Unsupervised Specific Emitter Identification Method Using Radio-Frequency Fingerprint Embedded InfoGAN

Jialiang Gong, Xiaodong Xu, and Yingke Lei

Abstract—**Machine learning approaches are becoming increasingly popular to improve the efficiency of specific emitter identification (SEI). However, in most non-cooperative SEI scenarios, supervised and semi-supervised learning approaches are often incompatible due to the lack of labeled datasets. To solve this challenge, an unsupervised SEI framework is proposed based on information maximized generative adversarial networks (InfoGANs) and radio frequency fingerprint embedding (RFFE). To enhance individual discriminability, a gray histogram is first constructed according to the *bispectrum* extracted from the received signal before being embedded into the proposed framework. In addition to the latent class input and the RFFE, the proposed InfoGAN incorporates a priori statistical characteristics of the wireless propagation channels in the form of a structured multimodal latent vector to further improve the GAN quality. The probabilistic distribution of the *bispectrum* is derived in closed-form and the convergence of the InfoGAN is analyzed to demonstrate the influence of the RFFE. Numerical results indicate that the proposed framework consistently outperforms state-of-the-art algorithms for unsupervised SEI applications, both in terms of evaluation score and classification accuracy.**

Index Terms—**Specific emitter identification, generative adversarial network, unsupervised deep learning, radio frequency fingerprint, Nakagami-m.**

I. Introduction

SPECIFIC emitter identification (SEI) refers to the problem of identifying individual radio emitters from received signals [1]. Potential applications include auxiliary authentication and intrusion detection in military and civilian communications [2], [3], self-organized networking [4] and cognitive radio [5].

Classical supervised SEI schemes usually consist of two major stages: (i) the extraction of discriminable features from the received signal, known as radio frequency fingerprints (RFFs), and (ii) the classification of the features through a predefined fingerprint database. In the first stage, feature extraction can be applied either to the transient signal [6], [7] or to the steady-state signal [8]–[10]. In the second stage, a proper classifier is developed to identify individual devices using discriminative RFFs. Recently, Reising [11] and Williams [12] applied the multiple discriminant analysis/maximum likelihood (MDA/ML) classifier for high-dimensional feature classification, while Brik [13] introduced an advanced support vector machine (SVM) classifier. Deep learning (DL) has also recently demonstrated great potential in various wireless applications [15]–[18], resulting in improved performance compared to traditional techniques. When a large enough training dataset with high-quality labels is available, data-driven deep neural networks are able to retrieve abstract characteristics through multiple hidden layers with nonlinear activation functions. However, in most non-cooperative scenarios, such as electronic interception and radio surveillance, it is quite difficult to obtain sufficient labeled training data. In such a case, unsupervised SEI is preferred. However, existing data-driven approaches rarely consider the domain knowledge in the field of wireless communication, thereby leading to unstable convergence and unsatisfactory classification performance.

Motivated by the promising success of generative adversarial networks (GANs) [19], we propose an unsupervised SEI framework based on information maximized GANs (InfoGANs) [20] and using radio frequency fingerprint embedding (RFFE). Given the number of target emitters, we add a loss function to the GAN training that maximizes the mutual information between the generated data and the emitter class latent input, imitating the processing method in InfoGAN. However, unlike InfoGAN, the proposed framework referred to as RFFE-InfoGAN, employs both artificial RFF and a new latent input vector. The major contributions of this work can be summarized as follows:

- To enhance the individual discriminability of different emitters in a non-cooperative scenario, a gray histogram is constructed according to the *bispectrum* extracted from the received signal before being embedded into the proposed framework. Hereafter, the histogram feature vector is denoted by $\boldsymbol{p_T}$.
- To improve the GAN quality, in addition to the latent code $\boldsymbol{c}$ (i.e., the class latent input in this paper) and $\boldsymbol{p_T}$, the generative network (GAN generator) is structured as

Manuscript received January 24, 2019; revised June 7, 2019, September 7, 2019, November 18, 2019, and January 31, 2020; accepted February 25, 2020. Date of publication March 5, 2020; date of current version March 27, 2020. This work was supported in part by the National Natural Science Foundation of China under Grant 61271272, and in part by the National Key Research and Development Program of China under Grant 2018YFA0701603. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Matthieu R. Bloch. *(Corresponding author: Xiaodong Xu.)*

Jialiang Gong and Xiaodong Xu are with the Department of Electrical Engineering and Information Science, University of Science and Technology of China, Hefei 230026, China (e-mail: jl2012@mail.ustc.edu.cn; xdxu@ustc.edu.cn).

Yingke Lei is with the School of Electronic Countermeasure, National University of Defense Technology, Hefei 230037, China (e-mail: leiyingke@163.com).

Digital Object Identifier 10.1109/TIFS.2020.2978620

a deep neural network that takes a structured multimodal latent vector z as a new input. Note that z incorporates a priori knowledge of the channel distribution (*Nakagami-m* is adopted in this paper) to better match the underlying characteristics of real-world data.

- The probabilistic distribution of the *bispectrum* is derived in closed-form, and the convergence analysis is analyzed to demonstrate the influence of the introduced $\boldsymbol{p_T}$.
- Various experimental results on both simulated and real world data are presented. First, we evaluate our method in terms of network properties such as convergence. Compared to baselines (the original GAN and InfoGAN), RFFE-InfoGAN achieves a faster convergence speed and higher evaluation score for the GAN generator. Then, we measure the effectiveness of the proposed framework along several factors including SNRs, modulation types, data sizes, and RFFE-InfoGAN outperforms existing solutions in terms of classification accuracy.

The rest of this paper is organized as follows. Section II presents the background of the deep-learning-based SEI and summarizes related works. Section III provides the design and implementation details of the proposed RFFE-InfoGAN. Experimental results for both simulated and real-world datasets are provided in Section IV, followed by a conclusion in Section V.

II. System Architecture and Problem Statement

A. Background of DL-Based Non-Cooperative SEI

Without loss of generality, consider a receiver that performs SEI to intercept wireless signals. Only one of K communication emitters is assumed to be active at a time. In other words, we do not take into account the problem of signal mixture and assume that the signals from each emitter can be captured individually. Denote by $s_k(t)$ the transmitted signal for the kth emitter; the continuous-time received signal can thus be formulated as

$$r_k(t) = s_k(t) * h_k(t) + n_k(t), \quad k = 1, 2, \dots, K, \tag{1}$$

where $h_k(t)$ represents the equivalent channel impulse response between the kth transmitter and the receiver, $n_k(t)$ stands for additive white Gaussian noise with zero mean and variance σ_{nk}^2, and "$*$" denotes the convolution operation. After sampling $r_k(t)$, we obtain discrete-time samples denoted by $\boldsymbol{x} = \{x_1, x_2, \dots, x_N\}$, where N stands for the number of received signal samples.

Before proceeding to introduce SEI, we focus on three key requirements:

1) **DL network**: In non-cooperative scenarios, we do not know which emitter k the received sample x_i comes from, and supervised and semi-supervised learning methods are usually powerless to perform SEI due to the lack of labeled data. Therefore, a DL architectureis desirable for training with unlabeled data.
2) **RFF**: Simply using received data in the selected DL network does not result in satisfactory performance; radio frequency fingerprint embedding (RFFE) is performed to improve the individual discriminability. The RFF $\boldsymbol{p_T}$ extracted from $\boldsymbol{x}$ should retain information of the received signal to the greatest extent and be robust to noise.
3) **Combination of DL network and RFF**: After an appropriate DL network and RFF are selected, the manually extracted RFF should play an important role training the selected DL network. In other words, the RFF should participate in the parameter iteration as much as possible.

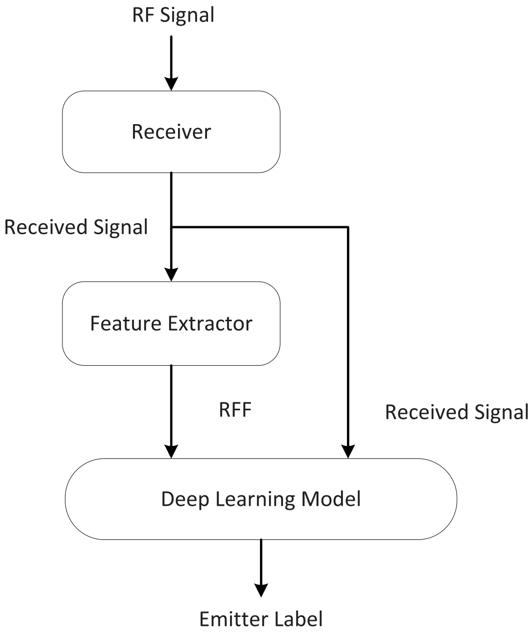


Fig. 1. Diagram of the DL based non-cooperative SEI.

In summary, after obtaining the received data $\boldsymbol{x}$, the RFF vector $\boldsymbol{p_T}$ of $\boldsymbol{x}$ is constructed by means of a feature extractor. Then, $\boldsymbol{x}$ and $\boldsymbol{p_T}$ are both used as inputs of an unsupervised DL model to perform emitter identification.

Figure 1 depicts a basic diagram of the domain knowledge incorporated and the DL-based non-cooperative SEI implemented in this paper. The primary function of the feature extractor is to extract RFFs from the captured signals before feeding them into the deep neural network. Note that the core operation of SEI is proposed by the DL module.

B. A Brief Description of GAN and Its Variants

GANs offer a promising unsupervised generative architecture for various tasks, including computer vision [21] and natural language processing [22]. Recent advances in GANs have shown their ability to learn complex data distributions. The core idea is to construct two networks, a generator (G) and a discriminator (D), as two players of a minimax game competing with each other, and aiming to make the generator G output distribution $p_G(x)$ match the distribution of real data $p_X(x)$ as closely as possible. Network training can be achieved via the adversarial action: G aims to approximate the real data distribution to fool D and D tries to distinguish the real data and the fake data generated by G. Ideally, G and D converge and reach a Nash equilibrium. Formally, we express G and D

as functions $G(z)$ and $D(x)$ that provide a mapping from the input space to the target space.

The target function of a GAN can be formalized as follows:

$$\min_{G}\max_{D} V(G, D) = \underset{\boldsymbol{x}\sim p_X(\boldsymbol{x})}{\mathbb{E}}[\log D(\boldsymbol{x})] + \underset{\boldsymbol{z}\sim p_Z(\boldsymbol{z})}{\mathbb{E}}[\log(1 - D(G(\boldsymbol{z})))], \quad (2)$$

where p_X denotes the distribution of real data and p_Z stands for a certain random distribution.

In recent years, many efforts have been directed to improving the performance through various changes to the original architecture: DCGAN [23] combined convolutional neural networks (CNNs) with GAN, and VEGAN [24] integrated a variational auto-encoder with GAN. Conditional GAN [25] and Triple-GAN [26] introduced conditions' information to the training while EBGAN [27] and W-GAN [28] modified the objective function to stabilize the training process.

GANs have recently been applied to many communication-related problems. For instance, O'Shea [29] proposed Comm-GAN to approximate the response function of a channel, Ray [30] applied the original GAN structure to extract RFFs for data augmentation and perform radio identification, and Zhu [31] utilized a variant of GAN for RF localization.

For the application of non-cooperative SEI, we choose an information-theoretic extended variant of GAN, called InfoGAN [20], as the basic DL model in this paper. By building upon the conditional variants of the original GAN, InfoGAN can learn interpretable and meaningful hidden representations by feeding G with both the original stochastic distribution $\boldsymbol{z}$ and a latent code $\boldsymbol{c}$ and then maximizing the mutual information between the generated data $G(\boldsymbol{z}, \boldsymbol{c})$ and the latent code $\boldsymbol{c}$. The latent code can be some prior features of the real data. For example, for the MNIST dataset, $\boldsymbol{c}$ can be sampled from a categorical distribution $Cat(K = 10, p = 0.1)$ that captures the 10 classes of handwritten digits in the dataset. Feeding this kind of latent code $\boldsymbol{c}$ into G enables the generated data to not only approximate the real data distribution but also to be separated into 10 categories; therefore, a classifier that is effective for the generated data can be applied to the real data, thereby achieving unsupervised classification.

