mathbf{r}_{\text{t},n}^{(k)}, \mathbf{y}_n, \theta)}{||\nabla_{\mathbf{r}_{\text{t}}} \mathcal{L}_n(\mathbf{r}_{\text{t},n}^{(k)}, \mathbf{y}_n, \theta)||_2}, \qquad (15)$$

which results in the perturbation given by $\boldsymbol{\delta} = \sqrt{P_T} \frac{\boldsymbol{\Delta}_n}{||\boldsymbol{\Delta}_n||_2}$. Formally, the nth signal perturbed using the BIM is given by

$$\mathbf{r}_{\text{a},n} = \mathbf{r}_{\text{t},n} + \mathbf{H}_{\text{a}} \sqrt{P_T} \frac{\boldsymbol{\Delta}_n}{||\boldsymbol{\Delta}_n||_2}. \qquad (16)$$

Similarly, for a frequency domain attack, the total perturbation is calculated according to

$$\boldsymbol{\Delta}_n^{(k+1)} = \boldsymbol{\Delta}_n^{(k)} + \sqrt{\alpha} \frac{\nabla_{\mathbf{R}_{\text{t}}} \mathcal{L}_n(\mathbf{R}_{\text{t},n}^{(k)}, \mathbf{y}_n, \phi)}{||\nabla_{\mathbf{R}_{\text{t}}} \mathcal{L}_n(\mathbf{R}_{\text{t},n}^{(k)}, \mathbf{y}_n, \phi)||_2}, \qquad (17)$$

yielding the perturbed frequency domain signal

$$\mathbf{R}_{\text{a},n} = \mathbf{R}_{\text{t},n} + \mathbf{H}_{\text{a}} \sqrt{P_T} \frac{\boldsymbol{\Delta}_n}{||\boldsymbol{\Delta}_n||_2}, \qquad (18)$$

where the final additive perturbation, $\boldsymbol{\Delta}_n$, in both the time and frequency domain is scaled by $\frac{\sqrt{P_T}}{||\boldsymbol{\Delta}_n||_2}$ to satisfy the power constraint of the adversary.

CW: The CW attack seeks to optimize (11) using SGD. Formally, the adversarial perturbation, with the effect of the adversary's channel, on the nth received signal is given by

$$\min_{\mathbf{w}} \left|\left| \frac{1}{2} \left(\tanh(\mathbf{w}) + 1 \right) - \mathbf{r}_{t,n} \right|\right|_2^2 + c \cdot \iota \left(\frac{1}{2} (\tanh(\mathbf{w}) + 1) \right), \quad (19)$$

where c is a constant and $\iota(\cdot)$ is empirically found to be

$$\iota(\mathbf{r}_{\text{a}}) = \max \left(\max_{i \neq t} \{ Z(\mathbf{r}_{\text{a}}, \theta)_i - Z(\mathbf{r}_{\text{a}}, \theta)_t \}, -\kappa \right), \qquad (20)$$

where $Z(\cdot, \theta)$ outputs the classifier's logit vector for $(\cdot)$ (i.e., the classifier predictions prior to softmax normalization), i is the correct modulation prediction, and t is the second most likely constellation prediction of $\mathbf{r}_{\text{a}}$ made by the classifier

parameterized by θ. In addition, the parameter κ controls the confidence with which misclassification occurs (nominally to class t) on $\mathbf{r}_{\text{a}}$. κ can be interpreted as a confidence parameter, with higher κ making a sample more likely to be misclassified when transferred to a different model. More details on this derivation are given in [48].

For each attack, we assume a naive adversarial attack instantiation, where we set $\mathbf{H}_{\text{a}} = \mathbf{I}$, following [13], [40], [41]. As a result, each element of the crafted perturbation retains its sign and magnitude after going through the channel. This general setup focuses on controlling the model's behavior at the transmitter and receiver, while considering the most stringent threat model for the adversary. Furthermore, since $\mathbf{H}_{\text{a}} = \mathbf{I}$, P_T directly corresponds to both the adversary's power constraint as well as the received power at the sink.

E. Resilience to Transferable Interference

We demonstrate the resilience of our wireless AMC receiver to transferable adversarial interference in architecture uncertainty and signal domain uncertainty environments. In the architecture uncertainty threat model, the adversary has access to one of the classifiers contained within the receiver trained on IQ time domain samples. In this case, $\boldsymbol{\delta}$, in both the FGSM and BIM attacks, is constructed using the gradient of the accessible classifier and transmitted alongside $\mathbf{r}_t$. Specifically, we evaluate the improvement provided by $\hat{\mathcal{C}}(\mathbf{r}_{\text{a}}, \theta_j) = \arg\max_k f_k^{(j)}(\mathbf{r}_{\text{a}}, \theta_j)$ when an attack is crafted using P_T and $\nabla_{\mathbf{r}_{\text{t}}} \mathcal{L}(\mathbf{r}_{\text{t}}, \mathbf{y}, \theta_i)$, which we evaluate $\forall i \neq j$.

In the signal domain uncertainty environment, the adversary crafts $\boldsymbol{\delta}$ using the gradient of either $f^{(i)}(\cdot, \theta_i)$ or $g^{(i)}(\cdot, \phi_i)$ and attempts to transfer the attack onto $g^{(i)}(\cdot, \phi_i)$ or $f^{(i)}(\cdot, \theta_i)$, respectively. Formally, the transferability of an attack from the time domain to the frequency domain is assessed through the accuracy improvement provided by $f^{(i)}(\cdot, \theta_i)$ when the adversary operates based on $g^{(i)}(\cdot, \phi_i)$ (and vice versa for assessing the transferability of an attack from the frequency domain to the time domain). In this scenario, we demonstrate the resilience of our wireless AMC receiver to transferable attacks targeted at degrading domain specific classifiers.

F. Assorted Deep Ensemble Defense

We now develop our defense against adversarial interference in a complete black box attack environment in which the adversary is blind to both the classification architecture and the signal domain used at the receiver. Here, we introduce our assorted deep ensemble (ADE) defense, which offers diversity among both classification architectures and signal representations. Contrary to deep ensemble models that have been proposed for other applications in prior work [49], our proposed ADE for AMC employs a variety of models trained on both IQ and frequency-based features. Furthermore, each classifier contained in our ADE defense is trained using Gaussian smoothing, which improves classification performance on out-of-distribution waveforms. Specifically, Gaussian smoothing involves adding multiple copies of each training sample into the training set, where each copied signal is randomly perturbed. When the training dataset is augmented with a
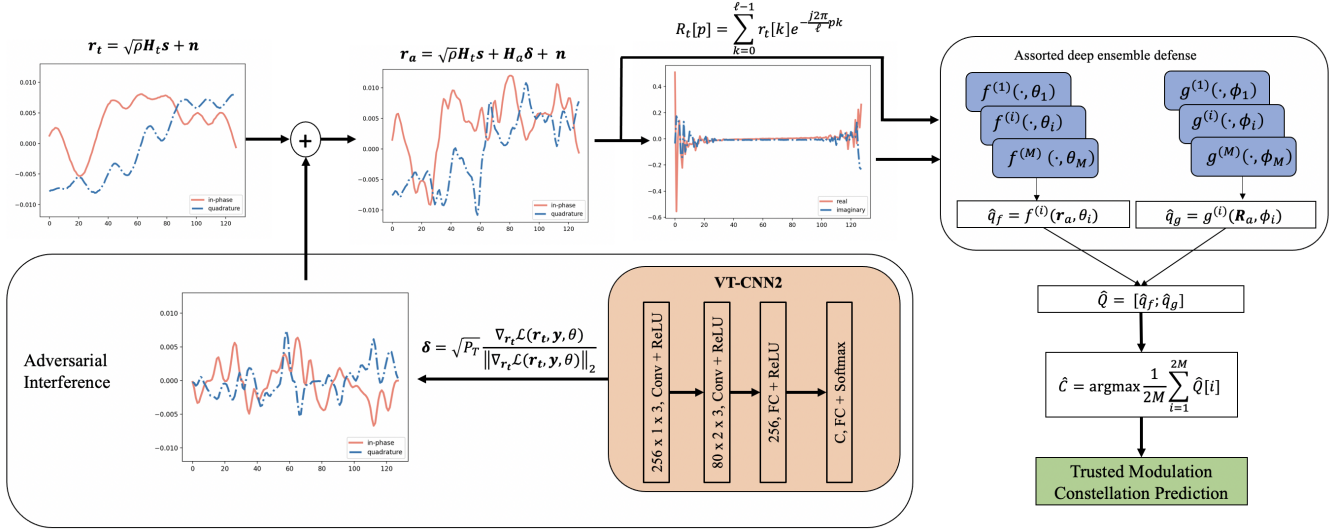
Fig. 2: Illustration of our AMC methodology in the black box adversarial attack environment on a GFSK modulation signal, where the adversary is forced to use the gradient of a surrogate model to craft the interference signal.

