

The Importance of Expert Knowledge for Automatic Modulation Open Set Recognition

Taotao Li*, Zhenyu Wen*, Senior Member, IEEE, Yang Long, Senior Member, IEEE,
 Zhen Hong[†], Member, IEEE, Shilian Zheng, Member, IEEE, Li Yu, Member, IEEE,
 Bo Chen, Member, IEEE, Xiaoniu Yang and Ling Shao, Fellow, IEEE

Abstract—Automatic modulation classification (AMC) is an important technology for the monitoring, management, and control of communication systems. In recent years, machine learning approaches are becoming popular to improve the effectiveness of AMC for radio signals. However, the automatic modulation open-set recognition (AMOSR) scheme that aims to identify the known modulation types and recognize the unknown modulation signals is not well studied.

Therefore, in this paper, we propose a novel multi-modal marginal prototype framework for radio frequency (RF) signals (MMPRF) to improve AMOSR performance. First, MMPRF addresses the problem of simultaneous recognition of closed and open sets by partitioning the feature space in the way of *one vs. other* and marginal restrictions. Secondly, we exploit the wireless signal domain knowledge to extract a series of signal-related features to enhance the AMOSR capability. In addition, we propose a GAN-based unknown sample generation strategy to allow the model to understand the unknown world.

Finally, we conduct extensive experiments on several publicly available radio modulation data, and experimental results show that our proposed MMPRF outperforms the state-of-the-art AMOSR methods.

Index Terms—Automatic modulation classification, machine learning, open-set recognition, radio signals, marginal prototype

1 INTRODUCTION

AUTOMATIC modulation classification (AMC) is widely applied in civilian and public security fields, such as cognitive radio [1], radio reconnaissance [2], electronic countermeasures [3], threat evaluation [4], and spectrum monitoring [5], etc. AMC is considered one of the key technologies for adaptive communication systems and has important application prospects [1].

Most AMC methods focus on the closed-set setting which assumes that all testing modulation types are known in the training phase [6], [11], [12], [13], [14], [15], [16]. However, rejecting unknown radio frequency (RF) signals is essential for many real-world applications. For instance, RF fingerprint technology [17], [18] has been used for device authentication in IoT networks. To this end, the RF authentication system needs to have the OSR capability to reject unknown devices that are not in the database. Another example is that AMOSR is important for low-altitude small flying object detection tasks (e.g., drone) in which the computer-vision based solutions are easily limited by the performance of the equipment and the influence of the

environment [19]. Spectrum monitoring has the potential advantage of detecting these objects while avoiding the influence of the environment [20].

The key of AMOSR is to efficiently reject the unknown modulation type while identifying the known modulation type (detailed in §2.2). Classical deep learning usually appends a fully connected layer at the end of the network to help with the classification task, and softmax is used to train the network. As shown in Fig. 1b, softmax assigns probability distributions, i.e., feature subspaces, to each closed-set class in the feature embedding space. It means that each unknown class is also assigned to a uniform existing feature subspace, which makes the distribution of known classes and unknown classes in the feature space highly overlapping [10]. As a result, AMOSR becomes particularly difficult for traditional deep learning classification models by using softmax function [21]. Fig. 1c show that AMOSR needs to make an accurate description of the labeled known classes, at the same time, to mitigate the overlapping area between known and unknown classes in the feature space [22].

In order to reduce the risk of open-set space by using the softmax function, some prototype-based algorithms are proposed, such as GCPL [9], RPL [7], ARPL [10]. These methods generally use prototypes to represent each known class in the feature embedding space and conduct open-set recognition (OSR) or closed-set recognition (CSR) tasks by computing the distance between the sample and the prototype. However, as shown in Fig. 2b-2d and 2g-2i, it is obvious that the high-density areas of unknown samples (red) are highly overlapping with the known classes (purple) in the RF feature embedding space. It means that the current advanced OSR algorithms provide poor results for

• T. Li, Z. Wen, Z. Hong, L. Yu, B. Chen, X. Yang are with the Institute of Cyberspace Security, and the Institute of Information Engineering, Zhejiang University of Technology, Hangzhou, Zhejiang 310023 China. Email: {2111903074, zhenyuwen, zhong1983, lyu }@zjut.edu.cn.

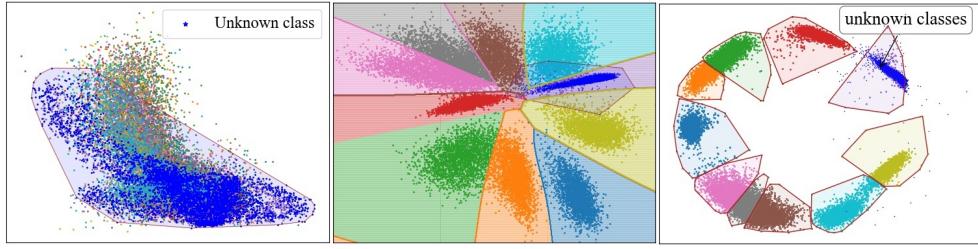
• Shilian Zheng is with No. 011 Research Center, Science, and Technology on Communication Information Security Control Laboratory, Jiaxing 314033, China. E-mail: lianshizheng@126.com.

• Y. Long is with the Department of Computer Science, Durham University, Durham, UK. E-mail: yang.long@ieee.org.

• L. Shao is with the Inception Institute of Artificial Intelligence, Abu Dhabi, UAE. E-mail: ling.shao@ieee.org.

• To whom correspondence should be addressed. E-mail: zhong1983@zjut.edu.cn

* These authors contributed equally to this work.



(a) Original I-Q signals (b) Traditional AMC problem (c) Current AMOSR problem

Fig. 1. The difference between AMOSR and AMC in dataset Sig-2019 [6], where blue denotes unknown class and other colors denote known classes: (a) original I-Q features without training; (b) the traditional AMC approach will apply the decision boundary surface of each class to perform the classification task, which misclassifies unknown samples as known classes; (c) the current AMOSR method (RPL [7]) will restrict the decision boundary surface for each class and use the open space to identify unknown samples.

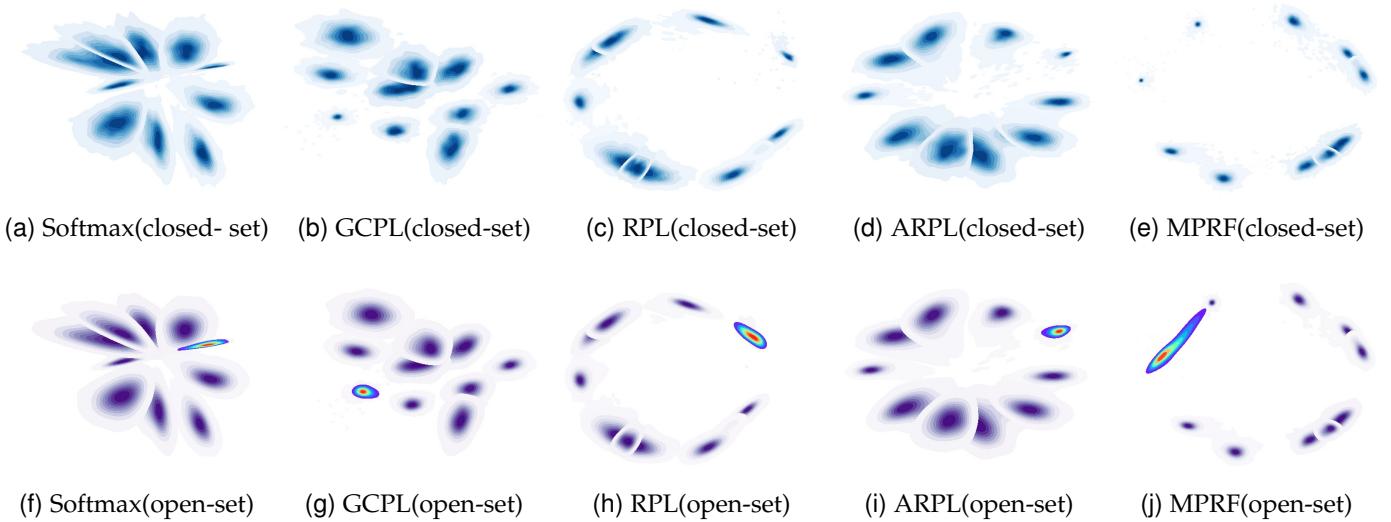


Fig. 2. The sample density heat map illustrates the RFF feature embedding space for known and unknown modulation types by 1D-CNN, as reported [7], [8], [9], [10]. In the Sig-2019 radio recognition dataset [6], we designate 10 classes of samples as known and 2 classes as unknown. The first row showcases the feature distributions within the closed-set sample space (represented by the purple heat map), while the second row illustrates the feature distributions of unknown samples (depicted by the rainbow heat map). In this heat map, a deeper shade of purple or red signifies a higher sample density. With a parameter set at 95% scope from the purple center (meaning the area outside the center is an open area eligible for rejection), the open-set rejection rate of existing methods hovers around 5%-12% (almost identical across all methods). Our technique, on the other hand, manages to pull a portion of the samples away from the target class centroid (represented by long stripes), achieving an open-set rejection rate of approximately 62.5%.

the AMOSR scenario.

In this paper, we propose a general algorithmic framework that combines the basic radio signal representations including amplitude, frequency, and phase of the RF signal with a marginal prototype-based network for AMOSR. We first design a marginal prototype algorithm for AMOSR of RF (MPRF), which can reduce the open-set space risk and closed-set empirical classification risk by compressing the known classes that are at the restricted margins according to their prototype centers in the feature embedding distribution space. Furthermore, we leverage the expert knowledge and extract general modal characteristics of the RF signal (e.g., frequency, amplitude and phase). Finally, to explore more unknown samples and allow the neural network to understand the concept of unknown samples we propose a generation scheme, MMPRF+. MMPRF+ effectively generates unknown samples which close to the borders of the known classes to further improve the robustness of our method. Fig. 2e and 2j demonstrate that our MPRF has a tighter closed-set space and significantly reduces the overlap area between the known classes and unknown classes. From Fig. 2e, the spatial scope of classes similar to the

unknown class is collapsed to a single point, which means that it has a smaller overlap area with the open set. From Fig. 2f, our method can pull some of the samples out from the center point (red) to avoid overlap (long strip distribution). With the combined effect of a smaller closed-set scope and pulled-out open-set samples, our MPRF can improve the open-set recognition performance. Our contributions are summarized as follows:

- To address the poor performance of the AMOSR task, we analyze and extract key RF features (i.e., amplitude, frequency, phase) that can significantly improve performance.
- Compared to previous OSR algorithms, a new optimization strategy is proposed with a marginal prototype and has a lower open space risk;
- We develop a multi-modal framework for the extracted generic RF features, which allows the retention of separate features for different modes and thus mitigates the problem of feature disappearance in OSR tasks;
- To enable the estimation of unknown open spaces, we propose an online adversarial generation strategy for unknown radio signals, which motivates the neural network

to see more unknown samples in the AMOSR task.

2 BACKGROUND AND CHALLENGES

2.1 Background of wireless communication

In wireless communication systems, the message will be modulated into a radio signal for wireless transmission. Modulation aims to add information to a set of signals by varying one or more properties of periodic electromagnetic waves (carriers) [23]. A transmitted time modulation signal $r(t)$ can be illustrated as

$$r(t) = \mathcal{S}(t) * h(t) \exp[j2\pi\Delta f t + \psi_0] + wgn(t), \quad (1)$$

where $*$ represents the convolution operation, $\mathcal{S}(t)$ denotes the modulated signal, $h(t)$ denotes the impulse response of the wireless channel, Δf denotes the frequency offset of carrier, ψ_0 denotes the initial phase which reflects the time delay, $wgn(t)$ denotes the white Gaussian noise in environment. For further distance transmission, the modulated signal is usually carried on a high-frequency carrier wave of frequency Δf . In addition, the different modulation of the signal is achieved by using variations in amplitude, frequency, and phase.

2.2 Problem Definition for AMOSR

An AMOSR scenario consists of three main phases, i.e., RF features (RFF) extractor, AMOSR, and AMC. In the first phase, a set of RF transmitters ($Txs = Tx_1, Tx_2, \dots, Tx_K$) is capable of transmitting RF signals to the RF receiver (Rx). A given RF signal $s_k(t)$ which is generated by the k -th Tx is transmitted to the Rx. In order to allow the computer to process the $(r_k(t))$, we need to transfer the continuous time series signal to a set of discrete time series signals via sampling $x_k(n) = sample(r_k(t))$, where $\forall x_k(t) \in \mathbb{R}^N$ and N denotes the dimension. Finally, the RFF extractor extracts the features RFF(i) and feeds them to a deep neural network (DNN) model to identify the type of modulation.

AMOSR V.S. AMC. As we discussed above, the DNN model is trained to identify the Txs or a set of categories of Tx that remain the same during both training and testing phases. In real world scenarios, there is large number of unknown Txs that may come from the classes that have not yet been encountered in the training classes. Thus, the AMOSR not only identify the differences between training classes (i.e., AMC), but also inference whether an unknown Txs comes from the training classes. As shown in Fig. 2, if a model performs well in AMC, it does not necessarily perform well in AMOSR.

The problem definition of AMOSR. Given a training RF dataset $\mathcal{D}_{tr} = \{(x_i(t), y_i) | i \in M, x_i(t) \in \mathbb{R}^N\}$ with M samples, where $y_i \in \{y_1, y_2, \dots, y_K\}$ denotes the label of sample x_i . We then define the potential unknown data as $\mathcal{D}_u = \{(x_u(t), y_u) | y_u = \{K+1\}\}$, which shows that potentially unknown data may come from quite different categories, and their specific categories are not important to AMOSR and we uniformly define its label as $K+1$. During the testing phase of AMOSR, there exists a large amount of test data, $\mathcal{D}_{te} = \{(x(t)', y') | y' \in \{1, 2, \dots, K+1\}\}$, and their labels belong to $\{1, 2, \dots, K+1\}$, i.e., contains both known and unknown samples. In the AMOSR scenario, we aim to recognize unknown samples \mathcal{D}_u in the test set as category $\{K+1\}$ as well as correctly classify the known category samples $\mathcal{D}_{te} - \mathcal{D}_u$.