III. The Proposed RF Fingerprint Embedded InfoGAN (RFFE-InfoGAN)

A. Proposed Framework

Without loss of generality, we consider the following radiation source identification problem. We have a set of received data $\boldsymbol{x} = \{x_1, x_2, \ldots, x_N\}$ and its corresponding RFF vector set $\boldsymbol{p_T} = \left\{p_{T_1}, p_{T_2}, \ldots, p_{T_N}\right\}$. We can also obtain a set of latent code $\boldsymbol{c} = \{c_1, c_2, \ldots, c_N\}$ by sampling from a categorical distribution $Cat(K, p = 1/K)$, where K is the number of emitters. Moreover, if further information about the distributions of the received data samples is available, the probability distribution p_z of the latent vector $\boldsymbol{z}$ can be determined. Overall, the underlying received signal distribution can be expressed as $p_{data}(\boldsymbol{x}\,|\boldsymbol{p_T})$, and the input latent distribution of G can be written as $p_z(\boldsymbol{z}\,|\boldsymbol{p_T}\ , \boldsymbol{c})$. We aim to identify the class to which every element in $\boldsymbol{x}$ belongs to by means of an unsupervised approach. The additional information provided by $\boldsymbol{p_T}$ can be merged into the input by concatenating $\boldsymbol{p_T}$ with the original input vector $\boldsymbol{z}$ or $\boldsymbol{x}$ to aid the unsupervised classification.

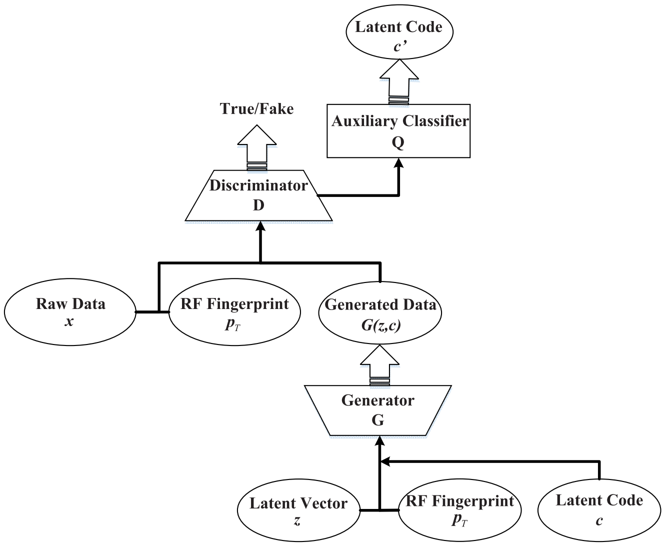


Fig. 2. The proposed framework leveraging InfoGAN with RFFs.

To perform unsupervised SEI, we develop a novel variant of GAN named RFFE-InfoGAN, which leverages the advantages of InfoGAN, manual RFFs, and the knowledge of the wireless propagation channel. The proposed framework is expected to learn a distribution that approximates the real communication signal and is closely related to the a priori information. Additionally, specific generated samples can be learned according to the latent code $\boldsymbol{c}$ so that the received signal can be unsupervisedly identified.

The RFFE-InfoGAN framework is comprised of three components:

a) Generative Model (G): Given a priori input latent code $\boldsymbol{c}$ and fingerprinting vector $\boldsymbol{p_T}$, G attempts to generate data $\hat{\boldsymbol{x}} = G(\boldsymbol{z}, \boldsymbol{c}) \sim p_G(\boldsymbol{z}\,|\boldsymbol{p_T}, \boldsymbol{c})$ to approximate the real received signal distribution $p_{data}(\boldsymbol{x}\,|\boldsymbol{p_T}\,)$ as closely as possible.

b) Discriminative Model (D): Given received data or generated data as inputs, D attempts to discriminate the received signal from generated signals. D acts as a binary classifier, assigning class label 1 to the received data vectors and 0 to those generated by G.

c) Auxiliary Classifier Model (Q): To maximize the mutual information between the generated data and the latent code $\boldsymbol{c}$, the authors in [20] introduced variational inference to transform the mutual information into a variational lower bound that is easy to compute. The calculation of this variational lower bound can be achieved by training Q. If the latent code $\boldsymbol{c}$ is sampled from a categorical distribution, then Q can be seen as a classifier for the labeled generated data. When the whole network converges, in other words, the generated data distribution approximates the real data, Q can be applied to the received data to obtain the classification results.

Similar to DCGAN [23], these three sub-networks use CNN structure. Figure 2 illustrates the proposed framework.

To this end, the global loss function of the RFFE-InfoGAN can be formulated as follows:

$$V(G,D,Q)=\min_{G,Q}\max_{D} V(G,D)+\lambda L(G,Q), \tag{3}$$

which consists of a modified GAN loss:

$$\begin{aligned} V(G,D) = {} & \mathbb{E}_{\boldsymbol{x}\sim p_{data},\boldsymbol{p_T}\sim p_{data}}[\log D(\boldsymbol{x}\,|\boldsymbol{p_T})] \\ & +\mathbb{E}_{\boldsymbol{z}\sim p_z,\boldsymbol{p_T}\sim p_{data},\boldsymbol{c}\sim p_c}[\log(1-D(G(\boldsymbol{z}\,|\boldsymbol{p_T},\boldsymbol{c})))], \end{aligned} \tag{4}$$

and a mutual information regularization term:

$$L(G,Q)=\mathbb{E}_{\hat{\boldsymbol{x}}\sim p_G(\boldsymbol{z}|\boldsymbol{p_T},\boldsymbol{c}),c'\sim p(\boldsymbol{c}|\hat{\boldsymbol{x}})}[\log Q(\boldsymbol{c'}\left|\hat{\boldsymbol{x}}\right.)]. \tag{5}$$

In the following, we elaborate on the process.

B. RF Fingerprinting Embedding Vector $\boldsymbol{p_T}$

The identification of different radiation sources depends on some inherent characteristics concealed behind the raw received signal. These features, such as design tolerance and nonlinearity of the hardware, asymmetry of the RF chains, and other impairments in the transmission process, are contained in the received signals. However, the underlying characteristics are difficult to learn using only the received signal as the input to the network. Manually extracted RFFs have been shown to be effective in the field of SEI. We propose to import proper RFFs into the deep neural network to help the network gain more distinctive information in terms of associations with particular emitters. Specifically, we present a new type of RFF to formulate $\boldsymbol{p_T}$ utilizing the *bispectrum* information.

SEI is also called emitter fingerprint identification, where the emitter considered in this paper refers to a communication transmitter and the fingerprints are features that reflect the difference between individual emitters. *Bispectrum*, a type of polyspectra, can be used to describe the degree of asymmetry and nonlinearity of signals and to measure the degree of signal deviation from a Gaussian distribution. Hence, to extract the fingerprint features, the *bispectrum* statistic is employed to reveal the cross-frequency dependence of the received signals. In general, the *bispectrum* $B_x(\omega_1,\omega_2)$ can be defined as:

$$B_x(\omega_1,\omega_2)=\sum_{\tau_1=-\infty}^{+\infty}\sum_{\tau_2=-\infty}^{+\infty} c_{3x}(\tau_1,\tau_2)e^{-j(\omega_1\tau_1+\omega_2\tau_2)}, \tag{6}$$

where $c_{3x}(\tau_1,\tau_2)$ represents the third-order cumulant, or skewness, of signal $x(n)$ with relative time delay τ_1 and τ_2, i.e.,

$$c_{3x}(\tau_1,\tau_2)=E\{\widehat{x}(n)x(n+\tau_1)x(n+\tau_2)\}. \tag{7}$$

$E\{\cdot\}$ stands for the expectation and $\widehat{x}$ denotes the conjugate operation on x.

To calculate $\boldsymbol{p_T}$, we first divide (ω_1,ω_2) into frequency bins with resolution $\Delta\omega$ and then record the amplitude and phase information of the complex matrix $B_x(\omega_1,\omega_2)$ as a new matrix $\boldsymbol{R}=[|B_x(\omega_1,\omega_2)|\,,\varphi_B(\omega_1,\omega_2)]$. We then convert the resulting $\boldsymbol{R}$ to a γ-bit greyscale image, as in [10], and finally compute its histogram vector $\boldsymbol{p_T}$.

C. Convergence Analysis

In this section, we provide a formal analysis of the convergence property of the RFFE-InfoGAN. We will prove that the network with objective function (3) can converge to its global optimum after taking the RFF vector as a new input.

After computing the *bispectrum* histogram $\boldsymbol{p_T}$, we concatenate it with the original network inputs $\boldsymbol{z}$ and $\boldsymbol{x}$ as new input vectors and then push the result to G and D, respectively. The new input vector can be seen as being sampled from the joint probability distribution of the original one and $\boldsymbol{p_T}$.

Lemma 1: If the distribution of the wireless channel is a Nakagami-m distribution:

$$p_{ch}(x)=\frac{2m^m x^{2m-1}}{\Gamma(m)\overline{p}^m}\exp(-\frac{mx^2}{\overline{p}}), \quad m\geq\frac{1}{2}, x\geq 0, \tag{8}$$

the probability density function of the amplitude $p_{|B|}$ and phase p_φ of the bispectrum for the Nakagami-m channel is given by:

$$\begin{aligned} p_{|B|}(x) &= \mathcal{E}x^{-\frac{1}{3}}e^{-\frac{x^{\frac{2}{3}}}{4C}}\sum_{k=0}^{\infty}S_k(jx^{\frac{1}{3}})^{n-k}\Gamma(\frac{k+1}{2},\frac{x^{\frac{2}{3}}}{4C}), \\ p_\varphi(x) &= \mathcal{F}\sum_{k=0}^{\infty}\varphi_k(jx)^{n-k}\Gamma(\frac{k+1}{2},\frac{x^2}{4C}), \end{aligned} \tag{9}$$

where $\mathcal{E}$ and $\mathcal{F}$ are multiplicative constants, S_k and φ_k represent coefficients related to k, $n=2m-1$ and $C=\frac{m}{\overline{p}}$.

Then, the two components in (9) are used to compute the gray histogram $\boldsymbol{p_T}$.

See Appendix A for a detailed proof.

The latent vector input $\boldsymbol{z}$ is sampled from a random distribution. Note that the sampling of $\boldsymbol{z}$ has no impact on the calculation of the RFF vector $\boldsymbol{p_T}$ since these are two separate processes, resulting to independent distributions. Therefore, the new input distributions are:

$$p_z=p_z(z,\boldsymbol{p_T})=p(z)\times p(\boldsymbol{p_T}), \tag{10}$$

$$p_{data}=p(\boldsymbol{x},\boldsymbol{p_T})=p(\boldsymbol{p_T})\times p(\boldsymbol{x}\,|\boldsymbol{p_T}). \tag{11}$$

As stated in Section III.A, the three subnetworks of RFFE-InfoGAN, namely, G, D and Q, use CNN structure. Using the theorem proposed by [48] for fully connected deep networks, we can conclude the following:

Lemma 2: For a deep CNN using gradient descent, if the number of hidden nodes per layer m_l and the step size α satisfy:

$$\begin{aligned} m_l &= \Omega(poly(n)2^{\mathrm{O}(L)}), \\ \alpha &= \mathrm{O}(\frac{poly(n)}{2^{\mathrm{O}(L)}}), \end{aligned} \tag{12}$$

where $\Omega()$ *and* $\mathrm{O}()$ *denote Big-*Ω *and Big-*O *notations, $poly(n)$ represents a polynomial about the convolution kernel number n and L is the number of layers, then for iteration $k=1,2,\ldots,$ we have,*

$$Loss(\theta(k))\leq(1-\epsilon)^k Loss(\theta(0)), \tag{13}$$

where ϵ denotes a positive constant less than 1 *and $Loss(\theta(k))$ is the loss function at the kth iteration.*

See Appendix B for a detailed proof.