sufficient amount of random perturbations in this fashion, the classification performance of a single DL model increases on adversarial examples, since the random noise accounts for various distortions that may be induced by adversarial examples [39]. Finally, each classifier in our ADE is trained using the entire available training set (as opposed to bootsrap aggregating, i.e., *bagging*, traditionally used in ensemble training), where different random initializations (as well as, in our case, Gaussian noise signal perturbations in the training set) are used to create diversity among the models.

Algorithm 1 outlines the training process for our proposed ADE. Here, $\mathcal{N}$ denotes generating an $\ell \times 2$ matrix from a Gaussian distribution. Consistent with the design of black box attacks crafted in prior work [20]–[22], [42], the adversary uses a surrogate deep learning classifier to craft their attack, which is transmitted to the underlying classification model at the receiver (we will discuss the procedure used by the adversary to construct the surrogate model in our experiments in Sec. IV-E). In this fashion, our defense is especially applicable in overall uncertainty black box environments when an adversary cannot access the gradient of the underlying classifier at the receiver.

During deployment, the modulation constellation of an input, $\mathbf{r}_a$, is predicted by aggregating the output of each classifier in the ensemble trained on IQ features, $F = \{f^{(1)}(\cdot, \theta_1), \ldots, f^{(M)}(\cdot, \theta_M)\}$, and frequency features, $G = \{g^{(1)}(\cdot, \phi_1), \ldots, g^{(M)}(\cdot, \phi_M)\}$. Algorithm 2 outlines the deployment of our ADE defense against adversarial signals. The application of our black box defense is illustrated in Fig. 2.

IV. RESULTS AND DISCUSSION

In this section, we conduct an empirical evaluation of our AMC methodology. First, we overview the datasets that we use (Sec. IV-A). Next, we present the baseline classification performances in the absence of adversarial interference (Sec. IV-B). We then demonstrate our wireless receiver's resilience to transferable adversarial interference between classification

architectures (Sec. IV-C) and signal domains (Sec. IV-D). Finally, we evaluate our assorted deep ensemble (ADE) defense and demonstrate its robustness in black box attack environments over three comparative baselines (Sec. IV-E).

A. Datasets and Evaluation Metrics

We employ the GNU RadioML2016.10a (referred to as Dataset A) [50] and RadioML2018.01a (referred to as Dataset B) [23] datasets for our analysis. Each signal in the RadioML2016 dataset ($\mathbf{r}_{t,n}$) is normalized to unit energy and consists of a 128-length ($\ell = 128$) observation window modulated according to a certain constellation ($\mathbf{y}_n$). In order to isolate the impact of adversarial interference, we focus on the following four modulation schemes, which, as we will show in Sec. IV-B, have equivalent classification performance on both time and frequency domain features in the absence of adversaries: CPFSK, GFSK, PAM4, and QPSK. Each constellation set contains 6000 examples for a total of 24000 signals. In the RadioML2018.01a dataset, each considered signal is oversampled with an observation window of $\ell = 1024$. For a consistent comparison between datasets, signals in the RadioML2018.01a dataset are downsampled by $1/8$ to obtain an observation window of $\ell = 128$. We then focus on the following modulation constellations within the dataset: OOK, 8ASK, BPSK, and FM. Each modulation scheme contains 4096 examples for a total of 16384 signals. Following in line with previous studies [13], [40], [41], we focus on high SNR signals, where small perturbations can significantly degrade the classification performance of otherwise robust AMC models. Therefore, for each considered dataset, we present results at a constant SNR of 18 dB unless otherwise stated. However, our conclusions hold at SNRs greater than 2 dB, where previous works (e.g., [32]) have achieved high classification accuracy for deep learning-based AMC. We will analyze the effects of varying the SNR more closely in Sec. IV-E.

Algorithm 1 ADE Construction

1: **input:** $\mathcal{X}^{\text{IQ}}$: IQ feature-based training set
     $\mathcal{X}^{\text{DFT}}$: frequency feature-based training set
     $\boldsymbol{\zeta} = \{\zeta^{(1)}, \ldots, \zeta^{(M)}\}$: set of randomly initialized untrained deep learning architectures
     k: number of noisy samples generated per signal
     σ_{IQ}: standard deviation of Gaussian noise added to IQ samples
     σ_{DFT}: standard deviation of Gaussian noise added to DFT samples
2: **initialize:** $F \leftarrow \emptyset$
     $G \leftarrow \emptyset$
3: **for** $i = 1, \ldots, M$ **do**
4:    **for** $j = 1, \ldots, k$ **do**
5:       $\mathcal{X}^{\text{IQ}}_{\text{noisy}} \leftarrow \emptyset$
6:       $\mathcal{X}^{\text{DFT}}_{\text{noisy}} \leftarrow \emptyset$
7:       **for** $\mathbf{r}_t, \mathbf{R}_t \in \mathcal{X}^{\text{IQ}}, \mathcal{X}^{\text{DFT}}$ **do**
8:         $\mathbf{n}_j^{\ell \times 2} \leftarrow \mathcal{N}(\mu = 0, \sigma = \sigma_{\text{IQ}})$
9:         $\mathbf{N}_j^{\ell \times 2} \leftarrow \mathcal{N}(\mu = 0, \sigma = \sigma_{\text{DFT}})$
10:        $\tilde{\mathbf{r}}_t \leftarrow \mathbf{r}_t + \mathbf{n}_j$
11:        $\tilde{\mathbf{R}}_t \leftarrow \mathbf{R}_t + \mathbf{N}_j$
12:        $\mathcal{X}^{\text{IQ}}_{\text{noisy}} \leftarrow \mathcal{X}^{\text{IQ}}_{\text{noisy}} \cup \tilde{\mathbf{r}}_t$
13:        $\mathcal{X}^{\text{DFT}}_{\text{noisy}} \leftarrow \mathcal{X}^{\text{DFT}}_{\text{noisy}} \cup \tilde{\mathbf{R}}_t$
14:       **end for**
15:    **end for**
16:    $f^{(i)}(\cdot, \theta_i) \leftarrow$ train $\zeta^{(i)}$ on $\mathcal{X}^{\text{IQ}}_{\text{noisy}}$
17:    $F \leftarrow F \cup f^{(i)}(\cdot, \theta_i)$
18:    $g^{(i)}(\cdot, \phi_i) \leftarrow$ train $\zeta^{(i)}$ on $\mathcal{X}^{\text{DFT}}_{\text{noisy}}$
19:    $G \leftarrow G \cup g^{(i)}(\cdot, \phi_i)$
20: **end for**
21: **return** F, G