The key of AMOSR is the necessity to minimize both the open space risk R_o in M -dimensional full space \mathbb{R}^M as well as the closed space empirical risk R_e on AMC as illustrated in Eq. 2.

$$\arg \min_{f_\theta} \{\xi R_e(\mathcal{D}_{te} - \mathcal{D}_u; f_\theta) + (1 - \xi) R_o(\mathcal{D}_u; f_\theta)\} \quad (2)$$

where $f_\theta : \mathbb{R}^M \rightarrow K$ denotes the parameters of the multi-class recognition function, and ξ denotes the weight parameter. The function f_θ maps the closed-set data $\mathcal{D}_{te} - \mathcal{D}_u$ to the label $\{1, 2, \dots, K\}$. In particular, AMOSR is to measure the uncertainty of classifying an unknown sample as known or unknown, which can be formulated as

$$R_o(\mathcal{D}_u; f_\theta) = \frac{\int_{\mathcal{D}_u} f_\theta(x) dx}{\int_{\mathcal{D}_{te}} f_\theta(x) dx}. \quad (3)$$

We can observe that more unknown samples classified as known will increase the open space risk R_o . Then in a practical multi-classification task, AMOSR is viewed as multiple binary classification tasks, i.e., it needs to compare with K known classes separately. Therefore, the Eq. 2 can be further rewritten as

$$\arg \min_{f_\theta} \{\xi R_e(\mathcal{D}_{te} - \mathcal{D}_u; f_\theta) + (1 - \xi) \sum_{k=1}^K R_o(\mathcal{D}_u; f_\theta)\} \quad (4)$$

As a result, the goal of AMOSR aims to obtain a feature embedding function f_θ from the training set \mathcal{D}_{tr} , which can minimize Eq. 4.

2.3 Research Challenges: RFF vs. image

Although existing OSR algorithms [7], [9], [10], [24], [25] have made tremendous advances in computer vision, there are significant challenges in implementing them for AMOSR. Fig. 2 shows that the SOTA (state-of-the-arts) OSR algorithms do not work significantly in AMOSR scenarios. Additionally, Fig. 4 shows the proposed framework outperforms other SOTA solutions, which proves the importance of the basic components of the signal. We take 2FSK and 8FSK of the dataset Sig-2019 [6] as unknown classes and the rest 10 classes as known. It can be obviously noticed that the false positive rate (FPR) using the original I-Q data in the AMOSR task is 88.70% and the neural network has difficulty in rejecting the unknown classes effectively. In contrast, we extracted the key information of frequency, phase, and amplitude, which significantly reduce the FPR. As a result, we summarize the main reasons as follows.

High similarity of modulation methods. Fig. 3 gives the constellation diagram of 8 different signal modulation methods. After that, we can obviously observe that there is a significant difference between the major classes of modulation on the constellation diagram. However, the differences in the constellation diagrams between subclasses are subtle and are difficult to distinguish by constellation diagrams such as 2FSK, 4FSK, and 8FSK. Minor modifications of homogeneous modulation parameters lead to a high degree of similarity, which brings significant challenges for the AMOSR task.

Lack of expert knowledge. Inspired by the field of speech recognition, mel frequency cepstral coefficient (MFCC) [26] is widely used to enhance the discriminability of the speaker's voice patterns to achieve better performance.

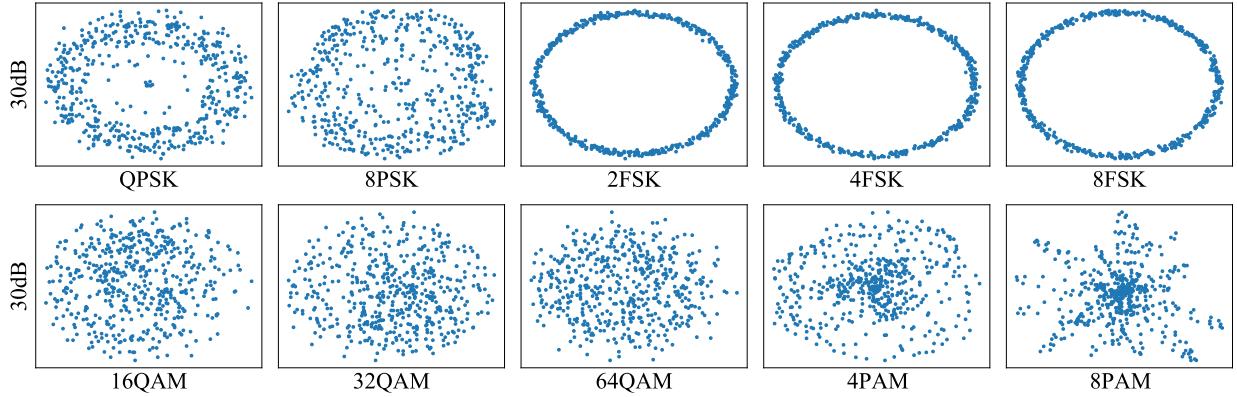


Fig. 3. Scatter plot of 8 modulated I-Q signals in the Sig-2019 [6] at a signal-to-noise ratio (SNR) of 30 dB.

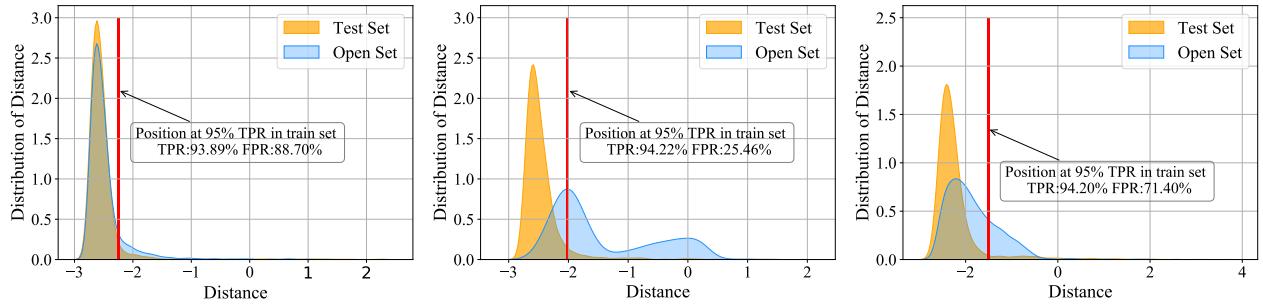


Fig. 4. AMOSR performance with different RF features in dataset Sig-2019 [6], where 2FSK and 8FSK as the unknown class: (a) original I-Q amplitude data; (b) frequency and amplitude features; (c) phase and amplitude features. It is obvious that the extracted frequency, phase and amplitude features have a lower false positive rate (FPR) compared to the original I-Q data when a threshold of 95% true positive rate (TPR) is set for the train set.

However, in AMOSR tasks, the expert knowledge (e.g., traditional radio signal processing method) is not well attached to the deep neural network. One of the key advantages of expert knowledge is to find a better representation of the features thereby allowing the neural networks better to distinguish the known classes and the unknown classes. **Catastrophic feature disappearance.** In supervised learning, the classifier aims to extract specific features that can accurately classify samples in the closed set space, but these features are not always valid for AMOSR. We believe the biases of supervised learning lead to the critical features for unknown classes disappear, i.e., catastrophic feature disappearance.

Key Design Ideas. Based on the above findings, our key ideas consist of three main points: (1) extracting more key RF features that can work for AMOSR; (2) avoiding catastrophic disappearance of the extracted key features during training; (3) designing more effective algorithms.

3 MULTI-MODAL BASED MARGINAL PROTOTYPE FRAMEWORK

3.1 Overview

To combine the advantages of both expert knowledge and deep learning, we propose a multi-modal based marginal prototype framework for AMOSR as shown in Fig. 5. First, the **feature extractor** extracts the expert knowledge of the radio signals through signal processing technologies, generating different modalities of the signal (§3.2). We then employ a **modal fusion** operation to combine different modalities and apply the proposed MPRF algorithm for

training and test of the AMOSR task (§3.3). After that, to prevent the catastrophic disappearance of features, we use the **modal factoring** operation to extract the internal differences between various modalities and employ a space learning method for simultaneous training with the proposed MPRF algorithm (§3.4.1). In particular, we introduce **space learning** operations to achieve a joint representation of fusion and factoring features. Finally, a new loss function, **similarity calculation** is developed for training MPRF for performing AMOSR (see §3.4.2). Table 1 shows the notations used in the following context.

3.2 Feature Extraction

In this subsection, we first show the significant improvement in performing AMOSR by adding expert knowledge. Next, we illustrate our multi-modal extraction method that transforms the original radio signal into different representations, which improves the ability of feature extraction of a deep neural network.

The importance of expert knowledge. We perform an AMOSR task on dataset Sig-2019 [6] and assumed that 10 classes are known and 2 classes are unknown samples. Fig. 4 shows that the increase of modal characteristics of the radio signal increases the distribution distance between the known and unknown classes. The following provides a deeper discussion of this observation.

As described in section 2.1, modulation signal $S(t)$ mainly uses the changes in amplitude, frequency, and phase to carry the information of the binary bit stream $s_b(t)$, i.e., Amplitude-shift keying (ASK), Frequency-shift keying

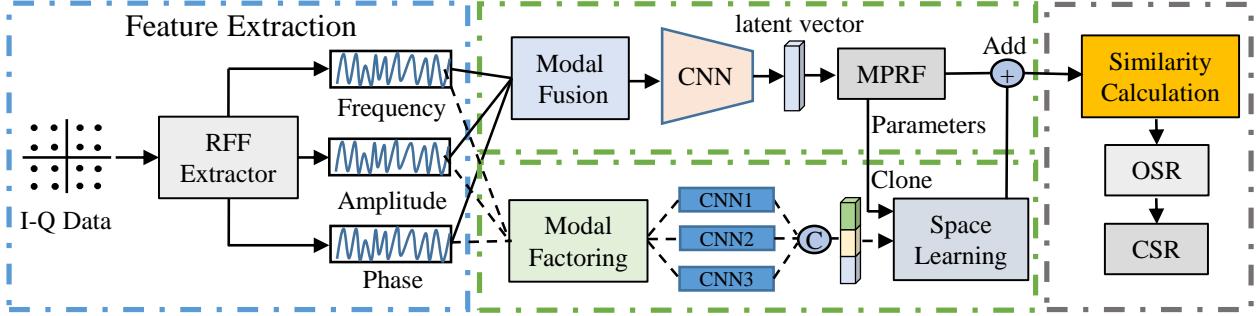


Fig. 5. The workflow of MMPRF framework. Firstly, the feature extraction module extracts key representations of the signals. Secondly, the modal fusion and factoring module uses CNN to learn the signal features by the MPRF algorithm respectively. Finally, the AMOSR and AMC tasks are performed by similarity calculation.

TABLE 1
Important Symbol Notations

Notation	Description
$\mathcal{I}(t), \mathcal{Q}(t)$	I-Q data
$x_k(t)$	The k -th RF discrete signal
Y	RF signal label
K	Thr number of categories
\mathcal{D}	Dataset
f_θ	Neural network parameters
$A(t)$	Amplitude of signal
$F(w)$	Frequency of signal
$\psi(t)$	Phase of signal
\mathcal{P}_n	Non-adjacent point
\mathcal{P}	Prototype point
\mathcal{R}	Limiting radius
z_{fu}	Fusion feature
z_{fa}	Factoring feature
ζ	Threshold
D_θ	Discriminator parameters
G_θ	Generator parameters

(FSK), and Phase-shift keying (PSK). In general, the $\mathcal{S}(t)$ can be represented in three ways, i.e.,

$$\begin{aligned} \text{ASK : } \mathcal{S}(t) &= A_c s_b(t) \exp[j2\pi f_c t], \\ \text{FSK : } \mathcal{S}(t) &= A_c \exp[j2\pi f_c t + j2\pi \Delta_f \int_0^t s_b(\tau) d\tau], \\ \text{PSK : } \mathcal{S}(t) &= A_c \exp[j2\pi f_c t + j2\pi (\Delta_p s_b(t) + \psi_0)], \end{aligned} \quad (5)$$

where A_c denotes the carriers amplitude, Δ_f denotes the frequency modulation factor, Δ_p is the phase modulation factor and τ denotes the time delay. From Eq. 5, we can obvious that the amplitude $A(t)$ in ASK, frequency $F(t)$ in FSK and phase $\psi(t)$ in PSK of the carriers are dependent with $s_b(t)$, which can be represented by

$$\begin{aligned} A(t) &= A_c s_b(t) \propto s_b(t), \\ F(t) &= f_c t + \Delta_f \int_0^t s_b(\tau) d\tau \propto \Delta_f \int_0^t s_b(\tau) d\tau, \\ \psi(t) &= \Delta_p s_b(t) + \psi_0 \propto \Delta_p s_b(t). \end{aligned} \quad (6)$$

Extracting features via expert knowledge. Eq. 5, 6 illustrate that the variation of $A(t)$, $F(t)$, $\psi(t)$ carries the main information of the modulated signal, so we can focus on extracting these three pieces of information to help the task of AMOSR. In addition, to facilitate signal information extraction and signal recovery, in-phase signals and the quadrature-phase signal are used to jointly characterize the relevant modulation information, i.e. I-Q data [27]. So we define the received discrete complex signal as $x_{IQ}(n) = \{x_I(n), x_Q(n)\}$ which is sampled from $r(t)$, denoted as shown in Eq. 7

$$\begin{aligned} \{x_I(n), x_Q(n)\} &= \text{sample}\{r_I(t), r_Q(t)\}, \\ \{r_I(t), r_Q(t)\} &= \{\text{Re}\{r(t)\}, \text{Im}\{r(t)\}\}, \end{aligned} \quad (7)$$

where $x_I(n)$ denotes the in-phase signal, $x_Q(n)$ represents the quadrature-phase signal, Re is the real part and Im indicates the imaginary part. We then can extract the discrete amplitude $A_r(n)$, discrete frequency spectrum $F_r(w)$ and discrete phase $\psi_r(n)$ modal of the signal at the Rx side by following operations.

$$\begin{aligned} A_r(n) &= \sqrt{x_I(n)^2 + x_Q(n)^2}, \\ F_r(w) &= \mathcal{F}[x_{IQ}(t)] = \sum_{t=0}^{M-1} x_{IQ}(t) \exp(-jw t), \\ \psi_r(n) &= \arctan(x_Q(n)/(x_I(n) + \varepsilon)), \end{aligned} \quad (8)$$

where $\mathcal{F}[\bullet]$ is the fast Fourier transform (FFT), w indicates the discrete frequency bit and ε denotes the microconstants. Fig. 5 shows that *modal fusion operation* (i.e., *concat*) combines the various representation of signals and then feed them into a CNN f_θ for feature extraction. The obtained fusion representation z_{fu} is formalized as

$$z_{fu} = \text{conv}(\text{concat}\{x_{IQ}(n), A_r(n), F_r(w), \psi_r(n)\}; f_\theta) \quad (9)$$

where $z_{fu} \in \mathbb{R}^d$, d denotes the representation space dimension and conv denotes the convolution layer. Finally, we subject the extracted z_{fu} to the AMOSR task by our proposed MPRF algorithm.