For the CNN structure of our proposed network, the number of nodes per layer and the learning rate satisfy the condition described in (12). Therefore, all the subnetworks converge to their optimum and the whole network also converges to its global optimum. Next, we will show the convergence result for the whole structure.

Lemma 3: The optimal D of the game defined by the objective function (3) is:

$$D_{opt} = \frac{p_{data}(\boldsymbol{x}, \boldsymbol{p_T})}{p_g(\boldsymbol{x}, \boldsymbol{p_T}) + p_{data}(\boldsymbol{x}, \boldsymbol{p_T})}. \tag{14}$$

See Appendix C for a detailed proof.

Lemma 4: The minimum of G is achieved if and only if $p_g(\boldsymbol{x}, \boldsymbol{p_T}) = p_{data}(\boldsymbol{x}, \boldsymbol{p_T})$.

See Appendix D for a detailed proof.

Proposition 1: The network converges after taking the RFF vector as a new input.

Proof: First, the regularization item $L(G, Q)$ in the objective function (3) is ignored because the G parameters are fixed when Q is updated, which implies that adding $\boldsymbol{p_T}$ as an additional input to G has no influence on the convergence of Q. As a result, we only need to analyze the modified objective function (4).

Then, D_{opt}, the optimal D, should strike a balance between the received data distribution and generated data distribution defined by G, as summarized in Lemma 3.

Given D_{opt}, holding the parameters of D constant and updating G, the minimax game in (4) can be reformulated as $V(G, D) = \min_{G} V(G, D_{opt})$. Lemma 4 illustrates its optimum.

In summary, subnetworks G and D converge to their optima when updating one while holding the other constant. Finally, these subnetworks reach their equilibrium and converge. Therefore, we can conclude that adding RF fingerprinting vector $\boldsymbol{p_T}$ does not affect the convergence of the network. ■

D. Distribution of Latent Input Vector z

For datasets with interclass differences, the representations of each class are usually different. Intuitively, this implies that the underlying data distributions we attempt to learn by G are also different. Inspiring results from GMGAN [32] show that the network can achieve superior performance by changing the input latent vector $\boldsymbol{z}$ of G from a fixed distribution to a multimodal one. If we have some a priori information about the received signal, we can construct a multimodal probability distribution as the distribution of $\boldsymbol{z}$ instead of using the original standard normal distribution. The new $\boldsymbol{z}$ can not only match the characteristics of received data better but also improve the final identification performance. Therefore, we introduce a popular distribution of wireless propagation channels as the new distribution of $\boldsymbol{z}$. In this way, the latent vector $\boldsymbol{z}$, as the input of the G network is in line with our application scenario.

In wireless communications, the received signal often suffers from multipath fading channels. Multipath fading is derived from the constructive and destructive combination of randomly delayed, reflected, scattered, and diffracted signal components. This type of fading is relatively fast and is therefore responsible for the short-term signal variation in our scenario. Depending on the nature of the radio propagation environment, different models can be used to represent the statistical behavior of the multipath fading envelope. We choose the *Nakagami-m* distribution [33], which can generally represent the statistical behavior of the multipath small-scale fading envelope and accurately express the characteristics of most channels. Then, the new multimodal latent vector $\boldsymbol{z}$ can be expressed as the sum of K subdistributions:

$$p_z(\boldsymbol{z}) = \sum_{i=1}^{K} c_i \, p_i(\boldsymbol{z}), \tag{15}$$

where $\{c_i\}$ $(i = 1, 2, \ldots, k)$ denotes a categorical random variable, i.e., the latent class c, to determinate the subdistribution from which the latent vector should be sampled. $p_k(\boldsymbol{z})$ denotes a *Nakagami-m* distribution given by:

$$p_k(\boldsymbol{z}) = \frac{2m^m \boldsymbol{z}^{2m-1}}{\Gamma(m)\overline{p}_k{}^m} \exp(-\frac{m\boldsymbol{z}^2}{\overline{p}_k}), \tag{16}$$

where m is the *Nakagami-m* fading parameter that ranges in $[\frac{1}{2}, \infty)$. $m = \frac{1}{2}$ correspond to one-sided Gaussian distribution, $m = 1$ correspond to the standard Rayleigh distribution and, in the limit of $m \to +\infty$, the *Nakagami-m* channel converges to an additive white Gaussian noise (AWGN) nonfading channel. Moreover, when $\frac{1}{2} < m < 1$ and $m > 1$, the distribution corresponds to a Hoyt channel and Rice channel, respectively. Γ stands for the *Gamma* function, and $\overline{p}_k$ denotes the average power of the kth signal.

When applying the *Nakagami-m* distribution to the latent vector, the parameters m and $\overline{p}_k$ are first roughly estimated. An approximate value of m can be obtained by $E^2(\boldsymbol{z}^2)/\mathrm{var}(\boldsymbol{z}^2)$ while $\overline{p}_k$ can be estimated by means of power estimation algorithms. According to the experimental results in Section IV, the performance of our proposed framework varies with different values of these critical parameters. Appropriate values improve the identification performance. For simplicity, we assume that values of m and $\overline{p}_k$ are fixed after the rough estimation.

E. Implementation Details

The full training procedure for our proposed framework is described in Algorithm 1.

In practice, due to the high computational complexity of GANs, training may become difficult to converge or even collapse. Hence, a series of tricks should be employed to avoid these situations.

1) Following the approach used in [19], we change the loss function of the G network from $-\log(1 - D(\hat{\boldsymbol{x}}))$ to $-\log(D(\hat{\boldsymbol{x}}))$, to avoid gradient vanishing problem caused by the case in which the D network may perfectly distinguish fake samples.
2) Label smoothing, which can be used to avoid extreme logistic values is a good regularization method [34]. Applying label smoothing in our approach can, to some

Algorithm 1 Training of RFFE-InfoGAN Model

```
Require:
    $K$: The number of emitter classes;
    $iter$:The number of training iterations;
    $b_G$, $b_{DQ}$: Batch sizes for networks $G$ and $D/Q$;
    $\alpha$: Learning rate;
    $m$: Nakagami-m fading parameter;
    $\overline{p}$: Average signal power;
    $\theta_G$, $\theta_D$, $\theta_Q$: Parameters of networks $G$, $D$, $Q$
    $Adam$: The optimizer algorithm
for $i = 1$ to $K$ do
    $\overline{p}_k$ =The average power of each signal
end for
for $j = 1$ to $iter$ do
    for $k = 1$ to $b_{DQ}$ do
        $\boldsymbol{x_k}, \boldsymbol{p_T} \sim p_{data}$    ◇ obtain a real sample
        $\boldsymbol{c_k} \sim p_c$    ◇ obtain a latent code
        $\boldsymbol{z_k} \sim p_z(m, \boldsymbol{c_k})$    ◇ obtain a latent vector
        $\boldsymbol{\hat{x}_k} \leftarrow G(\boldsymbol{z_k} | \boldsymbol{p_T}, \boldsymbol{c_k})$    ◇ generate a fake sample
        Freeze $\theta_G$
        $L_D^k = -(\log(D(\boldsymbol{x_k})) + \log(1 - D(\boldsymbol{\hat{x}_k})))$
            ◇ compute the loss of $D$
        $L_Q^k = \boldsymbol{c_k} \log(Q(\boldsymbol{\hat{x}_k}))$
            ◇ compute the loss of $Q$
    end for
    $L_D = \frac{1}{2*b} \sum_{k=1}^{b} L_D^k$
    $\theta_D \leftarrow Adam(\nabla_{\theta_D}, L_D, \alpha, \theta_D)$
        ◇ update the weights of $D$
    $L_Q = \frac{1}{b} \sum_{k=1}^{b} L_Q^j$
    $\theta_Q \leftarrow Adam(\nabla_{\theta_G}, L_Q, \alpha, \theta_Q)$
        ◇ update the weights of $Q$
    for $r = 1$ to $b_G$ do
        Freeze $\theta_D, \theta_Q$
        $\boldsymbol{c_r} \sim p_c$    ◇ obtain a latent code
        $\boldsymbol{z_r} \sim p_z(m, \boldsymbol{c_r})$    ◇ obtain a latent vector
        $\boldsymbol{\hat{x}_r} \leftarrow G(\boldsymbol{z_r} | \boldsymbol{p_T}, \boldsymbol{c_r})$    ◇ generate a fake sample
        $L_G^r = -\log(1 - D(\boldsymbol{\hat{x}_r}))$    ◇ compute the loss of $G$
    end for
    $L_G = \frac{1}{b} \sum_{r=1}^{b} L_G^r$
    $\theta_G \leftarrow Adam(\nabla_{\theta_G}, L_G, \alpha, \theta_G)$
        ◇ update the weights of $G$
end for
```

extent, suppress the vulnerability of the adversarial training in GAN. The process can be expressed as follows:

$$[L_{real}^D(\boldsymbol{x}), L_{fake}^D(\boldsymbol{x})] = \begin{cases} [\eta_1, 1 - \eta_1], & \boldsymbol{x} \sim p_g \\ [1 - \eta_2, \eta_2], & \boldsymbol{x} \sim p_{data}, \end{cases} \tag{17}$$

where $L_{real/fake}^D$ denotes the label of real/fake (i.e., received/generated) data in D and η_1, η_2 denote constants in the range of [0, 1]. We set $\eta_1 = 0$ and $\eta_2 > 0$ to apply label smoothing to only the received samples because applying label smoothing to fake samples may lead to a fake mode in the data distribution, resulting in model obscure.

3) To overcome model collapes, we use the following two techniques simultaneously. First, a variant of the *batch-norm* mentioned in [35] is applied. By dividing the whole dataset into batches separately in terms of the received dataset and generated dataset, rather than a mixed one, we introduce gradient coordination to prevent network collapse. Additionally, we imitate the approach used in DCGAN [23] and add a random noise vector to each layer of G to overcome model collapse by introducing variants into the generating process.

IV. Numerical Results

In this section, we evaluate the performance of RFFE-InfoGAN and illustrate its benefits and improvements comparison to other baselines, including original GAN and InfoGAN. First, the convergence of the network is verified. Then, we compare its identification performance to that of the state-of-the-art approaches.

The experiments are conducted on two datasets:

1) Simulated Data: The simulated dataset is generated via *Matlab*. The modulation formats include binary phase-shift keying (BPSK), quadrature phase-shift keying (QPSK), and quadrature amplitude modulation of order 16 (16-QAM). The carrier frequency is set to 2000 MHz, with a sampling frequency of 5MHz and various oversampling rates. Similar to [41], we use the Taylor polynomial to model different power amplifiers. The coefficients of the model are chosen as detailed in Appendix E. We set the number of training and test samples to $N_{tr} = 10000$ and $N_{te} = 1000$ for each class. Additionally, the signal is passed through various signal-to-noise ratios (SNRs) and channel situations include AWGN, Rayleigh fading, and Rice fading channels.

2) Real-World Data: The real-world signals used in the experiments are collected as IQ data from shortwave, ultra-shortwave, and industrial scientific medical (ISM) bands. For data collected from radio stations, 5 different shortwave and 5 different ultra-shortwave stations of the same manufacturer, factory batch, and working mode act as objective emitters. Moreover, we sample from several dissimilar situations. For shortwave radio stations, considering the propagation properties and actual situations of these radio stations, we vary the communication distances and modulation types. For ultra-shortwave radio stations, the collection conditions are set as direct and diffracted, indoor and outdoor, and the modulation type is fixed to frequency modulation (FM). Acquisition parameters are shown in Table I. For the Wi-Fi scenario, we use 5 single antenna access points (APs) of TP-Link TL-WR740 as target emitters. Data are collected indoor under different conditions, such as frequency flat fading and selective fading. See Table II for specific parameter settings.