Algorithm 2 ADE Deployment

1: **input:** $\mathbf{r}_a$: perturbed wireless signal
     $F = \{f^{(1)}(\cdot, \theta_1), \ldots, f^{(M)}(\cdot, \theta_M)\}$
     $G = \{g^{(1)}(\cdot, \phi_1), \ldots, g^{(M)}(\cdot, \phi_M)\}$
2: **initialize:** $\hat{q}_f \leftarrow \mathbf{0}^{M \times C}$
     $\hat{q}_g \leftarrow \mathbf{0}^{M \times C}$
3: **for** $i = 1, \ldots, M$ **do**
4:    $\hat{q}_f[i] \leftarrow f^{(i)}(\mathbf{r}_a, \theta_i)$
5:    $\mathbf{R}_a \leftarrow \sum_{k=0}^{\ell-1} r_a[k] e^{-\frac{j2\pi}{\ell} pk}$ **for** $p = 0, \ldots, \ell - 1$
6:    $\hat{q}_g[i] \leftarrow g^{(i)}(\mathbf{R}_a, \phi_i)$
7: **end for**
8: $\hat{Q}^{2M \times C} \leftarrow [\hat{q}_f; \hat{q}_g]$
9: $\hat{\mathbf{y}} \leftarrow \mathbf{0}^{C \times 1}$
10: **for** $j = 1, \ldots, C$ **do**
11:    $\hat{y}[j] \leftarrow \frac{1}{2M} \sum_{i=1}^{2M} \hat{Q}[i, j]$
12: **end for**
13: $\hat{C} \leftarrow \arg\max_i \hat{\mathbf{y}}_i$
14: **return** $\hat{C}$

TABLE I: The testing accuracy of each considered model on $\mathcal{X}^{(\cdot)}_{te}$ as well as the Pearson Product-Moment correlation coefficient (PPMCC) between the validation sets of Dataset A and Dataset B. The CNN outperforms every other considered model (although the CRNN delivers equivalent accuracy, it is achieved with a longer training time on both datasets).

Model	Input Features	Accuracy on Dataset A	Accuracy on Dataset B	PPMCC
FCNN	IQ	92.78%	99.35%	0.64
FCNN	Frequency	92.22%	99.10%	0.57
CNN	IQ	99.19%	99.59%	1.0
CNN	Frequency	99.25%	99.72%	0.85
RNN	IQ	93.61%	98.58%	0.80
RNN	Frequency	92.53%	99.15%	0.65
CRNN	IQ	99.61%	99.59%	0.80
CRNN	Frequency	98.92%	99.63%	0.52

In each experiment, we employ a 70/15/15 training/validation/testing dataset split, where the training and validation data are used to estimate the parameters of $f^{(i)}(\cdot, \theta_i)$ and $g^{(i)}(\cdot, \phi_i)$, and the testing dataset is used to evaluate each trained model's susceptibility to adversarial interference as well as the effectiveness of our proposed defense. In particular, the validation set is used to tune the model parameters using unseen data during the training process whereas the testing set is used to measure the performance of the resulting model. For each dataset, we denote the training, validation, and testing datasets, consisting of time domain IQ points or frequency domain feature components, as $\mathcal{X}^t_{tr}$, $\mathcal{X}^t_{va}$, $\mathcal{X}^t_{te}$, $\mathcal{X}^\omega_{tr}$, $\mathcal{X}^\omega_{va}$, and $\mathcal{X}^\omega_{te}$, respectively.

To measure the potency of the additive adversarial interference, we use the perturbation-to-noise ratio (PNR), which is given by

$$\text{PNR [dB]} = \frac{\mathbb{E}[\|\boldsymbol{\delta}\|_2^2]}{\mathbb{E}[\|\mathbf{r}_a\|_2^2]} \text{ [dB]} + \text{SNR [dB]}, \quad (21)$$

where $\mathbb{E}$ is the expected value. A higher PNR indicates higher levels of additive interference. We consider a perturbation to be imperceptible when PNR ≤ 1 dB because the perturbation power would be at or below the noise power. At high PNR (i.e., PNR > 1 dB), the underlying signal is masked to a greater extent by the perturbation, making effective classification difficult in any case due to the loss of salient features across classification architectures.

At each PNR, we measure the accuracy of the considered testing set (i.e., $\mathcal{X}^t_{te}$ or $\mathcal{X}^\omega_{te}$), which is given by dividing the total number of correctly classified samples by the total number of samples in the set. Although random predictions would yield an accuracy of 25% (since $C = 4$) in our experiments, we will see that adversarial perturbations can result in classification accuracies significantly below random guessing, indicating their potency on DL-based wireless communication networks.

B. AMC Wireless Receiver Performance

We begin by evaluating the performance of both $f^{(i)}(\cdot, \theta_i)$ and $g^{(i)}(\cdot, \phi_i)$ in the absence of adversarial interference. In Figs. 3 and 4, we plot the evolution of the classification accuracy across training epochs achieved by each deep learning architecture on their respective training and validation sets. *We see that each model trained using our proposed frequency*
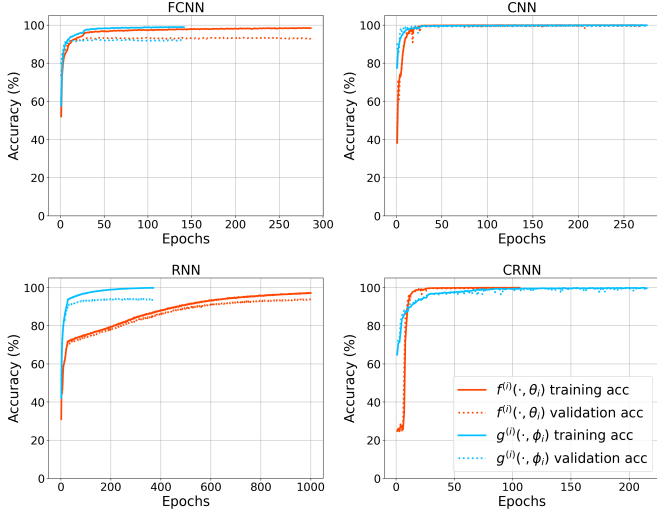
Fig. 3: The model training performance on Dataset A of each considered AMC architecture on the corresponding training and validation sets. We see that the frequency-based features $g(\cdot, \phi)$ outperform or match the time domain features $f(\cdot, \theta)$ in terms of both training and validation accuracy for each deep learning architecture. The CNN results in the fastest convergence and highest accuracy for both $f(\cdot, \theta)$ and $g(\cdot, \phi)$.
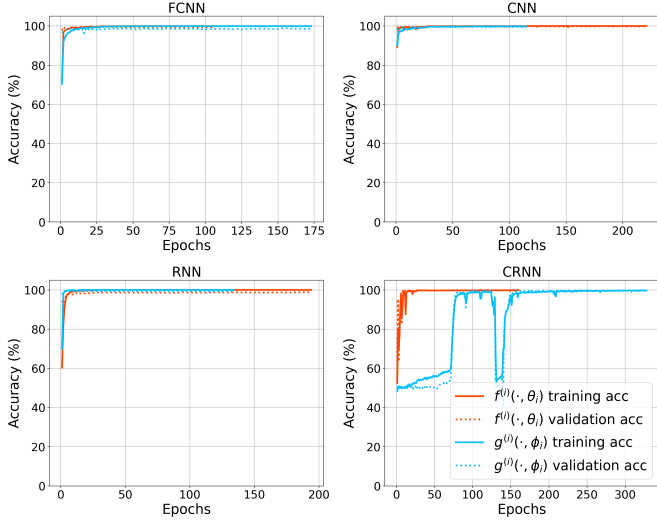