3.3 Marginal Prototype Framework for AMOSR

As shown in the Fig. 6, the proposed MPRF algorithm consists of two main steps, including *non-adjacent points for classification* (Fig. 6(a)-6(b)) and *adversarial marginal prototype constraint* (Fig. 6(c)-6(d)). During the *non-adjacent points for classification* phase, our design idea is to make the DNN model capable of identifying unknown samples. The non-adjacent points define the sample space that does not belong to the target class, i.e., "what is not like it?". To this end, as shown in Fig. 6(a)-6(b), the target class samples limited to a space of radius \mathcal{R} will move away from the non-adjacent point and the non-target class samples will move closer to the non-adjacent point. Finally, the model learns which samples do not belong to the target class and thus have the ability to reject unknown samples, i.e. the way of *one vs. other* are formed. However, the target class sample distribution is still sparse as shown in Fig. 6(b), bringing a significant AMOSR risk. Therefore, in the *adversarial marginal prototype constraint* phase, we propose a marginal prototype constraint that intends to further compress the spatial distribution of target class samples to mitigate the AMOSR threat. Our key idea is to first calculate the prototype points of the target class and cluster the target class samples towards the prototype points as shown in Fig. 6(c).

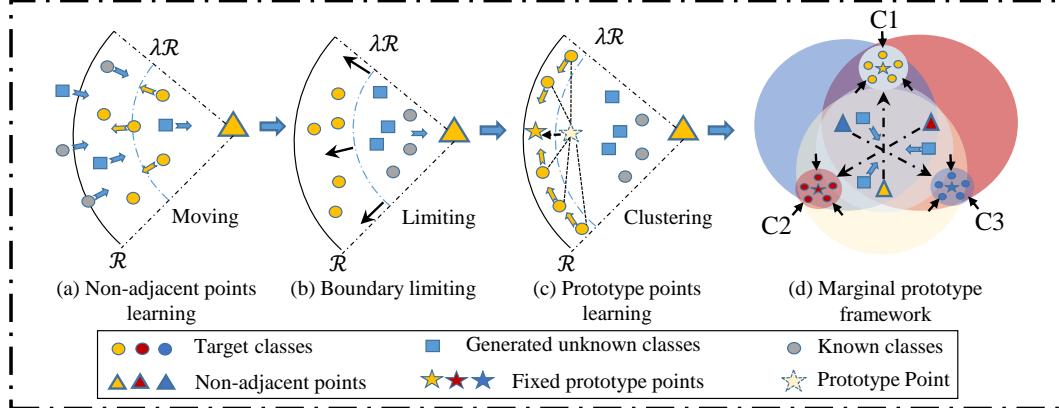


Fig. 6. The workflow of MPRF algorithm. First, MPRF discriminates the non-target class samples by being defined as non-adjacent points. After that, the target class samples will be far away from the non-adjacent points and at the edge of the feature embedding space. Finally, the target class samples will calculate their prototypes and further tighten them.

3.3.1 Non-adjacent Points for Classification

In this section, we introduce *non-adjacent points* that allow CNN able to perform AMC tasks while learning the ability to distinguish the samples from unknown classes. We first define the non-adjacent points of target category y_k as \mathcal{P}_n^k , which is regarded as the latent representation of the data $\mathcal{D}_{te}^{y_k} \cup \mathcal{D}_u$. Hence, the samples of $\mathcal{D}_{te}^{y_k}$ and \mathcal{D}_u should be closer to the non-adjacent points \mathcal{P}_n^k than the target class sample $\mathcal{D}_{te}^{y_k}$, formulated as

$$\max(\ell(\mathcal{D}_{te}^{y_k} \cup \mathcal{D}_u, \mathcal{P}_n^k; f_\theta)) \leq l, \forall l \in \ell(\mathcal{D}_{te}^{y_k}, \mathcal{P}_n^k; f_\theta), \quad (10)$$

where l denotes the distance and ℓ denotes the distance metric function. Based on Eq. 10, the sample space of the target class y_k creates an opposing form of *one vs. other*.

Distance metric. After that, in a d -dimensional feature space, we define the distance metric function ℓ , which is a foundation for performing AMC and AMOSR tasks. In this paper, we define the distance metric function $\ell(f_\theta(x(n)), \mathcal{P}_n^k)$ as illustrated in Eq 11 that is calculated by the Euclidean distance $\ell_e(f_\theta(x(n)), \mathcal{P}_n^k)$ subtracting the dot product distance $\ell_d(f_\theta(x(n)), \mathcal{P}_n^k)$. The set of sample features $x(n)$ and the specific distance metric can be represented as follows

$$\begin{aligned} \ell(f_\theta(x(n)), \mathcal{P}_n^k) &= \ell_e(f_\theta(x(n)), \mathcal{P}_n^k) - \ell_d(f_\theta(x(n)), \mathcal{P}_n^k). \\ x(n) &= \{x_{IQ}(n), A_r(n), F_r(w), \psi_r(n)\}, \\ \ell_e(f_\theta(x(n)), \mathcal{P}_n^k) &= \frac{1}{d} \|f_\theta(x(n)) - \mathcal{P}_n^k\|_2^2, \\ \ell_d(f_\theta(x(n)), \mathcal{P}_n^k) &= f_\theta(x(n)) \cdot \mathcal{P}_n^k. \end{aligned} \quad (11)$$

Eq. 11 indicates the closer distance the higher similarity between samples and non-adjacent points. Note that the combination of Euclidean distance and dot product distance can better reflect the similarity of two samples in terms of distance and angle [21].

Classification. Then, according to the defined distance metric, we need to determine the representations $f_\theta(x(n))$ of the target sample $x(n)_k$ belonging to which class. Therefore, in the feature space of multi-class classification, we drive the target class samples $x(n)_k$ to have a farther distance from non-adjacent points \mathcal{P}_n^k relative to other non-adjacent points $\mathcal{P}_n^{\neq k}$ as shown in Fig. 6 (a). We define the generated unknown classes and the other known classes in the sample

as non-target classes and drive them close to non-adjacent points \mathcal{P}_n^k . Our final class probabilities are calculated by

$$p(y = k|x(n), f_\theta, \mathcal{P}_n) = \frac{\exp(\ell(f_\theta(x(n)), \mathcal{P}_n^k))}{\sum_{i=1}^K \exp(\ell(f_\theta(x(n)), \mathcal{P}_n^i))}. \quad (12)$$

So we minimize the non-adjacent point-based classification loss \mathcal{L}_c and optimize the neural network parameters f_θ .

$$\mathcal{L}_c(x(n); f_\theta, \mathcal{P}_n) = -\log p(y = k|x(n), f_\theta, \mathcal{P}_n). \quad (13)$$

We can observe that when we try to minimize Eq. 13, the Eq. 12 is maximized, i.e., we are able to maximize the distance of the sample $\mathcal{D}_{te}^{y_k}$ to the \mathcal{P}_n^k , which is formulated as

$$\arg \max_{f_\theta} \ell(\mathcal{D}_{te}^{y_k}, \mathcal{P}_n^k; f_\theta). \quad (14)$$

Additionally, the above approach enables CNN to have the ability of discriminate the unknown, while bringing the following drawbacks. Maximizing Eq. 12 leads to a continuous enlargement of the open space, and the closed set space samples are not further reduced. As shown in Fig. 6 (a), the target class continues to move away from its non-adjacent point with an increase in radius R , which makes the spatial distribution of the target class samples larger and makes the distribution sparse. The following elaborates on how to further de-constraint the feature space.

3.3.2 Adversarial Marginal Prototype Constraint

In this subsection, we aim to restrict the distance of all sample $\mathcal{D}_{tr} \cup \mathcal{D}_u$ within the radius R^k , which is formalized as:

$$\max \ell(\mathcal{D}_{tr} \cup \mathcal{D}_u, \mathcal{P}_n^k; f_\theta) < R^k. \quad (15)$$

The Eq. 15 can be further derived to the following loss function as:

$$\mathcal{L}_b(x(n); f_\theta, \mathcal{P}_n^k, R^k) = \max\{\ell(f_\theta(x(n)), \mathcal{P}_n^k) - R^k, 0\}. \quad (16)$$

Minimizing Eq. 16 drives the target class samples $x(n)_k$ to be at the boundary of the feature space with radius R^k as shown in Fig. 6 (b). However, the distribution of these target class samples is still sparse, and we believe that further reduction of the closed set space of $x(n)_k$ can reduce the risk of open space. To achieve this, we use the target class's prototype points to tighten the known class distribution space.

We use Eq. 17 to compute the prototype points \mathcal{P}^k of n samples of $x(n)_k$ which allocates on the boundaries of space.

$$\mathcal{P}^k = \frac{1}{n} \sum_{i=0}^n f_\theta(x(n)_k^i). \quad (17)$$

The prototype points \mathcal{P}^k is the mean center of a class of embedded features, and we aim to move the similar samples closer to the center for the purpose of shrinking the closed set space. Thus, we define the optimization goal as:

$$\arg \min_{f_\theta} \ell(\mathcal{D}_{tr}^{y_k}, \mathcal{P}^k; f_\theta). \quad (18)$$

There are some limitations in calculating the prototype \mathcal{P}^k . As shown in Fig. 6 (c), we must consider the case that the target class sample is at the boundary of a circle with distance \mathcal{R} from the \mathcal{P}_n^k . The prototype point \mathcal{P}^k is computed in such a case may be close to the \mathcal{P}_n^k , i.e., the center of the circle. However, we expect the samples to be far from the \mathcal{P}_n^k and close to the \mathcal{P}^k . Once the prototype points are close to the \mathcal{P}_n^k , this deviates from our expectation and makes the CNN optimization harder. We must consider the relative position of prototype points in relation to non-adjacent points. Therefore, we make the prototype fixed by imposing restrictions on the \mathcal{P}^k to keep the prototype at the boundaries, i.e. the furthest point relative to a non-adjacent point. This operation provides a more precise direction of aggregation of the target classes as shown in Fig. 6 (c). This operation is formulated as

$$\begin{aligned} & \min \ell(f_\theta(x(n)^k), \mathcal{P}^k) \\ & \text{s.t. } \ell(\mathcal{P}^k, \mathcal{P}_n^k) - \mathcal{R}^k = 0. \end{aligned} \quad (19)$$

Then, we define the marginal prototype constraint loss \mathcal{L}_p in Eq 20.

$$\mathcal{L}_p(x(n)^k; f_\theta, \mathcal{P}^k, \mathcal{R}^k) = \ell(f_\theta(x(n)^k), \mathcal{P}^k) + \beta(\ell(\mathcal{P}^k, \mathcal{P}_n^k) - \mathcal{R}^k), \quad (20)$$

where $\beta \in (0, 1)$ is a hyperparameter.

Finally, in the adversarial marginal prototype learning, we combine the Eq. 13, 16 and 20 to obtain the overall loss function as

$$\begin{aligned} \arg \min_{f_\theta^*, \mathcal{P}_n^*, \mathcal{P}^*, \mathcal{R}^*} \mathcal{L} &= \mathcal{L}_c + \mathcal{L}_p + \alpha \mathcal{L}_b \\ &= -\log p(y = k|x(n), f_\theta, \mathcal{P}) \\ &\quad + \ell(f_\theta(x(n)), \mathcal{P}) + \beta(\ell(\mathcal{P}, \mathcal{P}_n) - \mathcal{R}) \\ &\quad + \alpha \max\{\ell(f_\theta(x(n)), \mathcal{P}_n) - \mathcal{R}, 0\} \end{aligned} \quad (21)$$

where \mathcal{R}^* , \mathcal{P}_n^* , \mathcal{P}^* are the learnable parameters, $\alpha \in (0, 1)$ is a hyperparameter.

3.3.3 MPRF Working Procedure

This subsection illustrates the working procedure of MPRF, which is summarized in Alg. 1. MPRF is the core OSR algorithm in the overall framework of Fig. 5. Line 1 shows that MPRF randomly generates K non-adjacent points in the feature embedding space and the CNN network parameters are initialized. Then, the MPRF calculates the loss of non-adjacent points and prototype points by using Eq. 21 to drive the movement of the samples in space (Line 4-8). Finally, the parameters are updated during the training procedure (Line 9-14).