In our experiments, the SEI network architecture is built on Python 3.6 in Tensorflow, and the network is trained on 4 NVIDIA GeForce GTX 1080 Ti GPUs using the Adam optimizer [36]. The configuration details of the deep neural network are listed in Table III and the architecture details are given in Table IV.

TABLE I

ACQUISITION PARAMETERS OF RECEIVED SIGNALS

Radio station type	Shortwave	Ultra-shortwave
Bandwidth (MHz)	0.1	0.15
Modulation	USB/AM	FM
Sampling frequency (MHz)	0.32	0.48
Sampling points	3.84e7	5.76e7

TABLE II

ACQUISITION PARAMETERS OF WIFI SIGNALS

Brand and model	TP-Link TL-WR740
Wireless protocol	802.11g
Bandwidth (MHz)	20
Modulation	Coded OFDM
Sampling frequency (GHz)	10

TABLE III

NETWORK CONFIGURATIONS

Dataset	**Simulated**	**Real World**
Number of classes (K)	5	5
Training set size	5e4	5e4
Learning rate (α)	1e-3	5e-4
Histogram bins (γ)	64	128
Iterations	200	250

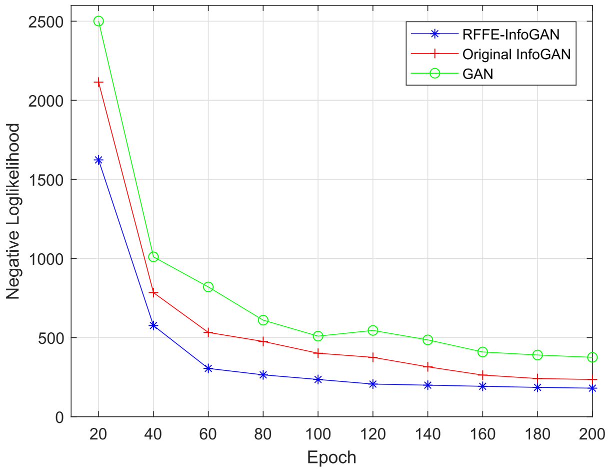


Fig. 3. Convergence performance of RFFE-InfoGAN and baselines.

A. Network Properties

Simulated datasets are used to analyze the properties of our proposed architecture.

1) Convergence: First, we show the impact of adding RFFs to the basic InfoGAN structure. The negative log-likelihood of the samples is chosen as a metric to evaluate the convergence performance of different models. The negative log-likelihood, which represents how closely the generated data are to the real data, is an effective metric of GAN convergence. The negative log-likelihood can be formulated as:

$$l_t = -\sum_{i=1}^{N} \boldsymbol{x_i} \log G(z_i \,|\boldsymbol{p_{Ti}}, \boldsymbol{c_i}\,), \tag{18}$$

where $\boldsymbol{x_i}$ denotes received data samples and $G(z_i\,|\boldsymbol{p_T}, \boldsymbol{c_i}\,)$ stands for the output of G at iteration t, i.e., the fake data sample.

Figure 3 illustrates the negative log-likelihood of samples as a function of the training epochs. RFFE-InfoGAN converges much faster than the baselines and can reach a better convergence plane.

2) Evaluation Scores: We now evaluate the generating performance of RFFE-InfoGAN. The evaluation score described below is used to measure and compare the generating performance of different models.

The quality of the samples generated by G can be evaluated by means of the methods proposed in [32], [37]. This indicator relies on a classifier that is pretrained on real datasets and the evaluation score can be divided into the following two parts:

a) Quality: An intermediate representation of the generated data $\boldsymbol{x}$ can be calculated via the pretrained classifier, and the quality of the generated data can be represented by the Euclidean distance between $\boldsymbol{x}$ and its nearest neighbor. If $y_l(\boldsymbol{x})$ denotes the activation output of the lth layer of the pretrained classifier given the input $\boldsymbol{x}$ and $N(\boldsymbol{x}) = \arg\min_{\boldsymbol{x'}\in X} \left\| y_l(\boldsymbol{x}) - y_l(\boldsymbol{x'}) \right\|_2$ denotes the nearest neighbor of $\boldsymbol{x}$, the quality score can be defined as:

$$q = \sum_{\boldsymbol{x}\in X} \frac{1}{|X|}\left(1 - \frac{\exp(\|y_l(\boldsymbol{x}) - y_l(N(\boldsymbol{x}))\|_2)}{\exp(\|y_l(\boldsymbol{x}) - y_l(N(\boldsymbol{x}))\|_2) + a}\right), \tag{19}$$

where a denotes a constant in [0, 1] and the score is normalized using a method similar to $sigmoid$.

b) Diversity: The entropy of the sum of the one-hot classification vector $y(\boldsymbol{x})$ is employed as the diversity score, where $y(\boldsymbol{x})$ is the output of the pretrained classifier given the input $\boldsymbol{x}$. If the generated data are diverse, i.e., the generated data cover all classes, the theoretical entropy reaches the maximum. The diversity score is defined as:

$$d = \frac{1}{\log N} H(\frac{1}{|X|}\sum_{\boldsymbol{x}\in X} y(\boldsymbol{x})). \tag{20}$$

Moreover, a single score, rather than two separate score, is needed to evaluate the performance of GAN models; the overall evaluation score can be obtained as:

$$score = \sqrt{q * d}. \tag{21}$$

The evaluation scores of the baselines and our proposed RFFE-InfoGAN on different datasets are presented in Table V.

According to Table V, in all cases, RFFE-InfoGAN achieves the highest score. Note that by applying the new pattern presented for latent vector z we can improve the evaluation score. According to the results, adding $\boldsymbol{p_T}$ as an RFF input results in a higher score than that of the network without $\boldsymbol{p_T}$. In other words, adding $\boldsymbol{p_T}$ can help the network generate samples that have both a higher quality and a richer diversity, thereby leading to a higher classification accuracy.

To further measure the impact of the new pattern of latent vector z, we investigate the effect of *Nakagami-m* parameter m on the evaluation scores.

For the simulated datasets, we choose several classic wireless channels, such as AWGN, Rayleigh, and Rice, to test

TABLE IV
NETWORK STRUCTURE

Generator G	Discriminator D	
Input noise z concatenate with p_T and c	Input signal concatenate with p_T	
MLP 1024, ReLU, batch norm	1×19 conv, 128, lReLU, batch norm, stride 1, maxpooling 2	
MLP 2048, ReLU, batch norm	15 conv1D, 32, lReLU, batch norm, stride 1, maxpooling 2	
Reshape 8×256×1	11 conv1D, 8, lReLU, batch norm, stride 1	
1×11 deconv, 32, stride 2×1, ReLU, batch norm	Discriminator D	Auxiliary Classifier Q
1×15 deconv, 128, stride 2×2, ReLU, batch norm	global pool	global pool
1×19 conv, 1, stride 1, sigmoid, weight norm	MLP 128 ReLU	MLP 256 ReLU, MLP 128 ReLU, MLP 32 ReLU
	MLP 1, sigmoid	MLP 5 softmax

TABLE V
EVALUATION SCORES FOR DIFFERENT MODELS ON DIFFERENT DATASETS

Model	**Simulated**	**Shortwave**	**Ultra-short**	**Wi-Fi**
GAN	0.625	0.564	0.598	0.577
InfoGAN	0.738	0.684	0.701	0.671
RFFE-InfoGAN (origin z)	0.805	0.725	0.815	0.764
RFFE-InfoGAN	**0.876**	**0.812**	**0.831**	**0.828**

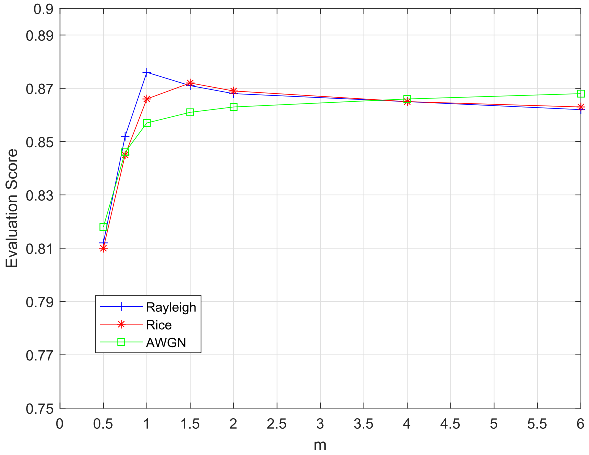


Fig. 4. Scores of RFFE-InfoGAN trained on different simulation datasets.

the performance. Figure 4 shows the impact on the evaluation scores for different m values of the latent vector. In consideration of generalization, we also test the impact of m on different real-world datasets, as demonstrated in Figure 5.

The plots show that the same model achieves quite different scores when applying different datasets and m. Note that the best score for each dataset is obtained with a different m. Therefore, if an accurate estimation algorithm is implemented to gauge m, the best m for each dataset easily be obtained. This approach could be considered in future work to improve the performance of RFFE-InfoGAN.

B. Classification Accuracy

In this subsection, we consider factors that can substantially affect the classification accuracy.

1) Classification Performance vs. SNR and Modulation Types: SNR and modulation types are the most common factors of the received signal itself and provide a simple starting point. Here, we compare the classification performance for different data modulation types as a function of SNR. Figure 6 shows the performance of different unsupervised schemes, including clustering methods such as K-means++ [38], DBSCAN [39] and baselines, and Figure 7 illustrates the performance of various input modulation types. AWGN channel and $K = 5$ are considered in this part of the experiment. For various input modulation types, RFFE-InfoGAN provides the best performance, and the classification accuracy can reach approximately 85%. Additionally at lower SNRs, the performance wins by a larger margin than at higher SNRs.

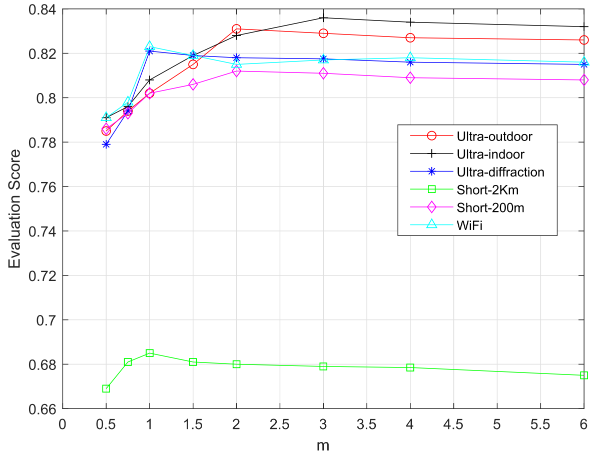


Fig. 5. Scores of RFFE-InfoGAN trained on different real-world datasets.

TABLE VI
CLASSIFICATION PERFORMANCE ON MULTIPLE MODULATION TYPES DATASETS WITH SNR = 10DB

Modulation	BPSK,QPSK	BPSK,16QAM	QPSK,16QAM
Accuracy	0.821	0.803	0.814

We also test whether multiple modulation types input can impact the classification performance. By mixing different modulation signals of an emitter as the received signal of the emitter, the scheme attempts to identify emitters through these mixture datasets with SNR = 10dB. The classification performance is shown in Table VI. Clearly, our proposed scheme does not rely on the input modulation type.

2) Classification Performance vs. Propagation Channels: The channel condition is another important factor that impacts

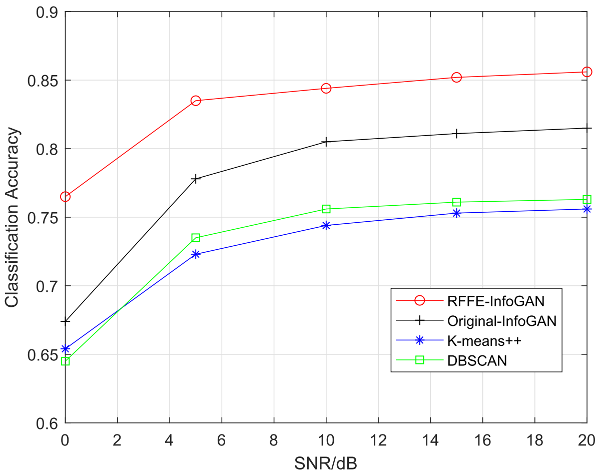


Fig. 6. Classification performance comparison for different unsupervised schemes under different SNRs.