Fig. 4: The model training performance on Dataset B of each considered AMC architecture on the corresponding training and validation sets. Similar to Fig. 3, we see that the CNN and CRNN achieve robust classification performance on both IQ and frequency features. Unlike Fig. 3, the FCNN and RNN experience similar performance to the CNN and CRNN.

feature-based input performs equivalently or outperforms its time domain counter-part model in terms of final accuracy (calculated with regards to the subset of considered modulation constellations in each dataset) and required training epochs, with the exception of the CRNN. We also see in Figs. 3 and 4 that, *among each considered architecture, the CNN and CRNN consistently obtain the best performance overall on their validation sets.* Contrarily, both IQ and frequency features present more challenges during training on the FCNN and RNN compared to the CNN and CRNN on Dataset
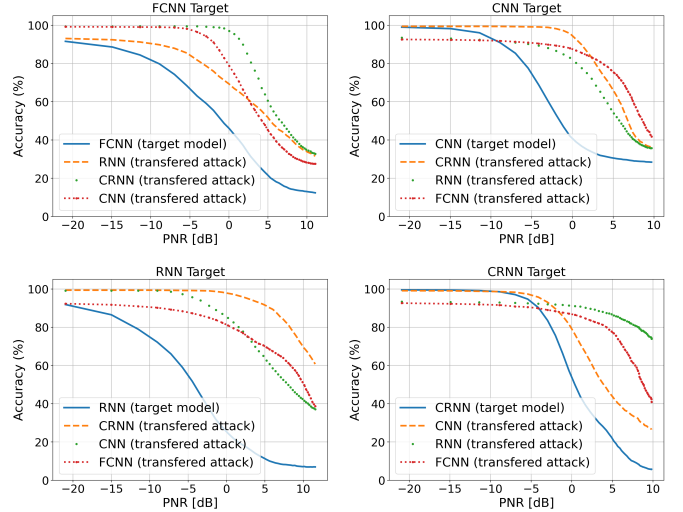

Fig. 5: The transferability of the BIM perturbation between classification architectures on Dataset A. We see that the effects of FCNN, CNN, and RNN instantiated attacks are nearly eliminated on the CRNN in the low PNR regime whereas the effects of CRNN instantiated attacks fail to strongly transfer onto RNNs.

A, whereas the CRNN presents more training instability on Dataset B. Specifically, on Dataset A, both the FCNN and the RNN experience slight overfitting to the training data and fail to converge on a validation accuracy greater than 94%, while the CNN and CRNN models present generalizable and robust performance nearing or exceeding 99%. Dataset B, however, experiences a sudden drop in training accuracy on the CRNN but does not experience overfitting on any classifier trained on either IQ or frequency features, while delivering a validation accuracy greater than 99% on each classification architecture.

Each trained model's accuracy achieved on its corresponding testing set is shown in Table I. In addition, we show the Pearson Product-Moment correlation coefficient (PPMCC) [51] between the validation accuracies (at the end of each training iteration) on Dataset A and Dataset B for the same set of signal features (i.e., IQ vs. FFT) on each dataset. Among all eight considered models, the CNN trained on frequency features achieves the highest testing accuracy while converging using the fewest epochs as shown in Figs. 3 and 4. Although the CRNN achieves a slightly higher testing accuracy, the higher number of required epochs results in substantially higher computational overhead (the CNN converges three times faster than the CRNN). Moreover, the CNN has the highest PPMCC correlation between Dataset A and Dataset B on their corresponding classifiers. *Therefore, our proposed CNN trained using frequency features is the most desirable model in terms of classification performance, training time, consistency between datasets, and computational efficiency.*

C. Architecture Uncertainty Performance

In Figs. 5 - 6, we demonstrate the ability of our wireless receiver to withstand the effects of transferable adversarial interference in architecture uncertainty environments under the FGSM and BIM attacks for Dataset A and Dataset B. From these figures, we see that the potency of adversarial attacks are
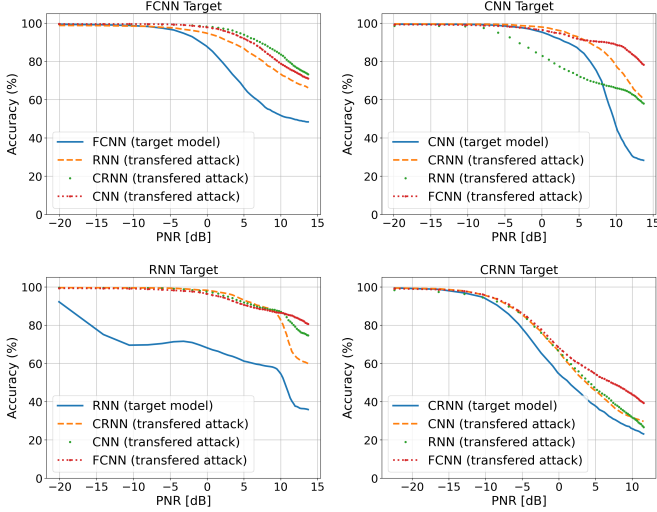
Fig. 6: The transferability of the FGSM perturbation between classification architectures on Dataset B. Here, transferrability is smaller than for Dataset A. The FCNN provides the greatest resilience in attacks targeted at the CNN, RNN, and CRNN, whereas the CRNN provides the greatest attack mitigation for interference targeted at the FCNN.

Fig. 7: The transferability of the FGSM attack between time and frequency domain classifiers on Dataset A. Both the RNN and CRNN mitigate the effects of the attack targeted at $g^{(i)}(\cdot, \phi_i)$ to the largest extent when transferred to $f^{(i)}(\cdot, \theta_i)$, while the frequency domain CNN classifier withstands time domain attacks to the greatest extent.

mitigated when they are transferred onto architectures differing from the ones used to craft the attacks. In particular, each classifier that experiences a transferred attack has a higher accuracy than the target model for all PNRs. Furthermore, in each case, we find particular classifiers that almost entirely withstand the effects of the additive interference. Specifically, in Fig. 5, the CRNN experiences nearly no degradation in the imperceptible PNR region for BIM attacks instantiated on the FCNN, CNN, and RNN, and similarly, the RNN reduces the effects of the CRNN instantiated attacks across nearly the entire considered PNR range. For Dataset B in Fig. 6, we observe the same general trends on the FGSM perturbation, except the attacks are less effective overall (due to the lower potency of the FGSM attack compared to the BIM attack), and thus, there is less variation in transferrability between architectures. *This indicates that black box adversarial attacks instantiated on AMC models are not directly transferable between the deep learning classification architectures considered in our wireless receiver.*

D. Signal Domain Uncertainty Performance

We now evaluate our receiver's ability to withstand transferable adversarial interference between signal domains. Figs. 7 and 8 give the results for the FGSM and BIM attacks on Dataset A. Each plot shows the transferrability of an attack on the time domain trained classifier, $f^{(i)}(\cdot, \theta_i)$, to the frequency domain trained classifier, $g^{(i)}(\cdot, \phi_i)$, and vice versa, for each architecture. In addition, Table II shows the transferrability results on Dataset B for time domain and frequency domain attacks on varying PNRs.