Algorithm 1: MPRF Algorithm for AMOSR Task

Input: Training data $\mathcal{D}_{tr} = \{x_i(n), y_i\}_{i=1}^M$, initialized CNN parameters f_θ , non-adjacent points \mathcal{P}_n , prototype \mathcal{P} , radius \mathcal{R} , and hyperparameter α, β in loss Eq. 21, learning rate lr

Output: The optimal parameters f_θ^* , \mathcal{P}_n^* , \mathcal{P}^* and \mathcal{R}^*

```

1 Initializing the CNN parameters  $f_\theta$ ,  $\mathcal{P}_n$  and  $\mathcal{R}$ ;
2 for epoch = 1 to Epoch do
3   for batch in  $\mathcal{D}_{tr}$  do
4     Calculate classification loss  $\mathcal{L}_c$  by Eq. 13 ;
5     Calculate boundary loss  $\mathcal{L}_b$  by Eq. 16 ;
6     Update local prototype  $\mathcal{P}^*$  by Eq. 17 ;
7     Calculate prototype loss  $\mathcal{L}_p$  by Eq. 20 ;
8     Calculate overall loss  $\mathcal{L} = \mathcal{L}_c + \mathcal{L}_p + \beta \mathcal{L}_b$ ;
9     Compute the backpropagation error by
10     $\frac{\partial \mathcal{L}}{\partial x(n)} = \frac{\partial \mathcal{L}_c}{\partial x(n)} + \frac{\partial \mathcal{L}_p}{\partial x(n)} + \alpha \frac{\partial \mathcal{L}_b}{\partial x(n)}$ ;
11    Update the parameters  $\mathcal{P}_n^*$  by
12     $\mathcal{P}_n^* = \mathcal{P}_n - lr \cdot (\frac{\partial \mathcal{L}_c}{\partial x(n)} + \alpha \frac{\partial \mathcal{L}_b}{\partial x(n)})$ ;
13    Update the parameters  $\mathcal{R}^*$  by
14     $\mathcal{R}^* = \mathcal{R} - lr \cdot \alpha \cdot \frac{\partial \mathcal{L}_b}{\partial x(n)}$ ;
15    Update the parameters  $f_\theta^*$  by
16     $f_\theta^* = f_\theta - lr \cdot (\frac{\partial \mathcal{L}_c}{\partial x(n)} + \alpha \frac{\partial \mathcal{L}_p}{\partial x(n)})$ ;
17  end
18  Update  $f_\theta = f_\theta^*$ ,  $\mathcal{R} = \mathcal{R}^*$ ,  $\mathcal{P}_n = \mathcal{P}_n^*$ ,  $\mathcal{P} = \mathcal{P}^*$ ;
19 end
20 return the parameters  $f_\theta$ ,  $\mathcal{P}_n$ ,  $\mathcal{P}$  and  $\mathcal{R}$ 

```

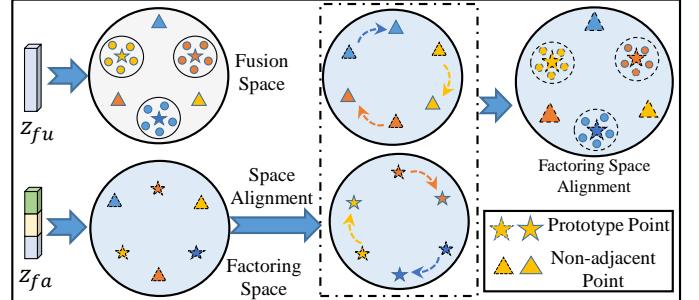


Fig. 7. Space learning operation: In MPRF, we want to merge different signal modal features to serve OSR. Therefore we need to align the features z_{fu} and z_{fa} in space, i.e., mainly align the respective non-adjacent and prototype points.

3.4 Multi-Modal Marginal Prototype Framework for AMOSR

The above MPRF based on modal fusion can extract the features z_{fu} of the combined representation of modalities. However, MPRF can not overcome the challenge of feature disappearance. To this end, we first extract each signal modality separately to allow the network to understand each modal while identifying their differences. In order to combine fusion features and factoring features for the AMOSR task, we use space learning operations to implement the alignment of feature space. The space alignment operation allows features from different modals to be jointly represented at the same scale. Finally, we define a similarity calculation to provide key criteria for AMC and AMOSR tasks.

3.4.1 Multi-Modal Learning

Multi-CNN modal factoring learning. Fig. 5 shows that we use a modal factoring operation to make different features $x_{fa}(n) = \{A_r(n), F_r(w), \psi_r(n)\}$ independent in the CNN channel during the training process so that they do not interact with each other. We believe that learning these features independently by multiple CNN models and combining them at the end of the network will effectively alleviate the challenge of catastrophic feature disappearance. This is because in this process, different signal modal features are individually preserved and can be used as key features for AMOSR tasks.

The extracted factoring features z_{fa} are then concatenated for subsequent computation, which can be represented by

$$\begin{aligned} z_A, z_F, z_\psi &= (\text{conv}_A(A_r), \text{conv}_F(F_r), \text{conv}_\psi(\psi_r); f'_\theta), \\ z_{fa} &= \text{concat}(\{z_A, z_F, z_\psi | z \in \mathbb{R}^d\}), \end{aligned} \quad (22)$$

where z_A, z_F, z_ψ denote the features representation of A_r, F_r, ψ_r under CNN f'_θ corresponding convolutional layer channels $\text{conv}_A, \text{conv}_F, \text{conv}_\psi$.

Space learning. Since the AMOSR task in this paper relies on distance measures for prototype and non-adjacent points (see in subsection 3.4.2), we need to align the relevant points in the fusion space z_{fu} and the factoring space z_{fa} . The above operation provides a uniform spatial reference, which fairly integrates the distances between samples and prototype points from different spaces. We therefore define the prototype and non-adjacent points in the fusion space as $\mathcal{P}^{fu}, \mathcal{P}_n^{fu}$. Similarly, $\mathcal{P}_n^{fa}, \mathcal{P}^{fa}$ represent the points in the factoring space. Then, we align the non-adjacent and prototype points for each category through Eq 23:

$$\min \frac{1}{K} (\|\mathcal{P}_n^{fu} - \mathcal{P}_n^{fa}\|_2^2 + \|\mathcal{P}^{fu} - \mathcal{P}^{fa}\|_2^2). \quad (23)$$

Fig. 7 shows that the prototype points and non-adjacent points of the factoring space are synchronously aligned towards their corresponding points in the fusion space. Finally, we can calculate their relevant distance differences under a uniform spatial metric.

3.4.2 Similarity Calculation

Once the fusion and factoring spaces are aligned and the CNN is trained jointly, we need to evaluate the performance of the AMOSR task. We measure the difference between the test sample and the known sample by using a similarity calculation. Therefore, we calculate the similarity by computing the distance between the test sample and the prototype points in the two spaces separately. So the MMPRF use Eq. 24 to calculate global prototype points for each class against the training set data.

$$\mathcal{P}^{fu} = \frac{1}{N} \sum_{i=1}^N \|z_{fu}\|_2^2, \mathcal{P}^{fa} = \frac{1}{N} \sum_{i=1}^N \|z_{fa}\|_2^2. \quad (24)$$

Then, for each sample, its distance relative to the prototype \mathcal{P}^{fu} and \mathcal{P}^{fa} in two feature space is jointly calculated by

$$\begin{aligned} \ell(x(n); \mathcal{P}^{fu}, \mathcal{P}^{fa}) &= (\lambda \ell(f'_\theta(x(n)), \mathcal{P}^{fu}) \\ &\quad + (1 - \lambda) \ell(f'_\theta(x(n)), \mathcal{P}^{fa})), \end{aligned} \quad (25)$$

Algorithm 2: MMPRF Algorithm for AMOSR Task

```

Input: Training set  $\mathcal{D}_{tr}$ , initialized CNN parameters
         $f_\theta, f'_\theta$ , non-adjacent points  $\mathcal{P}_n^{fu}, \mathcal{P}_n^{fa}$ ,
        prototype  $\mathcal{P}^{fu}, \mathcal{P}^{fa}$ , and hyperparameter  $\lambda$ 
Output: Prediction results  $\tilde{y}_k$ 
1 Initializing the network parameters  $f_\theta, f'_\theta$ ;
2 for epoch = 1 to Epoch do
3   for batch in  $\mathcal{D}_{tr}$  do
4     Feature extraction
5        $z_{fu} = f_\theta(x_{fu}(n)), z_{fa} = f'_\theta(x_{fa}(n))$ ;
6       Calculate local  $\mathcal{P}_n^{fu}, \mathcal{P}_n^{fa}$  and  $\mathcal{P}^{fu}, \mathcal{P}^{fa}$  for
7          $z_{fu}, z_{fa}$  by MPRF;
8       Update parameter  $f_\theta, \mathcal{P}_n^{fu}, \mathcal{P}^{fu}$ ;
9       Frozen  $\mathcal{P}_n^{fu}, \mathcal{P}^{fu}$  and space learning for
10       $\mathcal{P}_n^{fa}, \mathcal{P}^{fa}$  by Eq. 23;
11      Update parameter  $f'_\theta, \mathcal{P}_n^{fa}, \mathcal{P}^{fa}$ ;
12   end
13   Calculate global prototype  $\mathcal{P}^{fu}, \mathcal{P}^{fa}$ ;
14   Calculate threshold  $\zeta^*$  by Eq. 26;
15 end
16 Predicting test set  $x(n)'$  to get  $\tilde{y}_k$  by Eq. 27;
17 return  $\tilde{y}_k$ 

```

where λ denotes the hyperparameter. Similarity is expressed in terms of distance, and the smaller distances the higher similarity.

Finally, the test set samples will first be determined belonging to a class \tilde{y}_k based on the distance by Eq. 12 and compared with the corresponding class threshold ζ_k to determine whether they are unknown samples. To obtain the class threshold ζ^* , we consider the distance containing $\delta\%$ of the samples, which can be calculated as follows

$$\zeta^* = \arg \min_{\zeta_k} \{p[\ell(\mathcal{D}_{tr}^{y_k}; \mathcal{P}_k^{fu}, \mathcal{P}_k^{fa}) < \zeta_k] > \delta\}_{k=1}^K, \quad (26)$$

where ζ^* denotes the optimal threshold. The OSR confirmation calculation is shown below

$$\begin{aligned} \tilde{y}_k &= \arg \max_{y_k} (p(y = k | x(n)', f_\theta, \mathcal{P}_n^{fu})) \\ \tilde{y}_k &= \begin{cases} y_k, & \text{if } \ell(x(n)'; \mathcal{P}_k^{fu}, \mathcal{P}_k^{fa}) \leq \zeta^*, \\ y_{K+1}, & \text{else.} \end{cases} \end{aligned} \quad (27)$$

Alg. 2 describes the working procedure of MMPRF. We first perform feature extraction to obtain the z_{fu} and z_{fa} (Line 4). Then we calculate the prototype points and non-adjacent points in the two spaces separately by Eq. 24 (Line 5). The fusion features are optimized to obtain the updated parameters (Line 6). The space alignment operation will then update the parameters of the factoring features by Eq. 23 (Line 7-8). Finally, the prediction task is performed by Eq. 26, 27 (Line 10-14).

4 ONLINE ADVERSARIAL ENHANCEMENT OF MM-PRF

In the real physical world, our proposed MMPRF still has limited performance due to the presence of a large number of unknown samples. We need to let the deep learning model imagine sufficient unknown samples thereby adapting to the unknown world. In this paper, we further propose

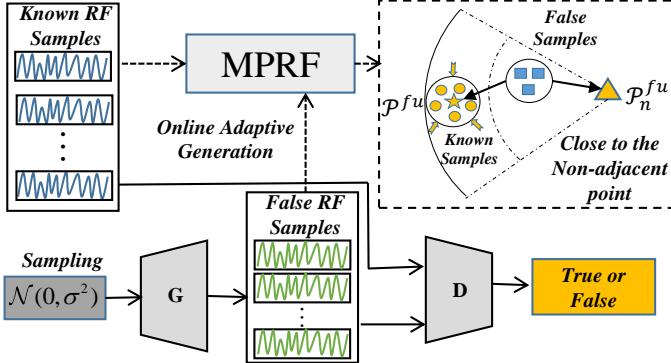


Fig. 8. The basic framework of MMPRF+.

the MMPRF+ algorithm which mainly uses GAN to generate unknown RF signal samples. When the distribution of unknown samples is difficult to predict, the key advantage of GAN is to generate unknown samples of arbitrary distribution [10], [28], [29]. However, in traditional GAN-based generation methods, the GAN is directed to generate samples with a high degree of similarity to the known sample distribution [30], which is not applicable in our AMOSR scenario. For the generated unknown samples we expect that they should be both somewhat similar and distinct from each other with respect to the known sample distribution [21]. Once the generated samples are highly similar to the known samples, it will disrupt the recognition performance of the neural network. To achieve the above purpose, we propose a GAN-based online adaptive generation method.

4.1 Online Learning Enhancement

As shown in Fig. 8, the MMPRF+ first utilizes GAN to generate samples that are similar to known samples but far from the prototype points. These samples are then added to the MPRF algorithm for joint training to allow the CNN to understand unknown classes.

Sample generation. A GAN consists of a generator G_θ and a discriminator D_θ , where the generator G_θ randomly samples from a priori Gaussian distribution latent $z = \text{sample}(\mathcal{N}(0, \sigma^2))$ to generate false samples, where σ denotes variance. The discriminator D_θ discriminates the authenticity of the samples, i.e., True or False. First, the GAN will train its discriminator D_θ to have superior discriminatory performance. Given a priori latent distribution $\{z_i\}_{i=0}^N$ and training data $\mathcal{D}_{tr} = \{x_i(n)\}_{i=0}^N$, the discriminator D_θ is optimized through Eq. 28.

$$\max_{D_\theta} \frac{1}{N} \sum_{i=1}^N [\log D_\theta(x_i(n)) + \log(1 - D_\theta(G_\theta(z_i)))]. \quad (28)$$

Then, the generator G_θ , in order to be able to generate more realistic samples, is optimized by generating a sample to spoof the discriminator D_θ , which can be calculated as

$$\min_{G_\theta} \frac{1}{N} \sum_{i=1}^N -\log D_\theta(G_\theta(z_i)). \quad (29)$$

Finally, after a continuous game of generator G_θ and discriminator D_θ with GAN, we will generate samples that are similar to known samples, but not unknown samples.

Generation of unknown samples. To generate unknown samples, our key idea is to add the generated samples online

to the MPRF training process, which means that we can be able to control the sample generation direction of the GAN in real-time. As shown earlier, in MPRF, non-adjacent points define non-target class regions and prototype points define target class regions, which suggests that the unknown class of samples should be close to the non-adjacent points. As shown in Fig. 8, we aim to generate similar samples of target class that are close to the non-adjacent points P_n and away from the prototype points P . Therefore, we use the triplet loss (see Eq .30) to control the direction of movement of generating unknown samples in the feature space. Then the direction of sample generation for the generator G_θ can be controlled by the following equation:

$$\min_{G_\theta} \frac{1}{N} \sum_{i=1}^N \max\{\ell(G_\theta(z_i), P_n) - \ell(G_\theta(z_i), P) + \mathcal{R}, 0\}. \quad (30)$$

The Eq. 30 shows that the generated samples $G_\theta(z_i)$ is close to the non-adjacent points P_n and always have a margin of distance \mathcal{R} from the prototype point P . We then define the cost function $\mathcal{J}(G_\theta(z_i))$ for simplification purposes

$$\mathcal{J}(G_\theta(z_i)) = \max\{\ell(G_\theta(z_i), P_n) - \ell(G_\theta(z_i), P) + \mathcal{R}, 0\}. \quad (31)$$

We sum the Eq. 29 and Eq. 31, and the optimization goal of generator G_θ is formalized below.

$$\min_{G_\theta} \frac{1}{N} \sum_{i=1}^N (-\log D_\theta(G_\theta(z_i)) + \gamma \mathcal{J}(G_\theta(z_i))), \quad (32)$$

γ denotes the hyper-parameter.