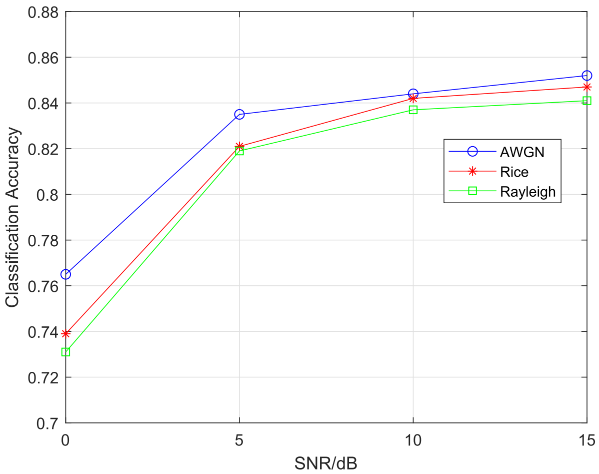


Fig. 8. Classification performance comparison of RFFE-InfoGAN on different propagation channels under different SNRs.

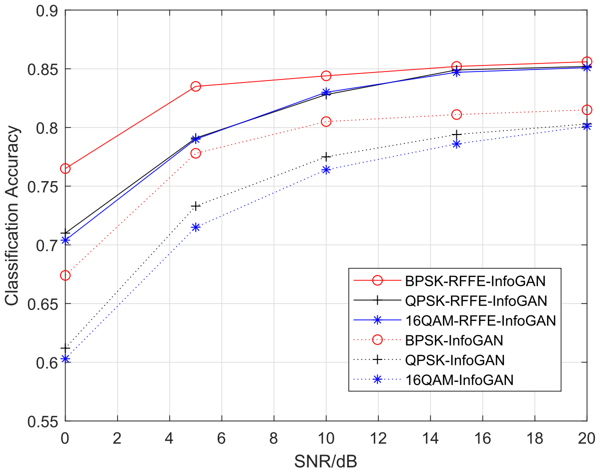


Fig. 7. Classification performance comparison of RFFE-InfoGAN and InfoGAN for different modulation types under different SNRs.

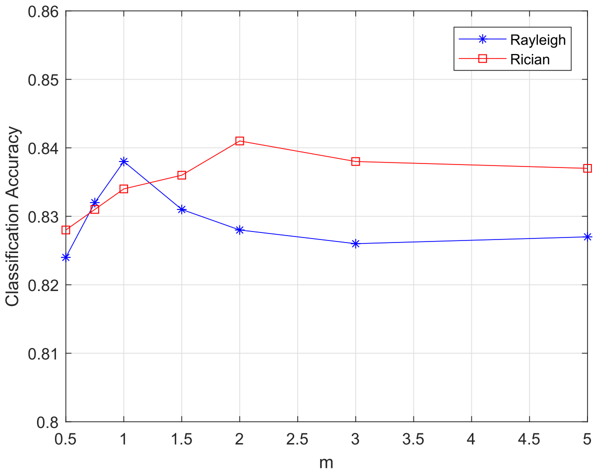


Fig. 9. Classification performance comparison of RFFE-InfoGAN on different propagation channels under different m.

wireless propagation. We choose several classic wireless channels (AWGN, Rayleigh, and Rician) to test our proposed scheme. BPSK modulation and $K = 5$ are considered in this part of experiment.

Figure 8 shows the classification accuracy of different channels as a function of the SNR. AWGN gives the best performance, and the Rayleigh fading channel has the worst performance. The classification accuracy of various channel situations can finally achieve approximately 85% at high SNRs.

As stated in Section IV-A.2, the value of the *Nakagami-m* parameter m impacts on the evaluation score for the network, which can further affect the classification performance. We test the influence of m with SNR = 10dB, as shown in Figure 9. The network achieves the best classification performance when m is chosen to match the wireless propagation channel of the received signal.

3) Classification Performance vs. Number of Emitters: The output of RFFE-InfoGAN is a set of predicted labels. With more classes, the number of parameters in the network must be larger. Figure 10 displays how the performance changes with the number of emitters. The experiments in this subsection consider BPSK modulation and an AWGN channel with SNR = 10dB.

The results show that the proposed RFFE-InfoGAN can classify up to 15 emitters with over 80% classification accuracy, although much better performance is achieved when identifying fewer classes. To improve the performance for a larger number of emitters, the network scale should be larger and the training method should be more elaborate, leading to a much higher computation cost. An effective algorithm to overcome this difficulty should be considered in future work.

4) Classification Performance vs. Oversampling Rate: The oversampling rate is another important factor in the simulation. We change the oversampling rate from 2 to 12 for BPSK and QPSK modulation in the AWGN channel with SNR = 10dB. Figure 11 shows the classification performance for different oversampling rates.

The results indicate that the proposed scheme is stable with respect to the oversampling rate, and better performance

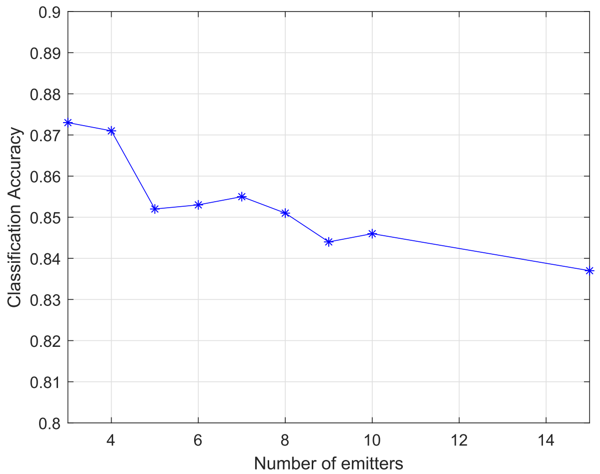


Fig. 10. Classification performance of RFFE-InfoGAN under different numbers of emitters.

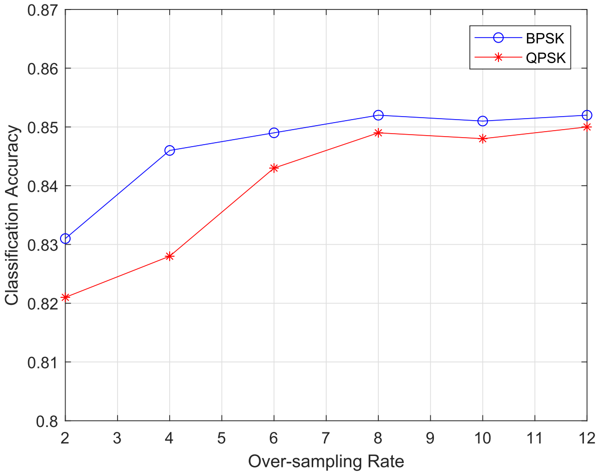


Fig. 11. Classification performance comparison of RFFE-InfoGAN on BPSK and QPSK under different oversampling rates.

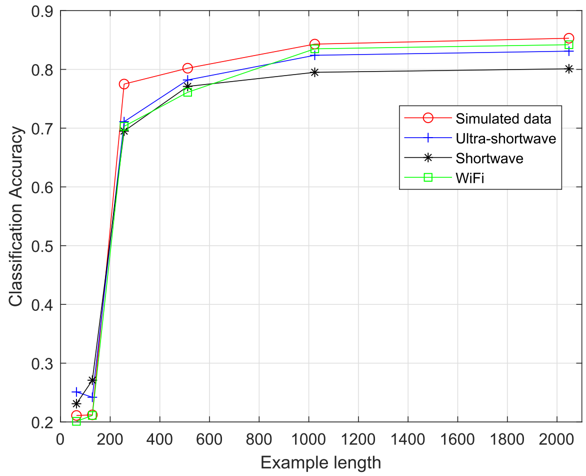


Fig. 12. Classification performance of RFFE-InfoGAN on different datasets under different example lengths l.

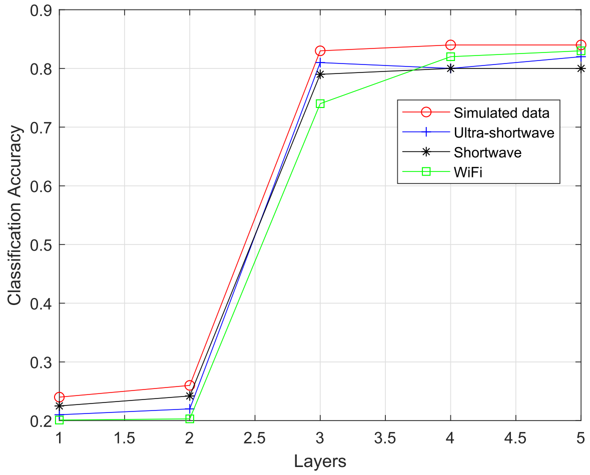


Fig. 13. Classification performance of RFFE-InfoGAN on different datasets under different numbers of layers.

is achieved for higher rates. To compromise between the accuracy and computational cost, we set the oversampling rate to 8 in the rest of the simulation experiments.

5) Classification Performance vs. Data Size: Machine learning methods are data-centric: the properties of the dataset itself often have enormous impacts on the final performance. In this part, we consider the influence of the length l of each input received sample. Figure 12 shows how the classification performance of the resulting model changes with l. In this experiment, we apply both the simulated data samples with SNR $= 20$dB, and real-world data samples.

The data sizes $(64, 128, 256, 512, 1024, and 2048)$ are used in this experiment. The performance is good for size of 512 and larger on all of the datasets. Considering the compromise of memory, training time, and classification accuracy requirements, we choose $l = 1024$.

6) Classification Performance vs. Network Depth: The number of model layers also has a substantial impact on the classification accuracy of deep networks. In RFFE-InfoGAN, the convolutional layers of G and D perform like a pair of inverse systems ignoring the dense layers. Therefore, the number of convolutional layers of G can indicate the depth of the network to some extent. Figure 13 shows the impact of the number of layers on the classification accuracy. Similar to Section IV.B2, we use both simulated data and real-world data in this part of the experiment.

In Figure 13, the classification performance becomes steady when the the number of layers is larger than 3. As a result, we choose 3 convolutional layers in G and D.

7) Classification Performance vs. Real-World Datasets: Without loss of generality, the proposed framework is tested on differentreal-world datasets. Some comparisons with classic approaches are necessary. As mentioned in Section IV-B.1, our proposed RFFE-InfoGAN outperforms traditional clustering methods in terms of unsupervised classification. Therefore, the comparisons should contrast the *bispectrum* histogram RFF input p_T with other RFFs under the REEF-InfoGAN structure. Recently, various RFF extraction methods have been proposed and shown to be effective. We choose several representative and recently proposed SEI methods, i.e., compressed *bispectrum* (CBS) [42], Hilbert-Huang Transform (HHT) [43],

TABLE VII
CLASSIFICATION ACCURACY COMPARISON OF DIFFERENT RFFS ON SHORTWAVE DATASETS

p_T	Histogram	CBS	HTT	EMD-EM^2	VMD-SF	NPE
AM Short-distance (200m)	**0.831**	0.601	0.627	0.693	0.721	0.688
USB Short-distance (200m)	**0.820**	0.588	0.604	0.688	0.719	0.641
USB Long-distance (2km)	**0.692**	0.471	0.497	0.520	0.511	0.503

TABLE VIII
CLASSIFICATION ACCURACY COMPARISON OF DIFFERENT RFFS ON ULTRA-SHORTWAVE DATASETS

p_T	Histogram	CBS	HTT	EMD-EM^2	VMD-SF	NPE
Indoor Direct(50m)	**0.834**	0.632	0.654	0.705	0.727	0.694
Outdoor Direct (200m)	**0.852**	0.645	0.667	0.711	0.738	0.713
Outdoor Diffraction (250m)	**0.736**	0.515	0.512	0.601	0.595	0.533

TABLE IX
CLASSIFICATION ACCURACY COMPARISON OF DIFFERENT RFFS ON WI-FI DATASETS

p_T	Histogram	CBS	HTT	EMD-EM^2	VMD-SF	NPE
Frequency flat fading (Indoor direct)	**0.807**	0.614	0.591	0.684	0.677	0.644
Frequency selective fading (Through walls)	**0.703**	0.432	0.441	0.501	0.514	0.529

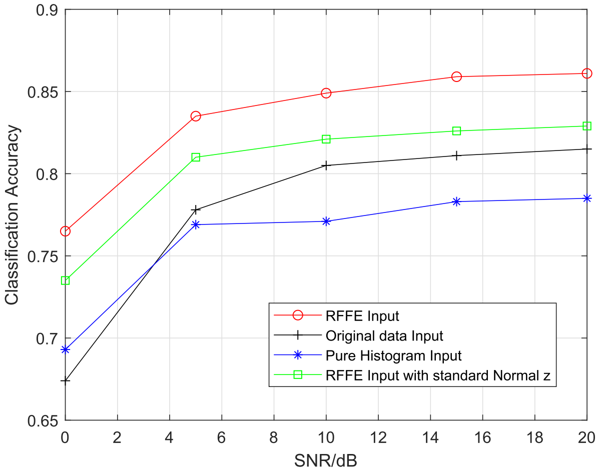


Fig. 14. The trajectories of the classification accuracy with different inputs.