Fig. 7, as well as Table II, demonstrate the resilience of each trained classifier to withstand FGSM attacks. For both datasets, we see that there are certain architectures for which transferrability from the time domain to the frequency domain,
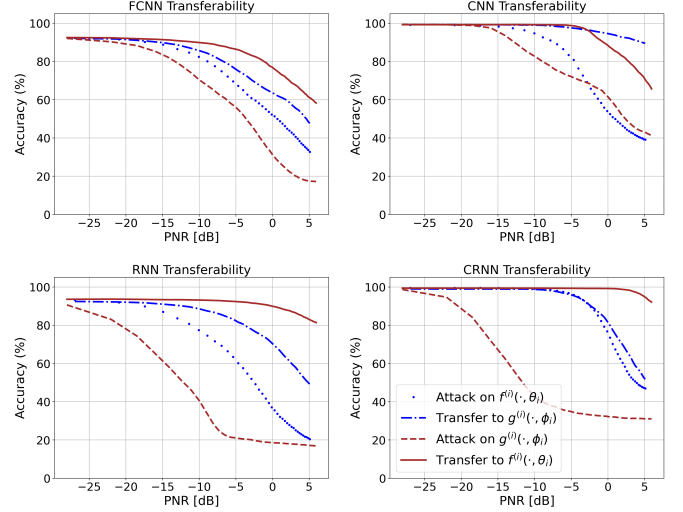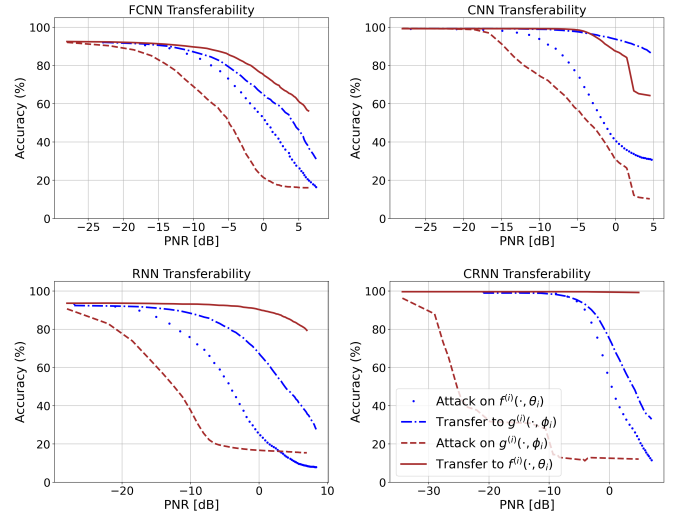

Fig. 8: The transferability of the BIM attack between time and frequency domain classifiers on Dataset A. Similar to Fig. 7, we see that the RNN and CRNN mitigate frequency domain-based attacks to the greatest extent while the the CNN withstands time domain instantiated attacks to the greatest extent.

and from the frequency domain to the time domain, are significantly mitigated. In particular, the RNN and CRNN architectures demonstrate the greatest resilience in that they achieve accuracy improvements greater than 70% and 65%, respectively, on Dataset A while the attack is less potent on Dataset B for the same DL architectures at 0 dB PNR. Furthermore, the CNN demonstrates significant gains in classification performance when an attack is transferred from the time domain to the frequency domain on both datasets (e.g., improving accuracy from 39.14% to 89.50% at 5 dB PNR on Dataset A and from 43.81% to 87.18% on Dataset B at 10 dB PNR). The ability of the CNN and CRNN to withstand attacks to the highest degree overall indicates their increased

TABLE II: The classification accuracies of the FGSM and BIM attacks between the time and frequency domain classifiers on Dataset B at low (-10 and -5 dB), medium (0 dB), and high (5 and 10 dB) perceptibility levels. Consistent with Dataset A, we see that that adversarial perturbations are not directly transferable between signal domains.

PNR (dB)	Model	IQ FGSM Attack		Freq. BIM Attack	
		$f(\cdot, \theta)$	$g(\cdot, \phi)$	$g(\cdot, \phi)$	$f(\cdot, \theta)$
-10	FCNN	98.82%	98.25%	95.20%	98.94%
	CNN	99.22%	99.10%	89.38%	97.97%
	RNN	69.40%	98.90%	76.03%	84.25%
	CRNN	94.71%	96.42%	99.06%	99.47%
-5	FCNN	96.46%	95.97%	90.84%	98.41%
	CNN	98.29%	98.41%	71.72%	93.65%
	RNN	70.01%	98.45%	74.61%	83.16%
	CRNN	75.34%	84.25%	97.97%	99.35%
0	FCNN	88.03%	91.74%	69.60%	94.79%
	CNN	95.20%	96.46%	62.08%	77.87%
	RNN	69.48%	95.52%	73.35%	80.68%
	CRNN	58.74%	72.50%	88.36%	97.84%
5	FCNN	66.27%	80.84%	53.30%	79.70%
	CNN	85.64%	91.66%	59.89%	64.04%
	RNN	63.67%	90.11%	70.79%	78.44%
	CRNN	37.23%	53.09%	57.53%	90.85%
10	FCNN	51.74%	70.67%	43.04%	63.63%
	CNN	43.81%	87.18%	61.51%	74.17%
	RNN	54.19%	85.55%	68.02%	74.65%
	CRNN	25.87%	40.36%	15.37%	63.38%

resilience to transferable adversarial interference.

The effectiveness of our proposed defense against the BIM adversarial attack, as shown in Fig. 8 and Table II, consistent with the response of the FGSM attacks: BIM instantiated adversarial attacks are not directly transferable between signal domains. Moreover, the time domain-based CRNN eliminates the degradation effects of the frequency domain instantiated attacks almost completely. However, the degree to which BIM attacks effectively transfer between domains differs between Dataset A and Dataset B, indicating that the mitigation of adversarial attacks in the domain uncertainty environment may be dataset dependent. *Thus, as shown by the instantiation of both considered attacks, our wireless receiver architecture mitigates the transferability of adversarial interference between IQ-based and frequency-based features to a significant degree.*

We analyze the CNN and CRNN's resilience to transferable attacks between signal domains on Dataset A more closely in Figs. 9 – 10. We consider the CNN's performance on a time domain attack compared to the case of no interference in Fig. 9, and the CRNN's performance on a frequency domain attack compared to the case of no interference in Fig. 10.

As shown in Figs. 9 and 10, both time and frequency features deliver robust AMC performance in the absence of adversarial interference with classification rates around 99% for both models. However, at a PNR of 5 dB, the classification rate drops to 39.14% and 31.03% when the FGSM attack is transmitted in the time domain, in Fig. 9, and frequency domain, in Fig. 10, respectively. From Figs. 9 and 10, we see that the adversarial interference pushes the majority of signals within the classification decision boundaries of the PAM4 modulation constellation for the CNN in the time domain, and the CPFSK constellation for the CRNN in the frequency
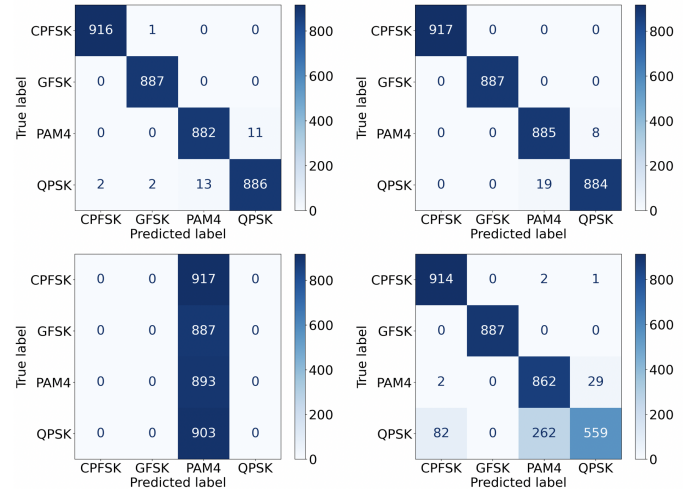