Training of unknown samples. Finally, the joint of Eq. 28 and Eq. 32 can generate a large number of unknown samples online that are far from the prototype point. The generated samples are fed to MMPRF for training and the parameters are optimized as follows.

$$\arg \min_{f_\theta, P_n^k, R^k, P^k} \mathcal{L} = \mathcal{L}_c + \mathcal{L}_p + \beta \mathcal{L}_b + \gamma \mathcal{J} \quad (33)$$

Alg. 3 describes the working procedure of MMPRF+. First, the discriminator D_θ is updated via Eq. 28 (Line 4). Then the generator generates fake samples and performs network updates through Eq. 32 (Line 5). Finally, the generated samples are added to the training process and the classifier parameters are updated according to Eq. 33 (Line 6).

5 EVALUATION

5.1 Experiment Setup

Experimental Environment. The MMPRF algorithm is implemented by using Pytorch 1.10.0 and execute on a computer running Ubuntu 18.04.6 LTS, with Intel(R) Core(TM) i9-10900K CPU@3.70 GHz and 2 NVIDIA GeForce RTX3090 GPUs. In addition we mainly use Adam [31] optimizer for optimization of CNN parameters and Table 2 depicts the relevant parameters for our MMPRF.

Dataset. Our dataset is primarily derived from publicly available wireless modulation type identification datasets.

- **RADIOML 2016.10A** [32]: A synthetic signal modulation classification dataset, generated with GNU Radio, consisting of 11 wireless modulations (8 digital and 3 analog) with varying SNR. The 8 digital modulations are BPSK, QPSK, 8PSK, 16QAM, 64QAM, BFSK, CPFSK, and

Algorithm 3: MMPRF+ Algorithm for AMOSR Task

Input: Training set \mathcal{D}_{tr} , initialized CNN parameters f_θ, f'_θ , generator G_θ , discriminator D_θ , non-adjacent points \mathcal{P}_n , prototype \mathcal{P} , and hyperparameter $\alpha, \beta, \lambda, \gamma$

Output: The optimal parameters $f_\theta^*, \mathcal{P}_n^*, \mathcal{P}^*$

- 1 Initializing the network parameters $f_\theta, f'_\theta, G_\theta, D_\theta$;
- 2 **for** $epoch = 1$ to $Epoch$ **do**
- 3 **for** $batch$ in \mathcal{D}_{tr} **do**
- 4 Update discriminator parameters D_θ by its gradient:

$$\nabla_{D_\theta} \frac{1}{N} \sum_{i=1}^N [\log D_\theta(x_i) + \log(1 - D_\theta(G_\theta(z_i)))]$$
- 5 Update generator parameters G_θ^* by its gradient:

$$\nabla_{G_\theta} \frac{1}{N} \sum_{i=1}^N (-\log D_\theta(G_\theta(z_i)) + \lambda J(G_\theta(z_i)))$$
- 6 Update the classifier parameters $f_\theta^*, \mathcal{P}_n^*, \mathcal{P}^*$ by its gradient:

$$\nabla_{f_\theta^*, \mathcal{P}_n^*, \mathcal{P}^*} \frac{1}{N} \sum_{i=1}^N [\mathcal{L}(x_i, y_i; f_\theta, \mathcal{P}_n, \mathcal{P}) + \gamma \mathcal{J}(G_\theta)]$$
- 7 **end**
- 8 Update the classifier by executing the MMPRF algorithm 2;
- 9 **end**
- 10 **return** Parameters $f_\theta, \mathcal{P}_n, \mathcal{P}$

TABLE 2
Network Architecture and Parameter Configurations

Parameters	Value	Layers	Activation
Probability threshold (δ)	0.95	Conv1d(4,2,1)	BN+LReLU
Hyper-parameter (α)	0.1	Conv1d(4,2,1)	BN+LReLU
Hyper-parameter (β)	0.01	Conv1d(4,2,1)	BN+LReLU
Generator weights (γ)	0.01	Conv1d(4,1,0)	BN+LReLU
Metric balance weights (λ)	0.5	AdaptiveAvgPool1d	
Learning rate (lr)	0.001	FC(d -dimension)	

PAM4, and 4 analog modulations are WB-FM, AM-SSB, and AMDSB. The SNR ranges is from -20dB to 18dB, and the length of each sample is 128. In our OSR task, **9 classes were set as known and 2 classes as unknown samples.**

- **Sig-2019** [6]: This dataset contains longer signals, which considers several nonideal effects of a real communication system, including carrier phase, pulse shaping, frequency offsets and noise. The dataset consists of 12 modulation categories, including BPSK, QPSK, 8PSK, OQPSK, 2FSK, 4FSK, 8FSK, 16QAM, 32QAM, 64QAM, 4PAM, and 8PAM. The SNR ranges is from -20dB to 30dB, the length of each sample is 512. In our OSR task, **10 classes were set as known and 2 classes as unknown samples.**
- **RADIOML 2018.01A** [11]: This real-world dataset, in addition to simulating a wireless channel, is collected from a real laboratory environment. The type of modulations are OOK, 4ASK, 8ASK, BPSK, QPSK, 8PSK, 16PSK, 32PSK, 16APSK, 32APSK, 64APSK, 128APSK, 16QAM, 32QAM, 64QAM, 128QAM, 256QAM, AM-SSB-WC, AM-

SSB-SC, AM-DSB-WC, AM-DSB-SC, FM, GMSK, OQPSK. The SNR ranges from -20dB to 30dB, and the length of each sample is 1024. In our OSR task, **19 classes were set as known and 5 classes as unknown samples.**

- **HisarMod2019.1** [33]: This dataset contains more models of channel fading in realistic scenarios and provides a better realistic simulation than the previous datasets. It contains 26 modulations with different SNR like AM-DSB, AM-SC, AM-USB, AM-LSB, FM, PM, 2FSK, 4FSK, 8FSK, 16FSK, 4PAM, 8PAM, 16PAM, BPSK, QPSK, 8PSK, 16PSK, 32PSK, 64PSK, 4QAM, 8QAM, 16QAM, 32QAM, 64QAM, 128QAM, 256QAM. In our OSR task, **20 classes were set as known and 6 classes as unknown samples.**
- **MIMOsigRef-SD** [34]: It records modulated signals in different mobile environments through an emulation process, such as M-QAM, MIL-STD-188-110 B/C standard-specific QAM, M-PSK, M-APSK, DVB-S2/S2X/SH standard-specific APSK, and M-PAM with different modulation orders, each with different multiple-input multiple-output (MIMO) system configurations in order to provide an extensive signal reference. In our OSR task, **20 classes were set as known and 10 classes as unknown samples.**

Models' setup and comparison baselines. To ensure the fairness of the comparison, we use a uniform network architecture for all baseline methods. As shown in Fig. 5, to facilitate the fusion learning of multi-channel data in the modal fusion of MMPRF, we use ResNet-18 as the backbone network both for our algorithms and other baseline algorithms. Then in the modal factoring, due to the necessity of using multiple models to learn each individual signal model signal, we reduce the number of parameters, using 3 simple 1DCNNs as the network architecture. Table 2 lists the structural information of the 1DCNN, where Conv1d means 1DCNN that contains parameters filters, stride, and padding respectively. The baseline algorithms are briefly summarized as follows.

- **Openmax** [35]: This method mainly uses extreme value theory (EVT) to calculate the classification probability.
- **GCPL** [9]: This model uses prototypes to represent each known class and force each class to be close to its corresponding prototype point.
- **CGDL** [25]: This method employs a β -variational autoencoder (β -VAE) to decouple the features and uses the reconstruction error of the autoencoder (AE) to perform the OSR task.
- **GCM-CF** [24]: This model extended the CGDL by using counterfactual causal inference to generate unknown samples to augment the unknown classes.
- **SR2CNN** [20]: This model combines the method of autoencoder and prototype point to achieve the AMOSR.
- **Open-GAN** [36]: This model mainly uses the discriminator of GAN to identify unknown samples.
- **RPL** [7]: This model uses reciprocal points to allow the neural network to understand unknown samples to improve the performance of OSR.
- **ARPL** [10]: In this model, the adversarial boundary restrictions is added to RPL to reduce open space risk.
- **ARPL+CS** [10]: This method adopts an GAN to generate unknown samples to support ARPL to perform OSR

tasks.

Evaluation Metrics. Following [9], [10], [25], [35], we use the area under the receiver operating characteristic curve (AUROC) to evaluate the performance of AMOSR. The AUROC has the ability to discriminate between unknown and known by ranking the samples from highest to lowest prediction probability [10]. In particular, it does not require the use of probability thresholds for comparison.

However, AUROC can only discriminate between known and unknown differences in probability distributions and does not evaluate the accuracy of the model's predictions for known classes. Following [21], we introduce a open-set classification rate (OSCR) metric to count the rejection rate of unknown classes and the accuracy of known classes, which consists of correct classification rate (CCR) and false positive rate (FPR). With the distance thresholds ζ^* computed in Section 3.4.2, our CCR represents the proportion of samples whose embedding features in the space are below the target distance threshold ζ_k^* and correctly classified as y_k :

$$\begin{aligned} CCR(x(t) \in (\mathcal{D}_{te} - \mathcal{D}_u); \zeta^*) = \\ \frac{|\{x(t) | \arg \max_k(p(y_k|x(t))) \cap \ell(x(t), \mathcal{P}_k) \leq \zeta_k^*\}|}{|\mathcal{D}_{te} - \mathcal{D}_u|}. \end{aligned} \quad (34)$$

Then the FPR denotes the proportion of samples from an unknown dataset \mathcal{D}_u whose embedding feature distance is higher than the target threshold ζ_k^* , which can be calculated by:

$$\begin{aligned} FPR(x(t) \in \mathcal{D}_u; \zeta^*) = \\ \frac{|\{x(t) | \arg \max_k(p(y_k|x(t))) \cap \ell(x(t), \mathcal{P}_k) > \zeta_k^*\}|}{|\mathcal{D}_u|}. \end{aligned} \quad (35)$$

Therefore, similar to AUROC, OSCR based on CCR and FPR is an indicator that evaluates the model by calculating the area under the corresponding curve [21]. *A larger OSCR indicates that the current model has better OSR performance.*

Methodology. To verify the effectiveness of our proposed method, our experiments are divided into three key sections. In the baseline method (§5.2), we compare the efficiency of the key metrics with the OSR baseline approach. In addition we consider the impact of different SNRs as well as different number of known classes. Then in micro-benchmarking (§5.3), we perform ablation experiments mainly by varying different RF features and hyperparameters. Finally in visualization analysis (§5.4), we visualize some features to discuss in more depth of some key results.

5.2 Experimental results for OSR

Comparison of AUROC and OSCR among OSR baselines

To verify the efficiency of the algorithm, we randomly select five AMOSR tasks for each modulation type dataset, where the setting of the AMOSR category is kept consistent in each baseline. Table 3 shows that MPRF outperforms SOTA methods and the improvement is between 20% and 30%. This indicates that the imposed marginal prototype constraints and the extracted RF modal fusion features effectively improves the performance of AMOSR. In particular, the MMPRF which incorporates modal factoring features on the basis of the MPRF is able to significantly improve the performance of the average 2% on AUROC and OSCR. This illustrates

that the challenge of the catastrophic disappearance of features can be partially mitigated by modal factoring. Finally, to further improve the ability to reject unknown classes, the MMPRF+ uses GAN to generate unknown RF samples and incorporates them into the network training, achieving the best performance on AMOSR.

Comparing to SOTA CSR methods. In addition, as reported in [6], [37], [38], the accuracy of CSR of the SOTA methods in RADIOML 2016.1, Sig-2019 and RADIOML 2018.01A are 94.51%, 98.51% and 99.4%. Our proposed algorithm also achieves similar accuracy at 93.63%, 98.31%, and 99.1%, which demonstrates that our algorithm can improve OSR performance while ensuring the accuracy of CSR.

Comparison of CCR and FPR among OSR baselines. In each dataset, a random set of AMOSR tasks is selected and Fig. 9 shows their OSCR curves, reflecting the variation pattern of CCR and FPR. We can obviously see that in different datasets, all three methods (MPRF, MMPRF, MMPRF+) proposed in this paper have a large OSCR area, indicating the unknown samples have a smaller overlap area with the known samples. At 95% CCR, we can observe the difference in FPR values obtained by different methods, where our method has a smaller FPR value against the other baseline algorithms. The smaller FPR value means that the model has a higher discriminatory ability for unknown samples while ensuring classification accuracy for known classes. In summary, our approach incorporating modulated signal-related features can significantly improve the model's ability to reject unknown samples, which is the key reason that SOTA CV-based OSR algorithms can not perform well in AMOSR tasks.

Comparison at different SNR levels. Since RF signal transmission in the real environment may be affected by noise, we compare the performance of the average AUROC with different SNR levels in a randomized 5 AMOSR task. As shown in the Fig. 10, we can see that at low SNR ([−20, −12]db in Fig. 10a, [−20, −10]db in Fig. 10b and [−20, −10]db in Fig. 10c), our method does not differ much from the baseline methods. This means that noise drowns out all information. As the SNR level increases, the performance of our methods are dramatically improved, compared to the baseline methods. In Fig. 10c, our solution outperforms RPL by 30% in the 15dB and 25dB SNR. The above results show that our method is suitable for real-world scenarios that the SNR is between 10 dB and 30 dB [39].

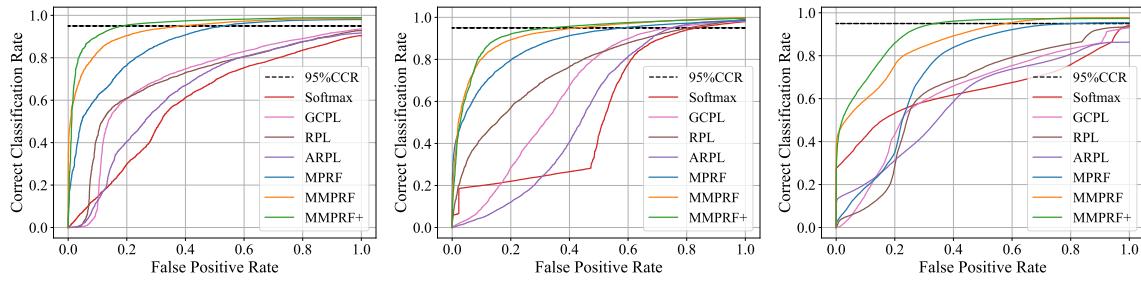
The baseline algorithms perform poorly is because of key features about the AMOSR task are not extracted, such as frequency, phase, under low SNR (i.e., high noise environment). However, the MMPRF in our proposed can effectively extract these features.