EMD-EM^2 [10], VMD-SF [44], and normalized permutation entropy (NPE) [8], as the control group to illustrate the advantages of our proposed RFFE-InfoGAN. Tables VII to IX reveal the classification accuracy results obtained via different RFF extraction methods on shortwave, ultra-shortwave and Wi-Fi datasets, respectively.

The results show that the histogram of *bispectrum* has the best classification accuracy. Thus, the combination of RFFE-InfoGAN and the histogram RFF vector result in the best classification performance.

Additionally, various factors of the sampling situation impact the classification performance. For instance, signals sampled at different propagation distance lead to different performance. Moreover, at the same distance, the type of modulation affects the classification performance. In conclusion, we can see that various propagation environments lead to different network behavior.

8) Classification Performance vs. Input Modifications: Different input modifications to the baseline model InfoGAN are tested to assess their influence. Received signal slices, *bispectrum* histogram RFF vectors and the combination of both are used as the inputs. Furthermore, we test the impact of the new pattern on the latent vector z, i.e., standard normal z and *Nakagami-m* z. Figure 14 shows the performance results.

Clearly, our combined pattern achieves the best performance. The improvement results may come from the better convergence performance and the resulting evaluation score of the network, according to the results in Section IV.A.

V. CONCLUSION

In this paper, we develop RFFE-InfoGAN, an unsupervised DL SEI framework using radio frequency fingerprint embedding, to perform identification in noncooperative scenarios. Supervised or semisupervised SEI algorithms can typically not be applied under such severe environments. Therefore, based on InfoGAN, we integrate the merits of traditional feature extraction methods, knowledge of wireless signal propagation, and GAN to perform RFFE-InfoGAN.

The proposed framework takes two new inputs, an artificial RFF vector $\boldsymbol{p_T}$ based on the histogram of the *bispectrum* information from the received signal to enhance individual discriminability, and a structured multimodal latent vector z, which incorporates a priori knowledge of the fading channel distribution to match the properties of the received signal, to improve the GAN quality. Furthermore, we reveal a closed-form expression of the probability distribution of p_T and analyze of the global convergence of the network to demonstrate the influence of RFF $\boldsymbol{p_T}$. Various experiments on both the network properties and the classification performance have been conducted on different impact factors on simulated and real-world datasets. The experimental results show that our proposed RFFE-InfoGAN outperforms the existing unsupervised methods and the baseline models in terms of both network property metrics and classification accuracy. Therefore, the proposed framework is an effective SEI algorithm in noncooperative scenarios.

Moreover, the proposed framework can be generalized to other applications such as single-frequency network positioning [49], where particular RFF vector $\boldsymbol{p_T}$ and the multimodal structured latent vector $\boldsymbol{z}$ should be investigated according to the specific scenario. To further improve the performance, more accurate knowledge of the wireless signal should be provided and integrated into the network, and if possible, semi-supervised learning methods that can merge supervised information into the network should be applied.

Appendix A
Proof of Lemma 1

The computation of the RFF used in the paper, that is, the histogram of *bispectrum*, can be formulated as:

$$\begin{aligned} c_{3x}(\tau_1,\tau_2) &= E\{\bar{x}(n)x(n+\tau_1)x(n+\tau_2)\},\\ B_x(\omega_1,\omega_2) &= \sum_{\tau_1=-\infty}^{+\infty}\sum_{\tau_2=-\infty}^{+\infty} c_{3x}(\tau_1,\tau_2)e^{-j(\omega_1\tau_1+\omega_2\tau_2)}. \end{aligned} \tag{22}$$

Express the distribution of the received signal $x(t)$ as $p_{data}(x)$. After normalization, the distribution is transformed to have zero mean ($E[x(t)]=0$) and variance $E[x^2(t)] = \sigma_x{}^2$. The third-order moment can be then calculated as $E[x^3(t)] = \gamma_{3x}$. In a wireless propagation environment, the received signal can be denoted as:

$$r(t) = x(t) * h(t) + n(t), \tag{23}$$

where $h(t)$ stands for the channel impulse response with distribution $p_{ch}(h)$, $n(t)$ denotes Gaussian noise and "$*$" represents the convolution operation. Notably, the third-order cumulant for a Gaussian distribution equals 0, so we can ignore the impact of $n(t)$ when calculating the *bispectrum*. After transforming the received signal into the frequency domain, (23) becomes:

$$R(\omega) = X(\omega)H(\omega), \tag{24}$$

where $H(\omega)$ is the transfer function of the wireless propagation system. Then, we can easily obtain the *bispectrum* according to the above formulas.

$$B(\omega_1,\omega_2) = \gamma_{3x}H(\omega_1)H(\omega_2)\bar{H}(\omega_1+\omega_2). \tag{25}$$

Generally, we can express the amplitude and phase of the *bispectrum* signal as,

$$\begin{aligned} |B(\omega_1,\omega_2)| &= |\gamma_{3x}|\,|H(\omega_1)|\,|H(\omega_2)|\,|H(\omega_1+\omega_2)|,\\ \varphi_B(\omega_1,\omega_2) &= \varphi(\omega_1)+\varphi(\omega_2)-\varphi(\omega_1+\omega_2). \end{aligned} \tag{26}$$

Without loss of generality, we can set the distribution of the wireless channel $h(t)$ to be a *Nakagami-m* distribution:

$$p_{ch}(x) = \frac{2m^m x^{2m-1}}{\Gamma(m)\overline{p}^m}\exp(-\frac{mx^2}{\overline{p}}), \quad m \geq \frac{1}{2}, x \geq 0, \tag{27}$$

where m is the *Nakagami-m* parameter and $\overline{p}$ denotes the average power of the received signal. Take the following abbreviations: $\mathcal{A} = \frac{2m^m}{\Gamma(m)\overline{p}^m}$, $n = 2m-1$ and $C = \frac{m}{\overline{p}}$. After spectrum transformation, the probability density function of the transfer function $H(\omega)$ can be formulated as:

$$\begin{aligned} p_H &= \int_0^\infty e^{j\omega x}p_{ch}(x)dx = E(e^{j\omega x})\\ &= \mathcal{A}\int_0^\infty x^n e^{j\omega x - Cx^2}dx\\ &= \mathcal{A}e^{-\frac{\omega^2}{4C}}\int_0^\infty x^n e^{-(C^{\frac{1}{2}}x-\frac{j\omega}{2C^{\frac{1}{2}}})^2}\\ &= \mathcal{A}\frac{e^{-\frac{\omega^2}{4B}}}{2^n(C^{\frac{1}{2}})^{2n+1}}\int_0^\infty (2C^{\frac{1}{2}}+j\omega)^n e^{-u^2}du\\ &= M\int_0^\infty\sum_{k=0}^{\infty}\binom{n}{k}(2C^{\frac{1}{2}}u)^k(j\omega)^{n-k}e^{-u^2}du\\ &= -\frac{1}{2}M\sum_{k=0}^{\infty}\binom{n}{k}(2C^{\frac{1}{2}})^k(j\omega)^{n-k}\Gamma(\frac{k+1}{2},\mathcal{CW})\Big|_0^{+\infty}\\ &= \mathcal{A}\frac{e^{-\frac{\omega^2}{4C}}}{2^n(C^{\frac{1}{2}})^{2n+1}}\sum_{k=0}^{\infty}\binom{n}{k}(2C^{\frac{1}{2}})^k(j\omega)^{n-k}\Gamma(\frac{k+1}{2},-\frac{\omega^2}{4C}), \end{aligned} \tag{28}$$

where $\binom{n}{k} = \frac{n(n-1)(n-2)\cdots(n-k+1)}{k!}$ denotes the extended binomial coefficient for $k \in N$, $\mathcal{CW} = (C^{\frac{1}{2}}x - \frac{j\omega}{2C^{\frac{1}{2}}})^2$ and $\Gamma(\cdot)$ represents the upper incomplete gamma function.

Then, the probability density function of the amplitude and phase of *bispectrum* for the *Nakagami-m* channel can be formulated as (29), as shown at the bottom of the next page, where $p_{H^3}(x)$ stands for the probability density function of $H^3(\omega)$.

These two components of the *bispectrum* are used to compute the statistical histogram and participate in the proposed algorithm as the RFF vector p_T.

Appendix B
Proof of Lemma 2

$\mathcal{N}(0,1)$ denotes a standard Gaussian distribution, and $\lambda_{\min}(\boldsymbol{M})$ denotes the smallest eigenvalue of a matrix $\boldsymbol{M}$ if it is positive semidefinite.

The basic architecture for a standard CNN can be stated as follows.

We have some training inputs $\{x_i\}_{i=1}^n$, references $\{y_i\}_{i=1}^n$ and loss function (30), as shown at the bottom of the next page. Suppose θ is the network parameter we aim to optimize over and f is the prediction function, i.e., the output of the network.

Let $x \in \mathbb{R}^{d_0\times a}$ be the vector form of the input, where d_o is the number of input channels, usually set to 1 in the structure of our proposed network, while a is the dimension of the input data. $\boldsymbol{W}^{(1)} \in \mathbb{R}^{d_1\times a}$ is the weight matrix of the first layer, and $\boldsymbol{W}^{(l)} \in \mathbb{R}^{d_l\times qd_{l-1}}$ is the weight matrix of the lth layer for $2 \leq l \leq L$. $\sigma(\cdot)$ is the activation function. For layer l, we let the number of channels be d_l and $\boldsymbol{x}^{(l-1)} \in \mathbb{R}^{d_{l-1}\times a_{l-1}}$ be the input. The convolution operation can be expressed as an affine transformation. An operator $\varphi(\cdot) \in \mathbb{R}^{qd_{l-1}\times a_{l-1}}$ is used to map the input. For example, when the stride is 1 and convolution kernel $q=5$, the map matrix is formulated as (31), as shown at the bottom of the next page.

Define $\boldsymbol{x}^{(l-1)}_{:,0} = \boldsymbol{x}^{(l-1)}_{:,a_{l-1}+1:a_{l-1}+4} = \boldsymbol{0}$ as zero-padding and in the analysis, we view q as a constant.