Fig. 9: The CNN classifiers on Dataset A, with no interference, achieve roughly equivalent performance using IQ features (top left) and frequency features (top right). Moreover, the instantiation of a 5 dB PNR FGSM attack in the time domain (bottom left) is significantly mitigated on the frequency domain classifier (bottom right), with the largest incongruency being between PAM4 and QPSK.
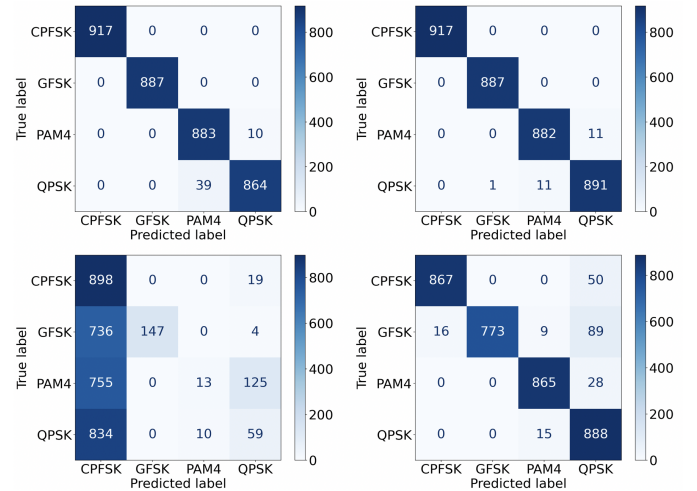

Fig. 10: The CRNN classifiers on Dataset A, with no interference, achieve roughly equivalent performance using frequency features (top left) and IQ features (top right), and furthermore, the instantiation of a 5 dB PNR FGSM attack in the frequency domain (bottom left) is significantly mitigated on the time domain feature-based model (bottom right).

domain. This is largely due to the nature of the untargeted attack in which the adversary's sole objective is to induce misclassification without targeting a specific misclassified prediction. The attacks are mitigated to a large extent when they are transferred from the time domain to the frequency domain (Fig. 9) and from the frequency domain to the time domain (Fig. 10) with accuracies of 89.50% and 94.25%, corresponding to classification accuracy improvements of 50.36% and 63.22%, respectively. Frequency domain-based models correctly classify a majority of CPFSK and GFSK modulation schemes corrupted with time domain-based attacks, with the largest incongruency being between PAM4 and QPSK. Time domain-based models, on the other hand, exhibit stronger per-
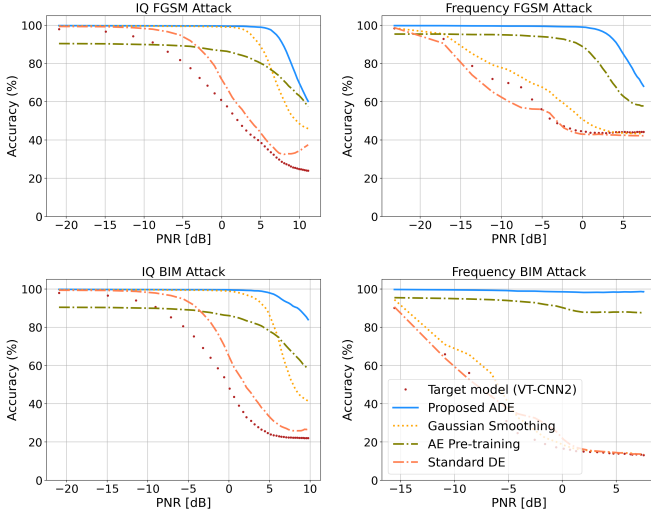
Fig. 11: Defense performance of our proposed ADE method compared with three baselines for various types of attack construction on Dataset A. We see that the ADE outperforms the baseline methods on each considered attack for each PNR.
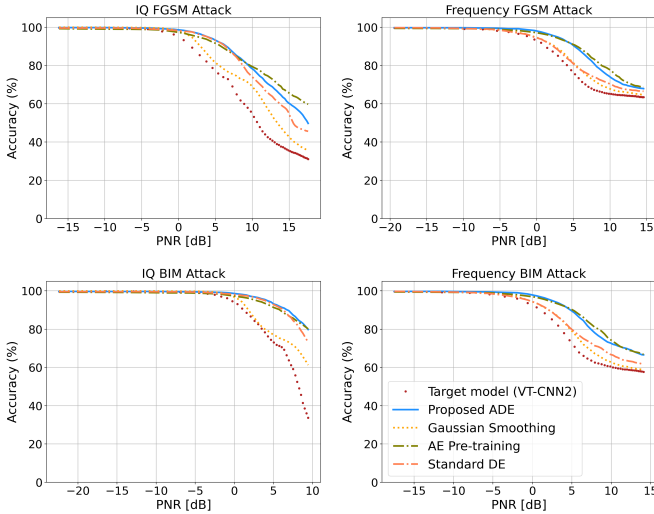

Fig. 12: Defense performance of our proposed ADE method compared with three baselines for various types of attack construction on Dataset B. We see that the ADE performs equivalently or outperforms baseline methods on each considered attack.
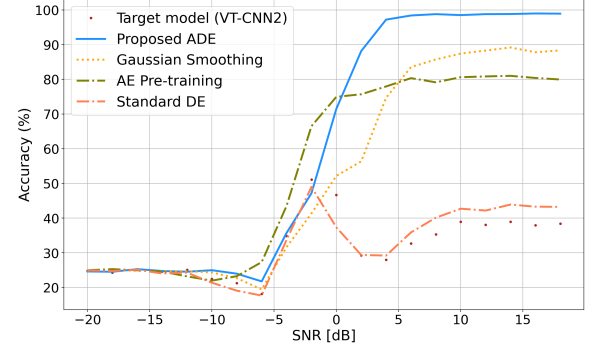

Fig. 13: Defense performance of our proposed ADE method compared with each baseline for varying SNRs at a constant PNR of 5 dB on Dataset A. We see that our method outperforms each baseline in the high SNR regime, while each method expectedly performs ineffectively in the low SNR regime.

surrogate models (e.g., the AMC classifier in [26] or other classifiers contained in our ADE) can be used by the adversary to generate transferable adversarial interference signals, and in such cases, we expect consistent attack mitigation by our ADE due to its demonstrated resilience against transferable adversarial attacks as shown in Sec. IV-C & IV-D.

To construct our defense strategies outlined in Algorithms 1 and 2, we use the CNN and CRNN architectures since, as demonstrated from the architecture uncertainty and signal uncertainty environments, they provide the greatest resilience to transferable adversarial interference. We use the following hyper-parameters for constructing the defense for both Dataset A and Dataset B: $M = 4$, $k = 30$, $\sigma_{IQ} = 0.001$, and $\sigma_{DFT} = 0.005$.

We compare our method to three previously proposed methods for adversarial interference mitigation in AMC: Gaussian smoothing [40], autoencoder pre-training [41], and deep ensembles (DE) [49]. Gaussian smoothing consists of retraining a single classifier with samples augmented with random noise in order to improve classification performance on various distortions that may be encountered during deployment such as adversarial examples. Autoencoder pre-training, on the other hand, trains an autoencoder and uses its encoder to calculate a latent space representation of the input data, which is then used to train an AMC classifier, with the rationale being that fewer degrees of freedom (i.e., a lower dimensional representation in the latent space) will prevent misclassifications from adversarial attacks. DEs employ the same classification architecture and use the same data representation for each classifier in a constructed ensemble, with the outputs aggregated to obtain the model's predicted modulation constellation.