Moreover, in Fig. 10c, we find an unpredictable downward trend in the performance of AUROC for several baseline methods as the SNR increases. Tracing the reasons for this, we believe that there may be several points: (i) The dataset RADIOML 2018.01A has more categories and higher recognition difficulty, making the performance poor at high SNR; (ii) At low SNR, the neural network classification recognition performance is poor and the task performance of OSR is at the blind guessing stage (50% uniform performance for binary classification). The increase of SNR makes the classification performance of the network improve, but

TABLE 3

The AUROC and OSCR results of on detecting known and unknown samples. Results are averaged among five randomized trials.

Methods	RADIOML 2016.10A		Sig-2019		RADIOML 2018.01A		HisarMod2019.1		MIMOSigRef-SD	
	AUROC	OSCR	AUROC	OSCR	AUROC	OSCR	AUROC	OSCR	AUROC	OSCR
Softmax	53.13	52.26	51.26	51.02	56.35	53.97	81.10	81.10	80.35	80.35
Openmax [35]	60.20	56.99	53.20	46.72	58.24	54.90	77.61	77.42	78.41	78.30
GCPL [9]	67.72	64.02	70.15	69.76	66.09	61.06	86.51	86.51	82.80	82.80
CGDL [25]	52.15	46.99	57.44	52.85	54.28	50.56	63.78	62.15	56.48	56.48
GCM-CF [24]	49.09	45.88	64.42	62.08	41.57	40.64	60.02	56.38	55.46	55.43
SR2CNN [20]	76.74	68.34	58.46	56.73	90.22	79.28	79.90	79.07	84.07	78.94
Open-GAN [36]	69.91	65.87	73.60	71.44	82.62	74.56	66.96	66.55	62.07	58.11
RPL [7]	72.17	67.18	71.55	70.78	66.87	57.45	88.64	88.64	86.59	86.59
ARPL [10]	64.73	59.38	58.69	57.96	37.67	35.41	76.10	76.10	64.86	64.86
ARPL+CS [10]	54.23	50.70	60.72	59.30	45.92	41.69	60.85	60.68	56.21	56.21
MPRF (Ours)	87.93	81.39	87.97	87.10	91.50	86.07	94.28	94.16	92.75	84.07
MMPRF (Ours)	89.54	87.75	90.11	89.27	94.06	88.44	94.53	94.40	95.62	87.47
MMPRF+ (Ours)	92.56	90.04	92.42	92.14	95.49	90.04	96.64	96.63	96.57	90.06

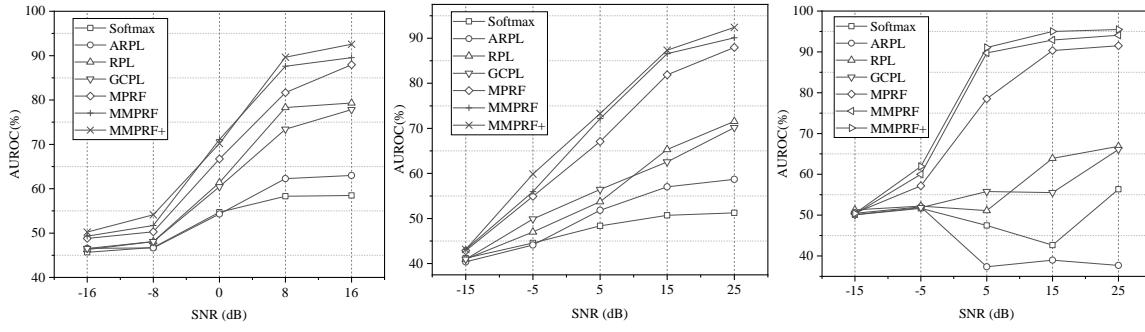


(a) RADIOML 2016.10A

(b) Sig-2019

(c) RADIOML 2018.01A

Fig. 9. Thr curves of OSCR include metrics CCR, FPR: (a) in dataset RADIOML 2016.10A, the WBFM, QPSK as unknown modulation types; (b) in dataset Sig-2019, the 2FSK, 8FSK as unknown modulation types; (c) in dataset RADIOML 2018.01A, the GMSK, 32APSK, OQPSK, 8ASK, BPSK as unknown modulation types.



(a) RADIOML 2016.10A

(b) Sig-2019

(c) RADIOML 2018.01A

Fig. 10. AUROC performance for different levels of SNR: (a) for dataset RADIOML 2016.10A, we divided the SNR range [-20,18] dB into five intervals on average; (b) for dataset Sig-2019, we divided the SNR range [-20,30] dB into five intervals on average; (c) for dataset RADIOML 2018.01A, we divided the SNR range [-20,30] dB into five intervals on average.

TABLE 4

The OSCR results for different K known classes and $(N - K)$ unknown classes in Sig-2019.

	K=2 (%)	K=4 (%)	K=6 (%)	K=8 (%)	K=10 (%)
MPRF	90.43	94.68	97.60	84.66	89.21
MMPRF	91.73	96.60	98.80	86.92	97.81
MMPRF+	92.63	99.53	99.05	87.01	99.75

it also identifies unknown classes as known classes, which leads to some algorithm performance degradation under high SNR instead.

Varying the number of known classes.. We change the number of known classes and evaluate their critical OSCR performance. As shown in Table 5, our approach achieves

a stable performance while varying numbers of known classes, compared to baselines. Also, in the dataset Sig-2019 and RADIOML 2018.01A, the reduction of the number of known classes leads to a decrease of OSCR performance. This suggests that more category samples allow neural network to learn more knowledge to further improve the performance of AMOSR. Then in Table 4, we show the results for different K known classes as well as $N - K$ unknown classes on Sig-2019. It is clear to observe that our proposed MMPRF+ has better OSCR performance. At first ($K=2, 4, 6$) OSCR recognition rate increases when the number of known classes increases. But when $K=8$, the recognition rate shows a decrease. The above results indicate that when the number of known classes becomes larger, the

TABLE 5
The OSCR results for different number of known classes.

Methods	RADIOML 2016.10A				Sig-2019			RADIOML 2018.01A		
	K=10 (%)	K=7 (%)	K=6 (%)	K=11 (%)	K=9 (%)	K=7 (%)	K=23 (%)	K=17 (%)	K=12 (%)	
Softmax	57.4	77.0	73.9	56.0	41.9	59.4	70.4	49.9	66.2	
GCPL	71.0	80.9	80.3	86.2	60.3	62.3	85.6	81.8	74.7	
RPL	78.4	80.4	81.7	60.0	66.6	63.1	86.2	64.5	68.6	
ARPL	61.0	68.7	84.6	64.1	41.7	66.8	81.0	36.9	54.7	
MPRF	85.9	91.0	87.4	91.1	91.1	75.0	89.0	89.1	84.4	
MMPRF	88.2	91.6	89.7	95.7	93.5	73.6	92.8	92.3	87.5	
MMPRF++	88.9	93.5	85.4	97.7	93.7	74.5	93.3	92.4	90.3	

TABLE 6

The results of AUROC, OSCR and boosting (Δ) on detecting known and unknown samples by augmenting feature to baselines. In column Δ , the left side indicates the boost of AUROC and the right side indicates the boost of OSCR.

Methods	RADIOML 2016.10A			Sig2019-12			RADIOML 2018.01A		
	AUROC(%)	OSCR(%)	$\Delta(\uparrow \%)$	AUROC(%)	OSCR(%)	$\Delta(\uparrow \%)$	AUROC(%)	OSCR(%)	$\Delta(\uparrow \%)$
Softmax+	64.21	61.42	20.85/17.52	64.21	61.42	55.62/49.73	64.01	61.12	13.59/13.24
GCPL+	75.34	71.58	11.25/11.80	69.57	68.64	28.92/27.67	71.93	70.71	8.83/15.80
RPL+	79.79	74.44	10.55/10.81	79.79	74.44	31.01/24.52	78.12	75.93	16.82/32.16
ARPL+	66.36	63.22	2.51/6.46	66.36	63.22	13.06/9.07	53.59	51.45	42.26/45.29
MPRF	87.93	81.39	-/-	87.97	87.10	-/-	98.82	97.22	-/-

probability that the unknown classes is similar to known classes increases.

5.3 Micro-Benchmarking

In this section, we discuss the impact of hyperparameters as well as some key features for ablation.

Efficiency of data augmentation. We first impose the key RF features (phase, frequency and amplitude) discussed in this paper, and then evaluate impact of augment features to various models' performance gains.

Table 6 shows that our data augmentation method can improve the performance of all algorithms and our proposed methods can still achieve the best performance, compared the boosted methods (i.e., adding the data augmentation to the baseline solutions). The Δ column indicates the average performance increase on AUROC as well as on OSCR, where the performance can be improved by about 5%-50%, which shows that the RF features used in this paper can improve the efficiency of AMOSR for different algorithms.

Ablation analysis of different RF features. In order to verify the impact of different features (e.g., $F(w)$ and $\psi(t)$), we perform an ablation analysis on the MPRF algorithm.

In the Fig. 11 shows four types of settings, i.e., 1) performing MPRF with both $F(w)$ and $\psi(t)$ (noted as **MPRF**), 2) without $F(w)$ (noted as **MPRF(w/o F(w))**), 3) without $\psi(t)$ (noted as **MPRF(w/o $\psi(t)$)**), 4) without both $F(w)$ and $\psi(t)$ (noted as **MPRF(w/o F(w), $\psi(t)$)**).

For FPR, the MPRF with $F(w)$ and $\psi(t)$ has the lowest FPR in most AMOSR tasks and the MPRF without $F(w)$, $\psi(t)$ has the highest FPR value across different data sets and different openset tasks. Next, the MPRF with $F(w)$ or with $\psi(t)$ produce different degrees of reduction on FPR, but this reduction is varying among different openset tasks. For example, in the openset 2 of Sig-2019, the MPRF without $F(w)$ does not provide a significant performance gain. The above results show that it is difficult to guarantee that individual features will be effective in all tasks. Therefore our MPRF operates on the fusion of different features, hoping to maintain superior performance in all AMOSR tasks.

TABLE 7
The OSCR results of AMOSR for different RF features. Results are averaged among five AMOSR tasks.

	RADIOML 2016.10A	RADIOML 2018.01A	Sig-2019
MPRF(w/o $F(w), \psi(t)$)	70.67	83.36	70.38
MPRF(w/o $\psi(t)$)	77.89	86.65	78.23
MPRF(w/o $F(w)$)	76.66	86.13	70.50
MPRF	84.87	87.27	87.1

Fig. 11 shows that the same trend is observed in the OSCR metrics, where the MPRF without $F(w)$ and $\psi(t)$ has the lowest OSCR performance and our MPRF has the best performance among different dataset and openset tasks.

Table 7 shows the average OSCR results of five openset tasks across three datasets. The proposed feature extraction method can bring considerable performance gain to the AMOSR. However, in openset 5 of dataset RADIOML 2018.01A, the MPRF has a higher FPR and a smaller OSCR than the MPRF without $F(w)$. The above results suggest that the specific RF features are critical to a particular OSR task. Once the feature is fused with other features for the recognition task, it may face a situation that the feature is weakened or disappears, i.e. the other features dominate the CSR task. Therefore, in this paper, we propose MMPRF to extract these feature differences individually to prevent feature weakening.

Ablation analysis of hyperparameters. Next we analyze impact of the hyperparameters α, β for MPRF, where α denotes the restriction of target class samples to non-adjacent points \mathcal{P}_n and β denotes the restriction of prototype points. Fig. 12 shows the OSCR performance in each dataset under various values of α and β . The results show that the area of OSCR is the smallest when α and β are set to 0. Furthermore, the area of OSCR increases with the increase of α and β .

The above observations illustrate that it is necessary to fix the position of the prototype points and restrict the distance of sample to non-adjacent points in radius \mathcal{R} . Once the sample is not restricted in the range of radius \mathcal{R} from the non-adjacent points, i.e. $\alpha = 0$, which means that the distribution of the known samples is not compact enough. This is because the space outside the range of radius

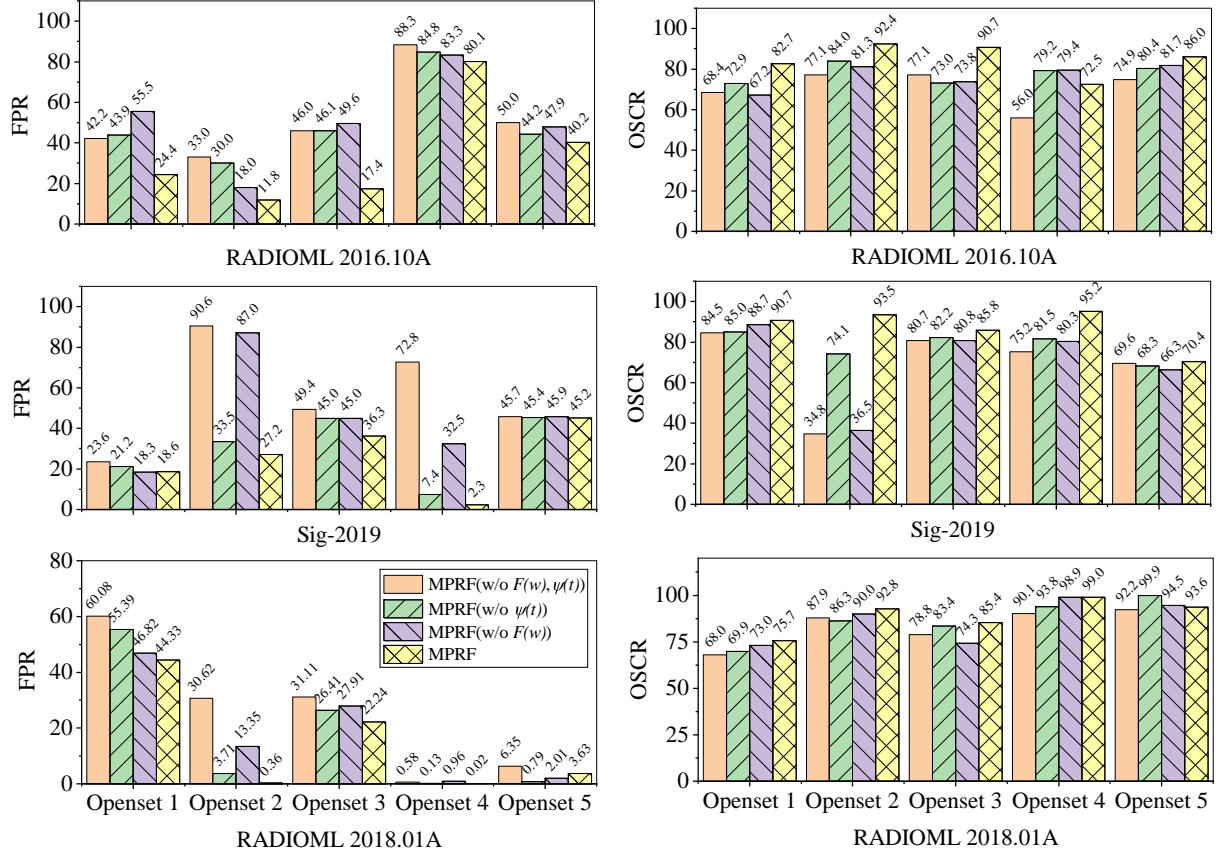


Fig. 11. The FPR and OSCR results for different features in the MPRF algorithm. We form different combinations of features $F(w)$ and $\psi(t)$ in the MPRF. Overall, MPRF with $F(w)$ and $\psi(t)$ has a smaller FPR as well as higher OSCR performance in different OSR tasks.

