Hindawi
Wireless Communications and Mobile Computing
Volume 2023, Article ID 2683780, 14 pages
https://doi.org/10.1155/2023/2683780

Research Article

Semi-supervised Learning for Automatic Modulation Recognition Using Haar Time–Frequency Mask and Positional–Spatial Attention

Hui Liu, Dan Zhong, Yuanpu Guo, Zehong Xu, Zhenlin Wu, and Chunxian Gao

Information and Communication Engineering, Xiamen University, Xiamen, China

Correspondence should be addressed to Chunxian Gao; gaochunxian@xmu.edu.cn

Received 16 February 2023; Revised 8 November 2023; Accepted 13 November 2023; Published 21 December 2023

Academic Editor: Mauro Femminella

Copyright © 2023 Hui Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Automatic modulation recognition plays an important role in many military and civilian applications, including cognitive radio, spectrum sensing, signal surveillance, and interference identification. Due to the powerful ability of deep learning to extract hidden features and perform classification, it can extract highly separative features from massive signal samples. Considering the condition of limited training samples, we propose a semi-supervised learning framework based on Haar time–frequency (HTF) mask data augmentation and the positional–spatial attention (PSA) mechanism. Specifically, the HTF mask is designed to increase data diversity, and the PSA is designed to address the limited receptive field of the convolutional layer and enhance the feature extraction capability of the constructed network. Extensive experimental results obtained on the public RML2016.10a dataset show that the proposed semi-supervised framework utilizes 1% of the given labeled data and reaches a recognition accuracy of 92.09% under 6 dB signals.

1. Introduction

Automatic modulation recognition (AMR) can detect the modulation type of received signal automatically without prior knowledge. It plays a pivotal role in civilian and military applications, such as cognitive radio, signal recognition, spectrum awareness, and electronic warfare. With the increasing number of users, the limited spectrum resources make it difficult to meet the dynamic needs such as 5G [1], etc. This makes AMR a highly challenging task [2].

Typically, traditional AMR methods can be divided into two categories: likelihood-based (LB) methods and feature-based (FB) methods [3]. LB methods [4–6] need prior knowledge and suffer from high-computational complexity. FB methods [7–9] rely heavily on the manual analysis when perform feature selection. Finding distinguishing features among multiple modulation types can be challenging [10].

Recently, inspired by the excellent approaches of deep learning (DL) [11–13], many researchers [14–17] have explored utilizing DL to achieve improved AMR performance. Hong et al. [16] utilized an recurrent neural network (RNN) to extract temporal features automatically, thus reducing the dependency on manual analysis. Yashashwi et al. [17] designed a learnable module that improves signal classification accuracy by correcting frequency offset and phase noise. These supervised learning (SL) methods require extensive labeled data and are prone to overfitting. However, the availability of high-quality labeled data is limited in practical AMR tasks due to the challenges and costs associated with its collection. Furthermore, the performance of AMR on SL methods may be heavily affected by inaccurate or incomplete label.

Therefore, some researchers [18, 19] have applied semi-supervised learning (SSL) to address AMR tasks. However, these SSL methods may suffer from low signal-to-noise ratio (SNR) and encounter difficulties in handling challenging environments such as congnitive radio. To tackle the challenge of low SNR, this paper introduces a novel SSL framework for AMR called HTF-PSA-SSL. The framework leverages a Haar time–frequency (HTF) mask and a positional–spatial attention (PSA) mechanism to enhance modulation recognition accuracy while minimizing the reliance on labeled data. In the first step, the 1D raw IQ signals undergo preprocessing by applying the discrete short-time Fourier transform (STFT). This transformation converts the signals into a 2D STFT

spectrogram, enabling a more comprehensive understanding of the signal's characteristics. Subsequently, we adopt the well-known SSL mean teacher (MT) [20] as the main framework in our approach, enabling us to effectively utilize a larger quantity of unlabeled data alongside the labeled data. Furthermore, we propose a HTF mask that enhances the utilization of unlabeled data by generating augmented samples and mitigating the risk of overfitting. Moreover, to enhance the strip-shaped features of signal, a PSA is added after each convolutional layer to help the network focus on crucial signal regions from time and frequency domain. Finally, to further verify the superiority of the proposed framework, we evaluate the performance of HTF-PSA-SSL on three public datasets, namely, RML2016.10a, RML2016.10b, and RML2016.04c [20]. The evaluation results show that HTF-PSA-SSL not only efficiently utilizes unlabeled data to improve its recognition performance but also achieves beneficial robustness.