Figs. 11 and 12 show the performance of our ADE when defending the transferred FGSM and BIM attack from the VT-CNN2 classifier on each dataset. On Dataset A, we see that our proposed defense outperforms the baselines for all attacks and PNRs considered, with almost no degradation in classification for PNRs below 2 dB. For example, our ADE method improves the classification accuracy from 22.23% to

formance on frequency domain-based attacks, with an overall misclassification rate of 5.75%.

E. Assorted Deep Ensemble Defense Performance

Lastly, we evaluate our proposed assorted deep ensemble (ADE) defense in the overall black box environment, where the adversary crafts an adversarial interference signal using the gradient of a surrogate classifier, as discussed in Sec. III-F. We assume that the adversary uses the VT-CNN2 classifier as the surrogate model, since it is a widely proposed model for DL-based AMC [13], [25], [35], [50]. The VT-CNN2 classifier is comprised of two convolutional layers containing 256 1×3 and 80 2×3 feature maps, respectively, followed by a 256-unit dense layer (as visualized in Fig. 2). Note that other

TABLE III: The testing accuracy of each model on $\mathcal{X}_{te}^{(\cdot)}$ for Dataset A when the CW perturbation is crafted by the adversary with varying confidence levels (κ). Our proposed ADE mitigates the attack to the greatest extent in comparison to the three baselines across multiple confidence levels of the attack.

κ	Input Features	Model Accuracies				
		VT-CNN2	**ADE**	Gaussian Smoothing [39]	Autoencoder pre-training [41]	Standard DE [49]
0	IQ	16.53%	**99.67%**	96.42%	87.00%	75.25%
	Frequency	11.94%	**99.58%**	40.22%	94.02%	36.92%
10	IQ	35.02%	**98.97%**	91.39%	83.19%	41.94%
	Frequency	12.00%	**99.19%**	13.06%	91.75%	21.83%
20	IQ	60.47%	**94.81%**	85.17%	82.06%	61.75%
	Frequency	15.67%	**98.72%**	15.80%	90.75%	16.78%
30	IQ	71.61%	**93.31%**	79.06%	82.08%	71.61%
	Frequency	16.36%	**98.36%**	16.58%	89.58%	16.67%
40	IQ	86.14%	**96.97%**	90.17%	86.58%	86.31%
	Frequency	17.58%	**98.50%**	17.72%	88.33%	17.78%
50	IQ	94.13%	**99.25%**	95.89%	89.39%	94.53%
	Frequency	18.44%	**98.19%**	18.50%	87.11%	18.47%

90.53% on the time domain BIM attack at 8 dB PNR, whereas Gaussian smoothing, autoencoder pre-training, and standard DEs achieve classification accuracies of 47.86%, 65.33%, and 25.94%, respectively, on the same attack. In comparison to DEs, we find that incorporating diversity into the classifiers comprising a deep ensemble, as we propose in our ADE, is necessary for mitigating transferable adversarial attacks. In our ADE, this diversity stems from the gradient mismatch between the adversary's classifier used to craft the attack and the classifiers used in the ADE at the receiver. In this regard, the gradient mismatch results in an attack that is less potent on the ADE, thus resulting in higher classification rates on adversarially perturbed inputs. In addition, for lower PNRs, we see that Gaussian smoothing outperforms autoencoder pre-training for time domain attacks, while autoencoder pre-training is significantly better than Gaussian smoothing for defending frequency domain instantiated attacks.

For Dataset B, we see that the performances of each defense are generally closer, which is consistent with our prior observations that attacks are less potent on this dataset. The performance of our ADE is comparable to autoencoder pre-training, while Gaussian smoothing performs similarly to no defense mitigation. Finally, we see that our proposed defense continues to mitigate the effects of attacks even after the received signal is masked by the perturbation (PNR > 0 dB).

In addition, Table III shows the performance of our ADE defense against the CW attack crafted on the VT-CNN2 classifier under varying confidence levels κ. Overall, we see that our proposed ADE defense significantly mitigates the effects of the CW perturbations across multiple confidence levels in a complete black box environment. By contrast, each of the baselines exhibits significant performance degradation as κ and/or the input features are varied. Similar results were observed on Dataset B (omitted for brevity).

We now consider the effect of varying the SNR while holding the PNR constant. Fig. 13 shows the classification performance of our method, as well as each considered baseline, for the SNR range of $[-20, 18]$ dB while holding the PNR constant at 5 dB. The results are shown here for the FGSM attack on the Dataset A (we find them to be consistent

for other cases as well). We see that, in comparison to the SNR of 18 dB at which the results have been presented in this paper, the performance obtained by our method is consistent when the SNR is greater than 2 dB. At SNRs lower than 2 dB, classification performance is overall degraded due to poor signal quality (consistent with prior work, e.g., [7], [8], [23]). In these cases, the classification performance, even of the improved gains in accuracy, are not very effective due to the poor signal environment induced from low SNR waveforms. Future work must consider methods to improve AMC performance in the low SNR regime before evaluating the effects of adversarial attacks in this setting.

V. CONCLUSION

Deep learning (DL) has recently been proposed as a robust method to perform automatic modulation classification (AMC). Yet, deep learning AMC models are vulnerable to adversarial attacks, which can alter a trained model's predicted modulation constellation with relatively little input power. Furthermore, such attacks are transferable, which allows the interference to degrade the performance of several classifiers simultaneously. In this work, we developed a novel wireless transmission receiver architecture – consisting of both time and frequency domain feature-based classification models – which is capable of mitigating the transferability of adversarial interference in black box environments. Specifically, we showed that our models are resilient to transferable adversarial attacks between DL classification architectures and between the time and frequency domain, where convolutional neural networks (CNNs) and convolutional recurrent neural networks (CRNNs) demonstrated the greatest degree of mitigation. Using these insights, we proposed our assorted deep ensemble defense, which defends a wireless receiver from complete black box adversarial perturbations. We found that our proposed method is capable of mitigating adversarial AMC attacks to a greater extent than previously proposed methods, thus increasing the robustness of deep learning AMC receivers from malicious behavior.

REFERENCES

[1] R. Sahay, C. G. Brinton, and D. J. Love, "Frequency-based automated modulation classification in the presence of adversaries," in *ICC 2021 - IEEE International Conference on Communications*, 2021, pp. 1–6.

[2] C.-Y Huan and A. Polydoros, "Likelihood methods for mpsk modulation classification," *IEEE Transactions on Communications*, vol. 43, no. 234, pp. 1493–1504, 1995.

[3] W. Wei and J. M. Mendel, "Maximum-likelihood classification for digital amplitude-phase modulations," *IEEE Transactions on Communications*, vol. 48, no. 2, pp. 189–193, 2000.

[4] F. Hameed, O. A. Dobre, and D. C. Popescu, "On the likelihood-based approach to modulation classification," *IEEE Transactions on Wireless Communications*, vol. 8, no. 12, pp. 5884–5892, 2009.

[5] J. L. Xu, W. Su, and M. Zhou, "Likelihood-ratio approaches to automatic modulation classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 4, pp. 455–469, 2011.

[6] P. Panagiotou, A. Anastasopoulos, and A. Polydoros, "Likelihood ratio tests for modulation classification," in *21st Century Military Communications (MILCOM)*, 2000, pp. 670–674.

[7] F. Meng, P. Chen, L. Wu, and X. Wang, "Automatic modulation classification: A deep learning enabled approach," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10 760–10 772, 2018.

[8] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *International conference on engineering applications of neural networks*, 2016, pp. 213–226.

[9] S. Huang, Y. Jiang, Y. Gao, Z. Feng, and P. Zhang, "Automatic modulation classification using contrastive fully convolutional network," *IEEE Wireless Communications Letters*, vol. 8, no. 4, pp. 1044–1047, 2019.