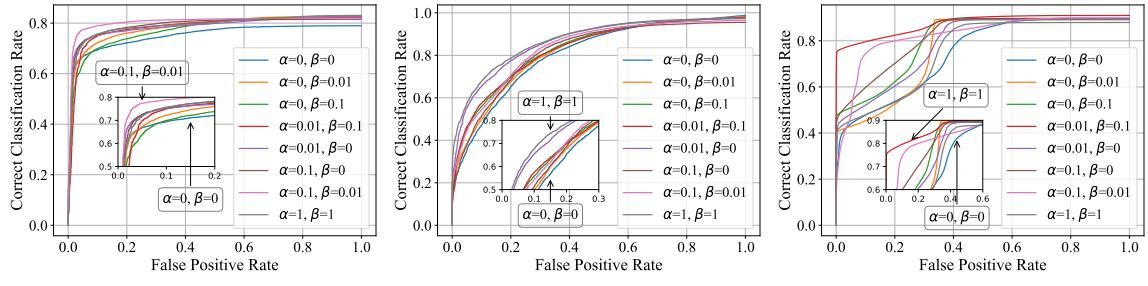


Fig. 12. OSCR curves under the influence of different hyperparameters α, β , where α denotes the restriction of target class samples to non-adjacent points and β denotes the restriction of prototype points.

\mathcal{R} is infinite, which increases the risk of open space. In addition, when $\beta = 0$, the prototype points may not be at the boundary, which likely reduce the gap between known classes thereby reducing the performance of AMOSR.

5.4 Visualization Analysis

t-SNE visualization for different methods. We visualize the key results on the dataset Sig-2019 obtained from several methods proposed, where the category 4PAM and 8PAM are used as the unknown class. As shown in the Fig. 13, We perform a t-SNE based visual analysis of the feature embedding space for known and unknown classes. Fig. 13a-13d demonstrate that the distances between unknown classes and known classes is increasing, and our MMPRF+ has the farthest distance. This indicates that MMPRF+ can

more easily identify the unknown classes, compared to others. The above results show that it is necessary to equip MPRF with the ability to identify unknowns by introducing non-adjacent points as opposed to the PRF.

Additionally, the measurement of extracting different features about the RF signal (MMPRF) and generating unknown samples (MMPRF+) are useful for AMOSR. In Fig. 13e-13h, we plot the distance distribution between the unknown class and the top-2 classes with the highest classification probability. The results show that the unknown class is mostly divided into class 8 (C8) and class 9 (C9), which indicates that the unknown class is more similar to C8, C9. In fact, C8 and C9 denote 32QAM and 64QAM, which have the part of the same amplitude modulation as

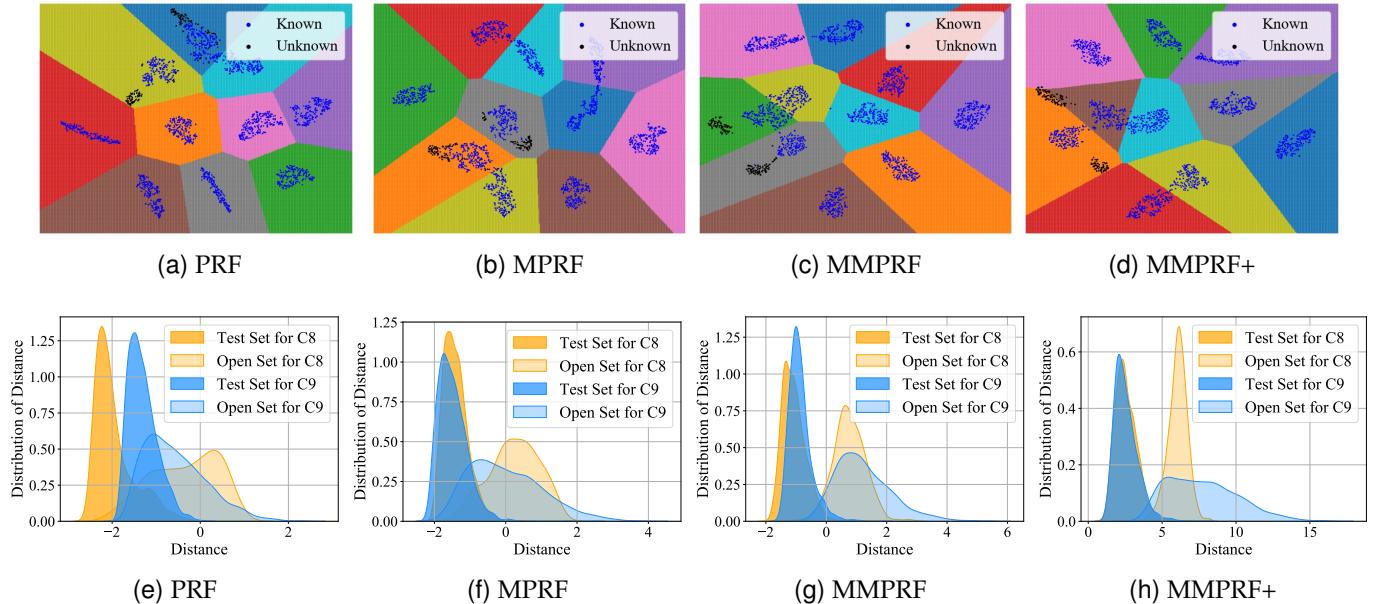


Fig. 13. We performed the visualization of the AMOSR task for the dataset Sig-2019, where we defined the unknown class as 4PAM, 8PAM: figure (a), (b), (c) and (d) denote t-SNE visualization results for different methods, where black denotes the unknown class, blue denotes the known class; figure (e), (f), (g) and (h) denote the comparison of the distance distribution between the unknown class and the top two classes of classification probability. In particular, PRF indicates that the prototype point-based AMOSR classification method targets RF signals (PRF).

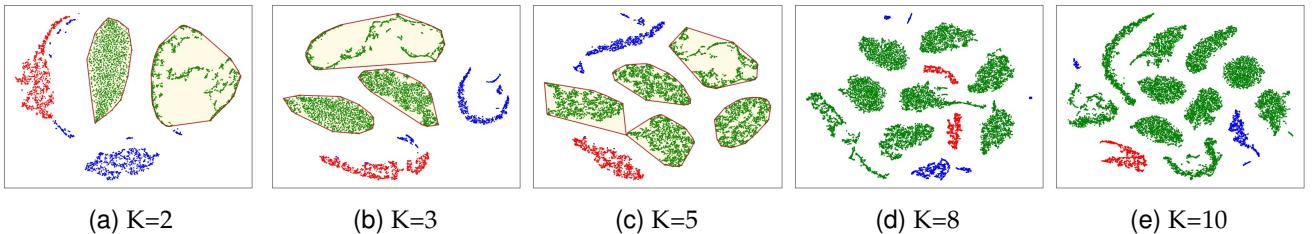


Fig. 14. The t-SNE visualization results of MPRF on known and unknown classes with different numbers of known classes. In the figure, green indicates the known classes of dataset RADIOML 2016.10A, and datasets Sig-2019 and RADIOML 2018.01A indicate the unknown classes, where red indicates dataset Sig-2019 and blue indicates dataset RADIOML 2018.01A.

the unknown class. However, the distance asymptotes from the method PRF to MMPRF+ indicates that the proposed methods improves the performance of the AMOSR task.

T-SNE visualisation for AMOSR tasks under cross-dataset. We visualize the feature space distributions of different number of known classes using t-SNE and test the AMOSR performance under cross-datasets. We use RADIOML 2016.10A as the known dataset for training and then test the model over Sig-2019 and RADIOML 2018.01A as the unknown dataset. Fig.14 displays the representation of both training and testing dataset, where the green area indicates the dataset RADIOML 2016.10A, and the red and blue area represent dataset Sig-2019 and dataset RADIOML 2018.01A. The above results show that our method has an excellent performance in rejecting the samples belonging to unknown classes. However, the performance of AMC is affected by cross-datasets, where some modulation types from RADIOML 2016.10A exist in Sig-2019 and RADIOML 2018.01A, but they are recognized as unknown classes. This may be caused the signals generated from various devices, make modulation identification difficult. In future work, we would like to tackle this problem by setting up a real-world test-bed to obtain a dataset that has the same modulation across groups of devices.

6 RELATED WORK

6.1 Automatic Modulation Classification

AMC is considered to be one of the key techniques for communication systems with adaptive modulation capabilities. Initially, traditional AMC research approaches are mainly based on maximum likelihood estimation (MLE), which considers modulation classification as a multiple-hypothesis testing problem. Various algorithms are proposed around the likelihood method, for example, Huan et al. [40] propose an algorithmic classifier based on the likelihood function (LF). However, the real-time nature and exact likelihood values of the ML-AMC algorithm remain difficult to achieve in complex environments due to the limitations of the algorithm itself [1]. After that, the researcher proposes the feature-based AMC algorithms [41], [42], mainly through data pre-processing, feature extraction and machine learning-based classification decisions. Since the excellent statistical features of feature-based AMC algorithms can achieve near-optimal performance with low complexity, a lot of research has been conducted on feature extraction, such as the use of channel state information (CSI) [43], [44], statistical learning based on higher-order moments and products [45], [46], [47], wavelet transform [48], etc. The above AMC is a combination of different feature extraction algorithms and ML classifiers, which require

expert knowledge in signal processing and experience in manual design.

Recently, deep learning-based modulation classifiers have gradually become mainstream, as they can automatically extract knowledge. O’Shea et al. [11] employ a VGG architecture to design a 1D-CNN model that makes it suitable for small radio signal classification tasks and achieves great results. Xu et al. [12] use this 1D convolution and narrow 2D convolution to propose a multi-channel CNN, named MCLDNN. Unlike the above, Chen et al. [6] propose the SigNet by accelerating the 1D convolution operator operation, which achieves more advanced performance compared to existing baseline methods. On the other hand, for the time characteristics of RF signals, researchers have used long short-term memory networks (LSTM) [15] to extract timing-related feature information, e.g., Rajendran et al. [16] use an LSTM-based model for modulation classification in a distributed wireless spectrum-aware network.

The current extensive AMC literature can be summarized as the following approaches: 1) use of signal processing tools with machine learning techniques to extract statistical features, and frequency domain features [1], [41], [42]; and 2) use of deep learning techniques [6], [11], [14], [16]. These algorithms are designed for closed-set recognition and it is difficult to guarantee their effectiveness on OSR which use SoftMax to compute the probability of each class. This leads to the fact that the open-set samples are also considered known samples and thus cannot reject the unknown classes. Compared with them, our paper not only draws on the signal processing techniques but more crucially provides an advanced open-set recognition classifier for wireless signals.

6.2 Open Set Recognition

Walter Scheirer et al. [22] first define the OSR problem and propose a basic framework to analyze it. Recent work [49] has proposed Semantic Shift Benchmark (SSB) that can define easy and hard splits based on the semantic similarity between open-set categories and training classes. The study confirms the correlation between strong closed-set and open-set performance for a semantic shift for a family of scoring rules, such as logit and softmax scores. In AMOSR, both distances of modulation frequency and SNR are key factors when considering the class novelty. Semantic shift pertains to the degree of similarity between the unknown and known classes, where a higher similarity leads to the majority of open-set failures. OSR problems are divided into three main categories, including discriminative models, generative models [50], and prototype learning [51].

In discriminative models, the detection of unknown classes in OSR can be regarded as a binary classification. Based on deep learning models, the most famous OSR method is designed using Openmax [35], which mainly uses the extreme value theory (EVT)-based Openmax layer instead of the Softmax layer in the neural network. Another study based on the reconstruction error of auto-encoders (AE) is carried out, where researchers have concluded that unknown samples have a greater reconstruction error than known samples, with the algorithms CROSR [52] and C2AE [53] achieving great performance. Then in generative models, GAN and variational auto-encoder (VAE) models are

widely used to generate unknown samples in order for DNNs to understand the unknown world. Many algorithms have been proposed, e.g. Zong et al. [28] propose a G-Openmax model combining GAN and Openmax. In [24], VAE is used to generate counterfactual images as unknown samples to improve the network’s understanding of the unknown world and achieve SOTA performance of OSR. The generative model has now greatly improved OSR performance and has become the dominant model [21], as it is able to see a lot of samples from an unknown world compared to the discriminative model.

Another mainstream approach is based on the idea of prototypes. Yang et al. propose the GCPL model [9], which uses a prototype to represent each known class in the feature space and forces the features of the training data to be close to the corresponding prototype. The advantage of GCPL is that it not only effectively reduces the empirical classification risk, but also increases the compactness of each class in the feature space, which helps to reduce the risk of open spaces. However, as it does not take into account the location of the prototypes, Chen et al. further proposed the RPL/ARPL algorithm [10], which mainly encourages the training data to move away from each other thereby reducing the overlap of the unknown sample space.