The principal contributions of this paper can be summarized as follows:

(1) To address the problem of insufficient labeled signals, we propose a SSL framework, HTF-PSA-SSL, which can effectively improve the modulation recognition accuracy using only a small amount of labeled data. Under 6 dB, HTF-PSA-SSL utilizes 1% of the labeled data and reaches an accuracy of 92.09%. Extensive experimental results obtained on the public dataset RML2016.10a, RML2016.10b, and RML2016.04c show that HTF-PSA-SSL also exhibits strong stability and robustness.

(2) We propose the HTF mask data augmentation method and the PSA mechanism to jointly enhance the performance of HTF-PSA-SSL from data and network. The HTF mask works on unlabeled data, introducing the data perturbations and helping HTF-PSA-SSL to better adapt the different input. The PSA filters strip-shaped features from the time domain and frequency domain, respectively, which helps the network to extract key information from the STFT spectrogram, remove redundant information. Experimental results show that HTF and PSA achieves a highest acuuracy of 93.18% under 16 dB. The ablation experiments further prove the superiority of HTF and PSA in enhancing the performance of HTF-PSA-SSL.

The structure of this paper is organized as follows. In Section 2, the related works are described. Section 3 introduces the signal model. The detailed design and implementation of HTF-PSA-SSL are described in Section 4, and the evaluation results are presented in Section 5. Finally, this paper is concluded by summarizing the proposed work in Section 6.

2. Related Work

2.1. SSL-Based AMR Methods. DL has been widely applied across multiple fields. Li et al. [21] proposed a DL-based remaining useful life (RUL) prediction method to address the sensor malfunction problem by exploring global and shared features. Zhang et al. [22] designed a blockchain-based decentralized federated transfer learning method to further address the data security and privacy problem. In AMR tasks, many SL-based methods [23–30] have achieved significant success. However, the classification performance of these models relies on copious amounts of labeled data, while the amount of labeled data is limited.

To handle this problem, O'Shea et al. [18] trained an encoder to reconstruct signals which first demonstrated that SSL methods could be applied to AMR tasks. Dong et al. [19] proposed an semi-supervised signal recognition convolutional neural network (SSRCNN) to make network more robust by using Kullback–Leibler (KL) divergence loss. Furthermore, Luo et al. [31] introduced the deep cotraining method to construct different sample views by using two CLDNNs [32] with different long short-term memory (LSTM) units which achieves better classification accuracy. Liu et al. [33] build a semi-supervised automatic modulation classification framework (SemiAMC) to efficiently extract features from unlabeled signals. Li et al. [34] introduced generative adversarial networks (GANs) to achieve high-recognition accuracy in AMR tasks. Moreover, Li et al. [35] designed a spatial signal transform module which improves the training stability of the whole SSL framework. And Kim et al. [36] proposed a denoising autoencoder-based relation network which can effectively extract information from the limited labeled signals. However, these SSL methods suffer from low SNR, model instability, or high complexity.

Motivated by this, we introduce the famous MT SSL model as our main framework. And we propose a HTF-mask augmentation method to enhance the model stability. Furthermore, an attention module named PSA is designed to improve the performance of model even under low SNR.

2.2. Data Augmentation Methods. There are serveral traditional augmentation methods applied in AMR tasks. O'Shea et al. [18] generate signals by adding different Gaussian noise to I/Q channel. Liu et al. [33] augmented unlabeled signal by rotating the given IQ signals with an angle randomly selected from $\{0, \pi/2, \pi, 3\pi/2\}$. Furthermore, Luo et al. [31] designed a augmentation method by exchanging the two channels of IQ signals. These methods act on signal directly. However, in scenarios with low SNR, these methods may resulting in the instability of model.

To further enhance the quality of augmented signals, a couple of augmentation methods in computer vision are being searched. Zhang et al. [37] proposed augmentation methods Mixup to extend the training distribution via linear feature interpolation which address the low performance achieved when evaluating adversarial examples. However, it has the problem of unnaturally mixing samples. Yun et al. [38] presented another method named Cutmix by randomly cutting a sample patch and pasting it in the corresponding position of another sample. It enhances the generalization ability of the constructed model but suffers from a fixed square mask shape. Harris et al. [39] designed Fmix which mixes two different samples with a random binary