Thus, the prediction function can be defined recursively as:

$$\begin{aligned} \boldsymbol{x}^{(0)} &= \boldsymbol{x}, \\ \boldsymbol{x}^{(l)} &= \sqrt{s_\sigma}\,\sigma(\boldsymbol{W}^{(l)}\varphi_l(\boldsymbol{x}^{(l-1)})), \quad 1 \le l \le L, \\ f_{cnn}(\boldsymbol{x}, \theta) &= \boldsymbol{a}^T \boldsymbol{x}^{(L)}, \end{aligned} \tag{32}$$

where s_σ is a scaling factor to normalize the input of every layer and is usually defined as $E_{x\sim\mathcal{N}(0,1)}[\sigma^2(x)]^{-1}$. $a \in R^{m_L}$ is the weights for the output layer.

The training of the CNN can be achieved via a randomly initialized gradient descent algorithm to find the global optimum of the loss function (30). The random initialization scheme we used is He-normal [46]. For every layer $l \in \{1, 2, \dots L\}$, each entry in $\boldsymbol{W}^{(l)}$ is sampled from a standard Gaussian distribution, i.e., $\boldsymbol{W}_i j^{(l)} \sim N(0, 1)$. Then the update rule can be formulated as:

$$\begin{aligned} \boldsymbol{W}^{(l)}(k) &= \boldsymbol{W}^{(l)}(k-1) - \alpha \frac{\partial L(\theta(k-1))}{\partial \boldsymbol{W}^{(l)}(k-1)}, \\ \boldsymbol{a}(k) &= \boldsymbol{a}(k-1) - \alpha \frac{\partial L(\theta(k-1))}{\partial a(k-1)}, \end{aligned} \tag{33}$$

where $\alpha > 0$ is the learning rate.

Substituting known variables, the gradient for the network can be calculated as (34), as shown at the bottom of the this page.

Inspired by [47], the convergence of the neural network is determined by the smallest eigenvalue of the Gram matrix introduced by the architecture. The main idea is summarized below:

The individual prediction for input x_i at the kth iteration is

$$t_i(k) = f(\theta(k), x_i). \tag{35}$$

Define $\boldsymbol{t}(k) = (t_1(k), t_2(k), \dots, t_n(k))^T$ as the sequence form for $t_i(k)$. Modeling as proposed by [47], we expand the conclusion into our deep neural network setting. The sequence $\{\boldsymbol{y} - \boldsymbol{t}(k)\}_{k=0}^{\infty}$ meets the following recursively:

$$\boldsymbol{y} - \boldsymbol{t}(k+1) = (\boldsymbol{I} - \alpha \boldsymbol{G}(k))(\boldsymbol{y} - \boldsymbol{t}(k)), \tag{36}$$

where

$$\begin{aligned} \boldsymbol{G}_{ij}(k) &= \left\langle \frac{\partial t_i(k)}{\partial \theta(k)}, \frac{\partial t_j(k)}{\partial \theta(k)} \right\rangle \\ &= \sum_{l=1}^{L} \left\langle \frac{\partial t_i(k)}{\partial \boldsymbol{W}^{(l)}(k)}, \frac{\partial t_i(k)}{\partial \boldsymbol{W}^{(l)}(k)} \right\rangle + \left\langle \frac{\partial t_i(k)}{\partial \boldsymbol{a}(k)}, \frac{\partial t_i(k)}{\partial \boldsymbol{a}(k)} \right\rangle \\ &= \sum_{l}^{L} \boldsymbol{G}^{(l)}_{ij}(k). \end{aligned} \tag{37}$$

$\langle, \rangle$ denotes the standard Euclidean inner product between two vectors or matrices.

For CNN, $\boldsymbol{G}^{(l)}$ has the following form (38), as shown at the top of the next page.

If the number of nodes in each layer of the neural network(a_l) is sufficiently large, for all $k \in \{1, 2, \dots K\}$, $\boldsymbol{G}^{(l)}(k)$ is closed to a fixed matrix $\boldsymbol{R}^{(l)} \in \mathbb{R}^{n\times n}$. $\boldsymbol{R}^{(l)}$ depends on only the input training data and the architecture of the neural network and has *no* dependence on the weight parameters θ. As a result, the dynamic of $\{\boldsymbol{y} - \boldsymbol{t}(k)\}_{k=1}^{\infty}$ approximates to a linear formula:

$$\boldsymbol{y} - \boldsymbol{t}(k+1) \approx (\boldsymbol{I} - \alpha \boldsymbol{R}(k))(\boldsymbol{y} - \boldsymbol{t}(k)). \tag{39}$$

Hence, the sequence converges to 0 for appropriate α and $\boldsymbol{R}(k)$, leading to a global optimum for the neural network.

The definition of $\boldsymbol{R}(k)$ and the convergence result for the CNN are similar to the theorems proposed for deep fully connected neural network in [48]. By replacing $x^{(l)}$, the output to the lth layer, $\varphi_l(x^{(l-1)})$, and following the proof in [48], we can draw a similar conclusion:

If the number of hidden nodes per layer and the step size satisfy:

$$\begin{aligned} m_l &= \Omega(poly(n)2^{\mathrm{O}(L)}), \\ \alpha &= \mathrm{O}(\frac{poly(n)}{2^{\mathrm{O}(L)}}), \end{aligned} \tag{40}$$

$$\begin{aligned} p_{|B|}(x) &= |\gamma_{3x}|\, p_{H^3}(x) = \frac{1}{3} x^{-\frac{1}{3}} \mathcal{A} \frac{e^{-\frac{x^{\frac{2}{3}}}{4C}}}{2^{n+1}\left(C^{\frac{1}{2}}\right)^{2n+1}} \sum_{k=0}^{\infty} \binom{n}{k} (2\sqrt{B})^k (jx^{\frac{1}{3}})^{n-k} \Gamma(\frac{k+1}{2}, -\frac{x^{\frac{2}{3}}}{4C}) \\ p_\varphi(x) &= p_H(x). \end{aligned} \tag{29}$$

$$\min_{\theta} Loss(\theta) = -\frac{1}{N} \sum_{i=1}^{n} y_i \log(f(x_i, \theta)) + (1 - y_i) \log(1 - f(x_i, \theta)). \tag{30}$$

$$\varphi_l(\boldsymbol{x}^{(l-1)}) = \begin{pmatrix} (\boldsymbol{x}^{(l-1)}_{1,0:4})^T & \cdots & (\boldsymbol{x}^{(l-1)}_{1,a_{l-1}-1:a_{l-1}+4})^T \\ \vdots & \ddots & \vdots \\ (\boldsymbol{x}^{(l-1)}_{d_{l-1},0:4})^T & \cdots & (\boldsymbol{x}^{(l-1)}_{d_{l-1},a_{l-1}-1:a_{l-1}+4})^T \end{pmatrix}. \tag{31}$$

$$\frac{\partial L(\theta)}{\partial \boldsymbol{W}^{(l)}} = (s_\sigma)^{\frac{L-l+1}{2}} \sum_{i=1}^{n} [(f(x_i, \theta) - y_i) x_i^{l-1} \boldsymbol{a}'(\prod_{k=l+1}^{L} \sigma'(\boldsymbol{W}^{(k)^T} x_i{}^{(k-1)}) \boldsymbol{W}^{(k)})]. \tag{34}$$

$$\boldsymbol{G}_{ij}^{(l)} = s_\sigma \sum_{r=1}^{a_l} [\sum_{k=1}^{d_l} \varphi_l(x_{i,k}^{(l-1)})\sigma'((\boldsymbol{w}_k^{(l)})^T x_{i,k}^{(l-1)})]^T [\sum_{h=1}^{d_l} \varphi_l(x_{j,h}^{(l)})\sigma'((\boldsymbol{w}_h^{(l)})^T x_{j,h}^{(l-1)})]. \tag{38}$$

$$\begin{aligned} V(D) &= \iint\limits_{\boldsymbol{x},\boldsymbol{p_T}} p_{data} \times \log D(\boldsymbol{x}\,|\boldsymbol{p_T})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{p_T} + \iint\limits_{z,\boldsymbol{p_T}} p(z) \times p(\boldsymbol{p_T}) \times \log(1 - D(G(z\,|\boldsymbol{p_T}, \boldsymbol{c})))\mathrm{d}z\mathrm{d}\boldsymbol{p_T} \\ &= \iint\limits_{\boldsymbol{x},\boldsymbol{p_T}} p_{data} \times \log D(\boldsymbol{x}\,|\boldsymbol{p_T})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{p_T} + p_g \times \log(1 - D(\boldsymbol{x}\,|\boldsymbol{p_T}))\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{p_T}, \end{aligned} \tag{42}$$

$\Omega()$ and O() denote Big-Ω and Big-O notations and $poly(n)$ represents a polynomial about the kernel number n, then, for $k = 1, 2, \ldots$, we have,

$$Loss(\theta(k)) \leq (1 - \alpha \cdot c)^k Loss(\theta(0)), \tag{41}$$

where c is a constant related to $\lambda_{\min}(\boldsymbol{R}^{(l)})$, the smallest eigenvalue of $\boldsymbol{R}^{(l)}$.

Appendix C
Proof of Lemma 3

If the parameters of G are fixed, taking p_T as an additional input, the training goal of D is to maximize (42), as shown at the top of the this page, where

$$\begin{aligned} p_{data}(\boldsymbol{x}, \boldsymbol{p_T}) &= p(\boldsymbol{p_T}, \boldsymbol{x}) = p(\boldsymbol{p_T}) \times p(\boldsymbol{x}\,|\boldsymbol{p_T}), \\ p_g(\boldsymbol{x}, \boldsymbol{p_T}) &= p_z(G^{-1}(\boldsymbol{x})) \times G^{-1'}(\boldsymbol{x}) \times p(\boldsymbol{p_T}). \end{aligned}$$

Set $\boldsymbol{u} = (\boldsymbol{x}\,|\boldsymbol{p_T})$, to find the optimum for $D(\boldsymbol{u})$, we take the gradient with respect to $D(\boldsymbol{u})$:

$$\frac{\mathrm{d}}{\mathrm{d}D(\boldsymbol{u})}[p_{data} \times \log D(\boldsymbol{u}) + p_g \times \log(1 - D(\boldsymbol{u}))] = 0. \tag{43}$$

By solving this equation, $V(D)$ finally achieves its maximum at:

$$D_{opt}(\boldsymbol{x}, \boldsymbol{p_T}) = \frac{p_{data}(\boldsymbol{x}, \boldsymbol{p_T})}{p_g(\boldsymbol{x}, \boldsymbol{p_T}) + p_{data}(\boldsymbol{x}, \boldsymbol{p_T})}. \tag{44}$$

Appendix D
Proof of Lemma 4

By substituting D_{opt} into (4), following the proof in GAN [19], the objective function can be rewritten as:

$$\begin{aligned} &V(G) \\ &= \iint\limits_{\boldsymbol{x},\boldsymbol{p_T}} p_{data} \times \log D_{opt} + p_g \times \log(1 - D_{opt})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{p_T} \\ &= \iint\limits_{\boldsymbol{x},\boldsymbol{p_T}} p_{data} \times \log(\frac{p_{data}}{p_{data} + p_g}) + p_g \times \log(\frac{p_g}{p_g + p_{data}})\mathrm{d}\boldsymbol{x}\mathrm{d}\boldsymbol{p_T} \\ &= KL(p_{data} \left\| \frac{p_g + p_{data}}{2}\right) + KL(p_g \left\| \frac{p_g + p_{data}}{2}\right) - \log 4. \end{aligned} \tag{45}$$

Clearly, if and only if $p_g(\boldsymbol{x}, \boldsymbol{p_T}) = p_{data}(\boldsymbol{x}, \boldsymbol{p_T})$, the objective V reaches its minimum $-log4$.

Appendix E
Setting Details of Amplifier Models

For simplicity, we choose to use the Taylor polynomial model to imitate the differences of different emitters. The third-order of Taylor polynomial is considered, and the number of emitters is set to K. Denote the coefficients of the Taylor polynomials as $\beta^{[j]} = (\beta_1^{[j]}, \ldots, \beta_3^{[j]})$, for $j = 1, 2, \ldots, K$. Following the settings in [10] and [44], we choose the basic coefficients of the Taylor polynomials of 5 different emitters, $\beta^{[1]} = (1, 0.5, 0.3)$, $\beta^{[2]} = (1, 0.01, 0.01)$, $\beta^{[3]} = (1, 0.6, 0.08)$, $\beta^{[4]} = (1, 0.08, 0.6)$ and $\beta^{[5]} = (1, 0.4, 0.01)$.