There exists a few AMOSR studies [20], [54], [55]. In more detail, these works suffer from the following shortcomings. (1)These works directly use the algorithm from computer vision tasks and do not extract the significant features that can be used to classify the known and unknown signals. (2) The existing work does not efficiently utilize the unique features extracted signals to increase the distance between the known and unknown classes. (3) The generalization of the existing work is limited, these can achieve good performance on some specific datasets and OSR tasks, but the performance can not be guaranteed when the datasets or OSR tasks are changed. Therefore, our study bridges these gaps. We propose an advanced OSR algorithm and combine it with signal processing knowledge. More importantly, the datasets and algorithms used in the existing research content are relatively homogeneous, and we have evaluated a large number of different datasets and algorithms that have verified their robustness of the algorithms.

7 CONCLUSION

In this paper, we first present the concept of the AMOSR task and review existing OSR methods. We describe the key challenges in the AMOSR task and propose a signal-specific feature extraction method. Based on this, the MPRF algorithm is proposed to effectively reduce the risk of open space by integrating the properties of prototype points and non-adjacent points. The MMPRF algorithm is then further proposed to improve the efficiency of the AMOSR task by factoring in multi-modal features. Finally, the GAN-based MMPRF+ algorithm allows CNN to see and learn as many unknown samples. Although our approach has achieved some improvement in results, there are still significant challenges with AMOSR tasks, i.e. it is difficult for us to guarantee good performance on all AMOSR tasks. In the future, we will conduct further in-depth research to improve the performance of difficult AMOSR tasks.

8 ACKNOWLEDGMENTS

This work was supported by National Natural Science Foundation of China under Grant 62072408 and 62073292, Zhejiang Provincial Natural Science Foundation of China under Grant LY20F020030, and New Century 151 Talent Project of Zhejiang Province.

REFERENCES

- [1] 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*, 2021.
- [2] F. Li, "Research on radar signal recognition based on automatic machine learning," *Neural Computing and Applications*, vol. 32, no. 7, pp. 1959–1969, 2020.
- [3] Y. Chen, M. Yang, J. Long, D. Xu, and F. Blaabjerg, "A dds-based wait-free phase-continuous carrier frequency modulation strategy for emi reduction in fpga-based motor drive," *IEEE Transactions on Power Electronics*, vol. 34, no. 10, pp. 9619–9631, 2019.
- [4] S. N. Daskalakis, R. Correia, G. Goussetis, M. M. Tentzeris, N. B. Carvalho, and A. Georgiadis, "Four-pam modulation of ambient fm backscattering for spectrally efficient low-power applications," *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 12, pp. 5909–5921, 2018.
- [5] A. Ali and W. Hamouda, "Spectrum monitoring using energy ratio algorithm for ofdm-based cognitive radio networks," *IEEE Transactions on Wireless Communications*, vol. 14, no. 4, pp. 2257–2268, 2014.
- [6] Z. Chen, H. Cui, J. Xiang, K. Qiu, L. Huang, S. Zheng, S. Chen, Q. Xuan, and X. Yang, "Signet: A novel deep learning framework for radio signal classification," *IEEE Transactions on Cognitive Communications and Networking*, 2021.
- [7] H.-M. Yang, X.-Y. Zhang, F. Yin, and C.-L. Liu, "Robust classification with convolutional prototype learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 3474–3482.
- [8] A. R. Dhamija, M. Günther, and T. E. Boult, "Reducing network agnostophobia," in *NeurIPS*, 2018.
- [9] G. Chen, L. Qiao, Y. Shi, P. Peng, J. Li, T. Huang, S. Pu, and Y. Tian, "Learning open set network with discriminative reciprocal points," in *Computer Vision-ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*. Springer, 2020, pp. 507–522.
- [10] G. Chen, P. Peng, X. Wang, and Y. Tian, "Adversarial reciprocal points learning for open set recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.
- [11] 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.
- [12] J. Xu, C. Luo, G. Parr, and Y. Luo, "A spatiotemporal multi-channel learning framework for automatic modulation recognition," *IEEE Wireless Communications Letters*, vol. 9, no. 10, pp. 1629–1632, 2020.
- [13] S. Peng, H. Jiang, H. Wang, H. Alwageed, Y. Zhou, M. M. Sebdani, and Y.-D. Yao, "Modulation classification based on signal constellation diagrams and deep learning," *IEEE transactions on neural networks and learning systems*, vol. 30, no. 3, pp. 718–727, 2018.
- [14] Y. Zeng, M. Zhang, F. Han, Y. Gong, and J. Zhang, "Spectrum analysis and convolutional neural network for automatic modulation recognition," *IEEE Wireless Communications Letters*, vol. 8, no. 3, pp. 929–932, 2019.
- [15] K. Greff, R. K. Srivastava, J. Koutník, B. R. Steunebrink, and J. Schmidhuber, "Lstm: A search space odyssey," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 10, pp. 2222–2232, 2016.
- [16] 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.
- [17] S. Rajendran and Z. Sun, "Rf impairment model-based iot physical-layer identification for enhanced domain generalization," *IEEE Transactions on Information Forensics and Security*, vol. 17, pp. 1285–1299, 2022.
- [18] J. Gong, X. Xu, and Y. Lei, "Unsupervised specific emitter identification method using radio-frequency fingerprint embedded infogan," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 2898–2913, 2020.
- [19] R. Akter, V.-S. Doan, T. Huynh-The, and D.-S. Kim, "Rfdoa-net: An efficient convnet for rf-based doa estimation in uav surveillance systems," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 11, pp. 12209–12214, 2021.
- [20] Y. Dong, X. Jiang, H. Zhou, Y. Lin, and Q. Shi, "Sr2cnn: Zero-shot learning for signal recognition," *IEEE Transactions on Signal Processing*, vol. 69, pp. 2316–2329, 2021.
- [21] Z. Xia, P. Wang, G. Dong, and H. Liu, "Adversarial motorial prototype framework for open set recognition," *arXiv preprint arXiv:2108.04225*, 2021.
- [22] W. J. Scheirer, A. de Rezende Rocha, A. Sapkota, and T. E. Boult, "Toward open set recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 7, pp. 1757–1772, 2012.
- [23] L. Peng, J. Zhang, M. Liu, and A. Hu, "Deep learning based rf fingerprint identification using differential constellation trace figure," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 1, pp. 1091–1095, 2019.
- [24] Z. Yue, T. Wang, Q. Sun, X.-S. Hua, and H. Zhang, "Counterfactual zero-shot and open-set visual recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 15404–15414.
- [25] X. Sun, Z. Yang, C. Zhang, K.-V. Ling, and G. Peng, "Conditional gaussian distribution learning for open set recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 13480–13489.
- [26] L. Muda, M. Begam, and I. Elamvazuthi, "Voice recognition algorithms using mel frequency cepstral coefficient (mfcc) and dynamic time warping (dtw) techniques," *arXiv preprint arXiv:1003.4083*, 2010.
- [27] 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, 2019.
- [28] Z. Ge, S. Demyanov, Z. Chen, and R. Garnavi, "Generative openmax for multi-class open set classification," *arXiv preprint arXiv:1707.07418*, 2017.
- [29] L. Neal, M. Olson, X. Fern, W.-K. Wong, and F. Li, "Open set learning with counterfactual images," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 613–628.
- [30] M. Kim, "MI/cgan: Network attack analysis using cgan as meta-learning," *IEEE Communications Letters*, vol. 25, no. 2, pp. 499–502, 2020.
- [31] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Computer Science*, 2014.
- [32] Deepsig dataset: Radioml 2016.10a. Accessed: 2016. [Online]. Available: <https://www.deepsig.io/datasets>
- [33] K. Tekbiyik, A. R. Ekti, A. Görçin, G. K. Kurt, and C. Keçeci, "Robust and fast automatic modulation classification with cnn under multipath fading channels," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. IEEE, 2020, pp. 1–6.
- [34] P. Ghasemzadeh, M. Hempel, S. Banerjee, and H. Sharif, "A spatial-diversity mimo dataset for rf signal processing research," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–10, 2021.
- [35] A. Bendale and T. E. Boult, "Towards open set deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 1563–1572.
- [36] S. Kong and D. Ramanan, "Opengan: Open-set recognition via open data generation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022.
- [37] S. Yunhao, X. Hua, J. Lei, and Q. Zisen, "Convlstmae: A spatiotemporal parallel autoencoders for automatic modulation classification," *IEEE Communications Letters*, vol. 26, no. 8, pp. 1804–1808, 2022.
- [38] Y. Mao, Y.-Y. Dong, T. Sun, X. Rao, and C.-X. Dong, "Attentive siamese networks for automatic modulation classification based on multitime constellation diagrams," *IEEE Transactions on Neural Networks and Learning Systems*, 2021.
- [39] S. Yun and L. Qiu, "Supporting wifi and lte co-existence," in *2015 IEEE Conference on Computer Communications (INFOCOM)*. IEEE, 2015, pp. 810–818.
- [40] C.-Y. Huan and A. Polydoros, "Likelihood methods for mpsk modulation classification," *IEEE Transactions on Communications*, vol. 43, no. 2/3/4, pp. 1493–1504, 1995.
- [41] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4, no. 4.

- [42] M. Bkassiny, Y. Li, and S. K. Jayaweera, "A survey on machine-learning techniques in cognitive radios," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1136–1159, 2012.
- [43] L. Xie and Q. Wan, "Cyclic feature-based modulation recognition using compressive sensing," *IEEE Wireless Communications Letters*, vol. 6, no. 3, pp. 402–405, 2017.
- [44] C. W. Lim and M. B. Wakin, "Compressive temporal higher order cyclostationary statistics," *IEEE Transactions on Signal Processing*, vol. 63, no. 11, pp. 2942–2956, 2015.
- [45] A. Swami and B. M. Sadler, "Hierarchical digital modulation classification using cumulants," *IEEE Transactions on Communications*, vol. 48, no. 3, pp. 416–429, 2000.
- [46] H.-C. Wu, M. Saquib, and Z. Yun, "Novel automatic modulation classification using cumulant features for communications via multipath channels," *IEEE Transactions on Wireless Communications*, vol. 7, no. 8, pp. 3098–3105, 2008.
- [47] A. Smith, M. Evans, and J. Downey, "Modulation classification of satellite communication signals using cumulants and neural networks," in *2017 Cognitive Communications for Aerospace Applications Workshop (CCAA)*. IEEE, 2017, pp. 1–8.
- [48] C.-S. Park, J.-H. Choi, S.-P. Nah, W. Jang, and D. Y. Kim, "Automatic modulation recognition of digital signals using wavelet features and svm," in *2008 10th International Conference on Advanced Communication Technology*, vol. 1. IEEE, 2008, pp. 387–390.
- [49] S. Vaze, K. Han, A. Vedaldi, and A. Zisserman, "Open-set recognition: A good closed-set classifier is all you need," in *International Conference on Learning Representations*, 2022.
- [50] C. Geng, S.-j. Huang, and S. Chen, "Recent advances in open set recognition: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 10, pp. 3614–3631, 2020.
- [51] L. I. Kuncheva and J. C. Bezdek, "Nearest prototype classification: clustering, genetic algorithms, or random search?" *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 28, no. 1, pp. 160–164, 1998.
- [52] R. Yoshihashi, W. Shao, R. Kawakami, S. You, M. Iida, and T. Naemura, "Classification-reconstruction learning for open-set recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4016–4025.
- [53] P. Oza and V. M. Patel, "C2ae: Class conditioned auto-encoder for open-set recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2307–2316.
- [54] Y. Guo, H. Jiang, J. Wu, and J. Zhou, "Open set modulation recognition based on dual-channel lstm model," *arXiv preprint arXiv:2002.12037*, 2020.
- [55] X. Zhang, T. Li, P. Gong, R. Liu, X. Zha, and W. Tang, "Open set recognition of communication signal modulation based on deep learning," *IEEE Communications Letters*, 2022.



Yang Long (Senior Member, IEEE) is an assistant professor in the department of computer science, Durham University. He is also an MRC innovation fellow aiming to design scalable AI solutions for large-scale healthcare applications. His research background is in the highly interdisciplinary field of computer vision and machine learning.



Zhen Hong (Member, IEEE) is a professor with the Institute of Cyberspace Security, and College of Information Engineering, Zhejiang University of Technology, China. His research interests include cyber-physical systems, the Internet of Things, wireless sensor networks, cybersecurity, and data analytics.



Shilian Zheng (Member, IEEE) received the Ph.D. degree in communication and information system from Xidian University, Xi'an, China, in 2014. His research interests include cognitive radio, spectrum management, and deep learning-based radio signal processing.



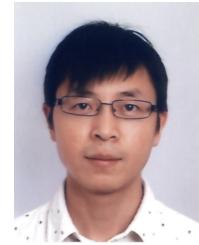
Li Yu (Member, IEEE) is currently a Professor at the College of Information Engineering, Zhejiang University of Technology. He has authored or co-authored three books and over 300 journal papers. His current research interests include cyber-physical systems security, networked control systems, motion control, and information fusion.



Bo Chen (Member, IEEE) Bo Chen received the Ph.D. degree in Control Theory and Control Engineering from the Zhejiang University of Technology, Hangzhou, China, in 2014. His current research interests include information fusion, distributed estimation and control, cyber-physical systems security, and networked fusion systems.



Xiaoniu Yang is currently a chief scientist with the Science and Technology on Communication Information Security Control Laboratory, Jiaxing, China. He is also an Academician of the Chinese Academy of Engineering and a Chinese Institute of Electronics Fellow. His current research interests are software-defined satellite, big data for radio signals, and deep learning-based signal processing.



Ling Shao (Fellow, IEEE) is currently the CEO and the Chief Scientist with the Inception Institute of Artificial Intelligence, Abu Dhabi, United Arab Emirates. His research interests include computer vision, machine learning, and medical imaging. He is a fellow of IAPR, IET, and BCS. He is also an Associate Editor of *IEEE TRANSACTIONS ON IMAGE PROCESSING*, *IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS*, *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, and other journals.



Taotao Li He is currently working toward the Ph.D. degree in control theory and control engineering with the College of Information Engineering, Zhejiang University of Technology, Hangzhou, China. His research interests include machine learning, wireless communication security, and deep learning-based radio signal processing.



Zhenyu Wen (Senior Member, IEEE) is currently a Tenure-Tracked Professor with the Institute of Cyberspace Security and College of Information Engineering, Zhejiang The University of Technology. His research interests include IoT, crowd sources, AI systems, and cloud computing.