[10] D. Hong, Z. Zhang, and X. Xu, "Automatic modulation classification using recurrent neural networks," in *IEEE International Conference on Computer and Communications (ICCC)*. IEEE, 2017, pp. 695–700.

[11] L. Huang, Y. Zhang, W. Pan, J. Chen, L. P. Qian, and Y. Wu, "Visualizing deep learning-based radio modulation classifier," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 1, pp. 47–58, 2021.

[12] A. Berian, K. Staab, N. Teku, G. Ditzler, T. Bose, and R. Tandon, "Adversarial filters for secure modulation classification," *arXiv preprint arXiv:2008.06785*, 2020.

[13] M. Sadeghi and E. G. Larsson, "Adversarial attacks on deep-learning based radio signal classification," *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 213–216, 2018.

[14] Y. Lin, H. Zhao, Y. Tu, S. Mao, and Z. Dou, "Threats of adversarial attacks in dnn-based modulation recognition," in *IEEE Conference on Computer Communications (INFOCOM)*, 2020, pp. 2469–2478.

[15] D. Ke, Z. Huang, X. Wang, and L. Sun, "Application of adversarial examples in communication modulation classification," in *2019 International Conference on Data Mining Workshops (ICDMW)*, 2019, pp. 877–882.

[16] B. Flowers, R. M. Buehrer, and W. C. Headley, "Evaluating adversarial evasion attacks in the context of wireless communications," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1102–1113, 2020.

[17] M. Sadeghi and E. G. Larsson, "Physical adversarial attacks against end-to-end autoencoder communication systems," *IEEE Communications Letters*, vol. 23, no. 5, pp. 847–850, 2019.

[18] Y. Arjoune and S. Faruque, "Artificial intelligence for 5g wireless systems: Opportunities, challenges, and future research direction," in *10th Annual Computing and Communication Workshop and Conference (CCWC)*, 2020, pp. 1023–1028.

[19] Y. Sagduyu, Y. Shi, and T. Erpek, "Adversarial deep learning for over-the-air spectrum poisoning attacks," *IEEE Transactions on Mobile Computing*, 2019.

[20] Y. Liu, X. Chen, C. Liu, and D. Song, "Delving into transferable adversarial examples and black-box attacks," in *International Conference on Learning Representations (ICLR)*, 2017.

[21] F. Tramer, N. Papernot, I. Goodfellow, D. Boneh, and P. McDaniel, "The space of transferable adversarial examples," *arXiv preprint arXiv:1704.03453*, 2017.

[22] N. Papernot, P. McDaniel, and I. Goodfellow, "Transferability in machine learning: from phenomena to black-box attacks using adversarial samples," *arXiv preprint arXiv:1605.07277*, 2016.

[23] T. J. O'Shea, T. Roy, and T. C. Clancy, "Over-the-air deep learning based radio signal classification," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 168–179, 2018.

[24] S. Peng, H. Jiang, H. Wang, H. Alwageed, and Y.-D. Yao, "Modulation classification using convolutional neural network based deep learning model," in *26th Wireless and Optical Communication Conference (WOCC)*. IEEE, 2017, pp. 1–5.

[25] Y. Tu, Y. Lin, C. Hou, and S. Mao, "Complex-valued networks for automatic modulation classification," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 9, pp. 10 085–10 089, 2020.

[26] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Engineering Applications of Neural Networks*, C. Jayne and L. Iliadis, Eds. Cham: Springer International Publishing, 2016, pp. 213–226.

[27] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems (NeurIPS)*, 2012, pp. 1097–1105.

[28] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR)*, 2016, pp. 770–778.

[29] D. Hong, Z. Zhang, and X. Xu, "Automatic modulation classification using recurrent neural networks," in *2017 3rd IEEE International Conference on Computer and Communications (ICCC)*. IEEE, 2017, pp. 695–700.

[30] S. Hu, Y. Pei, P. P. Liang, and Y.-C. Liang, "Robust modulation classification under uncertain noise condition using recurrent neural network," in *2018 IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–7.

[31] S. Rajendran, W. Meert, D. Giustiniano, V. Lenders, and S. Pollin, "Deep learning models for wireless signal classification with distributed low-cost spectrum sensors," *IEEE Transactions on Cognitive Communications and Networking*, vol. 4, no. 3, pp. 433–445, 2018.

[32] M. Kulin, T. Kazaz, I. Moerman, and E. De Poorter, "End-to-end learning from spectrum data: A deep learning approach for wireless signal identification in spectrum monitoring applications," *IEEE Access*, vol. 6, pp. 18 484–18 501, 2018.

[33] S. Peng, S. Sun, and Y.-D. Yao, "A survey of modulation classification using deep learning: Signal representation and data preprocessing," *IEEE Transactions on Neural Networks and Learning Systems*, pp. 1–19, 2021.

[34] D. Han, Z. Wang, Y. Zhong, W. Chen, J. Yang, S. Lu, X. Shi, and X. Yin, "Evaluating and improving adversarial robustness of machine learning-based network intrusion detectors," *IEEE Journal on Selected Areas in Communications*, pp. 1–1, 2021.

[35] L. Zhang, S. Lambotharan, G. Zheng, B. A. Sadhan, and F. Roli, "Countermeasures against adversarial examples in radio signal classification," *IEEE Wireless Communications Letters*, pp. 1–1, 2021.

[36] D. Adesina, C.-C. Hsieh, Y. E. Sagduyu, and L. Qian, "Adversarial machine learning in wireless communications using rf data: A review," *arXiv preprint arXiv:2012.14392*, 2020.

[37] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu, "Towards deep learning models resistant to adversarial attacks," in *International Conference on Learning Representations*, 2018.

[38] R. Sahay, D. J. Love, and C. G. Brinton, "Robust automatic modulation classification in the presence of adversarial attacks," in *2021 55th Annual Conference on Information Sciences and Systems (CISS)*, 2021.

[39] J. M. Cohen, E. Rosenfeld, and J. Z. Kolter, "Certified adversarial robustness via randomized smoothing," in *ICML*, 2019, pp. 1310–1320.

[40] B. Kim, Y. E. Sagduyu, K. Davaslioglu, T. Erpek, and S. Ulukus, "Channel-aware adversarial attacks against deep learning-based wireless signal classifiers," *arXiv preprint arXiv:2005.05321*, 2020.

[41] S. Kokalj-Filipovic, R. Miller, N. Chang, and C. L. Lau, "Mitigation of adversarial examples in rf deep classifiers utilizing autoencoder pre-training," in *International Conference on Military Communications and Information Systems (ICMCIS)*, 2019, pp. 1–6.

[42] N. Akhtar and A. Mian, "Threat of adversarial attacks on deep learning in computer vision: A survey," *IEEE Access*, vol. 6, pp. 14 410–14 430, 2018.

[43] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[44] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, 2014.

[45] W. Xu, D. Evans, and Y. Qi, "Feature squeezing: Detecting adversarial examples in deep neural networks," *arXiv preprint arXiv:1704.01155*, 2017.

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

[47] A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *arXiv preprint arXiv:1607.02533*, 2016.

[48] N. Carlini and D. Wagner, "Towards evaluating the robustness of neural networks," in *2017 IEEE Symposium on Security and Privacy (SP)*, 2017, pp. 39–57.

[49] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," *arXiv preprint arXiv:1612.01474*, 2017.

[50] T. J. O'Shea and N. West, "Radio machine learning dataset generation with gnu radio," in *GNU Radio Conference*, vol. 1, no. 1, 2016.

[51] D. Freedman, R. Pisani, and R. Purves, "Statistics (international student edition)," *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*, 2007.