When the number of emitters is larger than 5, we adjust the coefficients according to the theory stated in [50] to ensure the out-of-band radiation within a range of less than 15dBm with respect to the ideal situation without distortion.

Acknowledgment

The authors would like to thank anonymous reviewers for their valuable comments.

References

[1] A. C. Polak and D. L. Goeckel, "Identification of wireless devices of users who actively fake their RF fingerprints with artificial data distortion," *IEEE Trans. Wireless Commun.*, vol. 14, no. 11, pp. 5889–5899, Nov. 2015.

[2] B. Danev, H. Luecken, S. Capkun, and K. El Defrawy, "Attacks on physical-layer identification," in *Proc. 3rd ACM Conf. Wireless Netw. Secur. WiSec*, 2010, pp. 89–98.

[3] Y. A. Eldemerdash, O. A. Dobre, O. Ureten, and T. Yensen, "Identification of cellular networks for intelligent radio measurements," *IEEE Trans. Instrum. Meas.*, vol. 66, no. 8, pp. 2204–2211, Aug. 2017.

[4] Z. Zhang, K. Long, and J. Wang, "Self-organization paradigms and optimization approaches for cognitive radio technologies: A survey," *IEEE Wireless Commun.*, vol. 20, no. 2, pp. 36–42, Apr. 2013.

[5] O. Dobre, "Signal identification for emerging intelligent radios: Classical problems and new challenges," *IEEE Instrum. Meas. Mag.*, vol. 18, no. 2, pp. 11–18, Apr. 2015.

[6] B. Danev and S. Capkun, "Transient-based identification of wireless sensor nodes," in *Proc. IPSN*, Apr. 2009, pp. 25–36.

[7] Y.-J. Yuan, X. Wang, Z.-T. Huang, and Z.-C. Sha, "Detection of radio transient signal based on permutation entropy and GLRT," *Wireless Pers. Commun.*, vol. 82, no. 2, pp. 1047–1057, May 2015.

[8] G. Huang, Y. Yuan, X. Wang, and Z. Huang, "Specific emitter identification for communications transmitter using multi-measurements," *Wireless Pers. Commun.*, vol. 94, no. 3, pp. 1523–1542, Jun. 2017.

[9] P. Padilla, J. L. Padilla, and J. F. Valenzuela-Valdes, "Radio frequency identification of wireless devices based on RF fingerprinting," *Elec. Lett.*, vol. 49, pp. 1409–1410, Oct. 2013.

[10] J. Zhang, F. Wang, O. A. Dobre, and Z. Zhong, "Specific emitter identification via Hilbert–Huang transform in single-hop and relaying scenarios," *IEEE Trans. Inf. Forensics Security*, vol. 11, no. 6, pp. 1192–1205, Jun. 2016.

[11] D. R. Reising, M. A. Temple, and M. J. Mendenhall, "Improved wireless security for GMSK-based devices using RF fingerprinting," *Int. J. Electron. Secur. Digit. Forensics*, vol. 3, no. 1, p. 41, 2010.

[12] M. D. Williams, M. A. Temple, and D. R. Reising, "Augmenting bit-level network security using physical layer RF-DNA fingerprinting," in *Proc. IEEE Global Telecommun. Conf. GLOBECOM*, Dec. 2010, pp. 1–6.

[13] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in *Proc. 14th ACM Int. Conf. Mobile Comput. Netw. MobiCom*, 2008, pp. 116–127.

[14] Y. Jia, "The research on communication emitters identification technology," Ph.D. dissertation, Dept. Elect. Eng., Uni. of Elect. Sci., Tech. of China, Chengdu, China, 2017.

[15] T. O'Shea and J. Hoydis, "An introduction to deep learning for the physical layer," *IEEE Trans. Cognit. Commun. Netw.*, vol. 3, no. 4, pp. 563–575, Dec. 2017.

[16] Q. Mao, F. Hu, and Q. Hao, "Deep learning for intelligent wireless networks: A comprehensive survey," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 4, pp. 2595–2621, Jun. 2018.

[17] H. Ye, G. Y. Li, and B.-H. Juang, "Power of deep learning for channel estimation and signal detection in OFDM systems," *IEEE Wireless Commun. Lett.*, vol. 7, no. 1, pp. 114–117, Feb. 2018.

[18] S. Dorner, S. Cammerer, J. Hoydis, and S. T. Brink, "Deep learning based communication over the air," *IEEE J. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 132–143, Feb. 2018.

[19] I. Goodfellow *et al.*, "Generative adversarial Nets," in *Proc. Adv. Neural Inf. Process. Syst.*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. New York, NY, USA: Curran Associates, 2014, pp. 2672–2680. [Online]. Available: http://papers.nips.cc/paper/5423-generative-adversarial-nets.pdf

[20] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "InfoGAN: Interpretable representation learning by information maximizing generative adversarial Nets," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 2172–2180.

[21] C. Ledig *et al.*, "Photo-Realistic single image super-resolution using a generative adversarial network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 105–114.

[22] L. Yu, W. Zhang, J. Wang, and Y. Yu, "SeqGAN: Sequence generative adversarial nets with policy gradient," in *Proc. AAAI*, 2017, pp. 2852–2858.

[23] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," 2015, *arXiv:1511.06434*. [Online]. Available: http://arxiv.org/abs/1511.06434

[24] A. Makhzani, J. Shlens, N. Jaitly, I. Goodfellow, and B. Frey, "Adversarial autoencoders," 2015, *arXiv:1511.05644*. [Online]. Available: http://arxiv.org/abs/1511.05644

[25] A. Odena, C. Olah, and J. Shlens, "Conditional image synthesis with auxiliary classifier GANs," 2016, *arXiv:1610.09585*. [Online]. Available: http://arxiv.org/abs/1610.09585

[26] Z. Gan, L. Chen, and W. Wang, "Triangle generative adversarial networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5247–5256.

[27] J. Zhao, M. Mathieu, and Y. LeCun, "Energy-based generative adversarial network," 2016, *arXiv:1609.03126*. [Online]. Available: http://arxiv.org/abs/1609.03126

[28] M. Arjovsky, S. Chintala, and L. Bottou, "Wasserstein GAN," 2017, *arXiv:1701.07875*. [Online]. Available: http://arxiv.org/abs/1701.07875

[29] T. J. Oshea, T. Roy, N. West, and B. C. Hilburn, "Physical layer communications system design Over-the-Air using adversarial networks," in *Proc. 26th Eur. Signal Process. Conf. (EUSIPCO)*, Sep. 2018, pp. 529–532.

[30] D. Roy, T. Mukherjee, M. Chatterjee, and E. Pasiliao, "Detection of rogue RF transmitters using generative adversarial Nets," in *Proc. IEEE Wireless Commun. Network Conf. (WCNC)*, Apr. 2019, pp. 1–7.

[31] C. Zhu, L. Xu, X.-Y. Liu, and F. Qian, "Tensor-generative adversarial network with two-dimensional sparse coding: Application to real-time indoor localization," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[32] M. Ben-Yosef and D. Weinshall, "Gaussian mixture generative adversarial networks for diverse datasets, and the unsupervised clustering of images," 2018, *arXiv:1808.10356*. [Online]. Available: http://arxiv.org/abs/1808.10356

[33] M. K. Simon and M. Alouini, *Digital Communication Over Fading Channels: A Unified Approach to Performance Analysis*. Hoboken, NJ, USA: Wiley, 2000.

[34] S. Christian, V. Vincent, I. Sergey, S. Jon, and W. Zbigniew, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 2818–2826.

[35] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," 2015, *arXiv:1502.03167*. [Online]. Available: http://arxiv.org/abs/1502.03167

[36] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*. [Online]. Available: http://arxiv.org/abs/1412.6980

[37] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," 2016, *arXiv:1606.03498*. [Online]. Available: http://arxiv.org/abs/1606.03498

[38] O. Bachem, M. Lucic, and H. Hassani, "Fast and provably good seedings for K-Means," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, 2016, pp. 55–63.

[39] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN revisited, revisited: Why and how you should (Still) use DBSCAN," *ACM Trans. Database Syst.*, vol. 42, no. 3, pp. 1–21, Jul. 2017.

[40] M. Ester, H. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Aug. 1996, pp. 226–231.

[41] K. Merchant, S. Revay, and G. Stantchev, "Deep learning for RF device fingerprinting in cognitive communication networks," *IEEE Jour. Sel. Topics Signal Process.*, vol. 12, no. 1, pp. 160–167, Feb. 2018.

[42] L. Ding, S. Wang, F. Wang, and W. Zhang, "Specific emitter identification via convolutional neural networks," *IEEE Commun. Lett.*, vol. 22, no. 12, pp. 2591–2594, Dec. 2018.

[43] J. Zhang, F. Wang, Z. Zhong, and O. Dobre, "Novel Hilbert spectrum-based specific emitter identification for single-hop and relaying scenarios," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2015, pp. 1–6.

[44] U. Satija, N. Trivedi, G. Biswal, and B. Ramkumar, "Specific emitter identification based on variational mode decomposition and spectral features in single hop and relaying scenarios," *IEEE Trans. Inf. Forensics Security*, vol. 14, no. 3, pp. 581–591, Mar. 2019.

[45] Y. Li and Y. Liang, "Learning overparameterized neural networks via stochastic gradient descent on structured data," 2018, *arXiv:1808.01204*. [Online]. Available: http://arxiv.org/abs/1808.01204

[46] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 1026–1034.

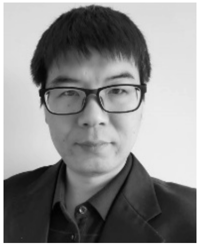
[47] S. S. Du, X. Zhai, B. Poczos, and A. Singh, "Gradient descent provably optimizes over-parameterized neural networks," 2018, *arXiv:1810.02054*. [Online]. Available: http://arxiv.org/abs/1810.02054

[48] S. S. Du, J. D. Lee, H. Li, L. Wang, and X. Zhai, "Gradient descent finds global minima of deep neural networks," 2018, *arXiv:1811.03804*. [Online]. Available: http://arxiv.org/abs/1811.03804

[49] L. Chen, L.-L. Yang, J. Yan, and R. Chen, "Joint wireless positioning and emitter identification in DVB-T single frequency networks," *IEEE Trans. Broadcast.*, vol. 63, no. 3, pp. 577–582, Sep. 2017.

[50] Q. Wu, M. Testa, and R. Larkin, "On design of linear RF power amplifier for CDMA signals," *Int. J. RF Microw. Comput.-Aided Eng.*, vol. 8, no. 4, pp. 283–292, Jul. 1998.

Jialiang Gong received the B.Eng. degree in electronic and information engineering from the University of Science and Technology of China (USTC), Hefei, China, in 2016, where he is currently pursuing the M.Eng. degree with the Department of Electronic Engineering and Information Science. His main research topic is deep learning-based wireless communication and signal processing.



Xiaodong Xu received the B.Eng. and Ph.D. degrees in electronic and information engineering from the University of Science and Technology of China (USTC) in 2000 and 2007, respectively. From 2000 to 2001, he served as a Research Assistant at the R&D Center, Konka Telecommunications Technology. Since 2007, he has been a Faculty Member with the Department of Electronic Engineering and Information Science, USTC. His research interests include the areas of wireless communications, signal processing, wireless artificial intelligence, and information-theoretic security.

Yingke Lei received the B.E. degree in communication engineering from the Electronic Engineering Institute, China, in 1998, and the Ph.D. degree in pattern recognition and intelligent systems from the University of Science and Technology of China (USTC), China, in 2010. He is currently a Professor with the National University of Defense Technology, China. His research interests are manifold learning, pattern recognition, and signal processing.