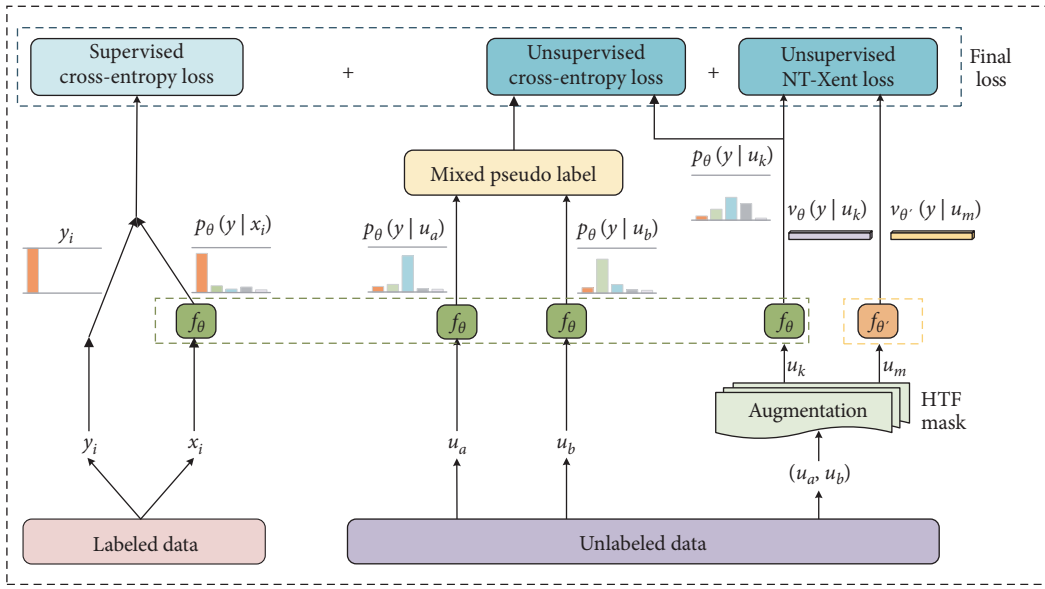

FIGURE 1: Overview of HTF-PSA-SSL.

mask obtained by applying a threshold to low-frequency images sampled from the Fourier space. However, the irregular shape of mask may generate negative samples and affect the network performance when applying to the STFT spectrograms.

Motivated by this, we propose a data augmentation method named the HTF mask. It augment the STFT spectrogram in both temporal and frequency domain to enlarge the amount of unlabel data thus enhance the stability of network.

2.3. Attention Mechanism Methods. Attention mechanism can tell a model where to focus, it also enhances the representation of features. Hu et al. [40] proposed a squeeze-and-excitation (SE) module to efficiently build interdependencies between channels. Qin et al. [41] presented a multispectral channel attention framework (Fca) to assigns varying weights to different channels by producing different frequency components of the discrete cosine transform for each channel. However, these methods only focus on channel attention and lack spatial attention. Woo et al. [42] designed a convolutional block attention mechanism (CBAM), which takes into account both channel and spatial attention. Linsley et al. [43] proposed the global-and-local attention (GALA) module that integrates local saliency and global contextual signals to guide attention toward image regions. However, these methods cannot efficiently capture the strip-shaped features of STFT spectrogram.

Motivated by this, we designed an attention mechanism named PSA. It performs two 1D pooling operations along the horizontal and vertical axes of the feature map to enhance the strip-shaped features of signal.

3. Signal Model

We consider a single-input single-output communication system, and the received signal $r(t)$ can be represented by

$$r(t) = c \times s(t) + n(t), \tag{1}$$

where $s(t)$ denotes the modulated signal from the transmitter, c denotes the path loss or gain term on the signal, and $n(t)$ refers to additive white Gaussian noise (AWGN). Then, the received signal $r(t)$ is sampled N times to obtain a complex-valued discrete-time signal $r(n)$ with a length of N.

The discrete Fourier transform (DFT) can only reflect the properties of the signal in the frequency domain and cannot analyze the signal in the time domain. Therefore, we transform the 1D signal into 2D time–frequency spectrograms using the STFT to associate it with the time domain. The calculation formula for a given discrete signal $r(n)$ can be denoted as follows:

$$\mathrm{STFT}\{r[n]\} = R(m, \omega) = \sum_{n=0}^{N} r[n] w[n-m] e^{-j\omega n}, \tag{2}$$

where $w(n)$ is the window function, and the size of the Hanning window is set to $N/8$.

4. Proposed Approach

4.1. Overview of HTF-PSA-SSL. HTF-PSA-SSL aims at constructing an SSL framework to effectively classify the radio signal modulation types. Figure 1 provides an overview of our proposed HTF-PSA-SSL. The training set can be divided into two parts: a small set of labeled data (x_i, y_i), a large set of unlabeled data u_i, and consisting of STFT domain data. To introduce model perturbations, MT models are built in our framework, where the student model f_θ is trained by the loss function and the teacher model $f_{\theta'}$ is an exponential moving average (EMA) of f_θ. For the small amount of labeled STFT spectrogram, we use the supervised cross-entropy loss L_{ce} over labeled samples (x_i, y_i) to constrain the learning

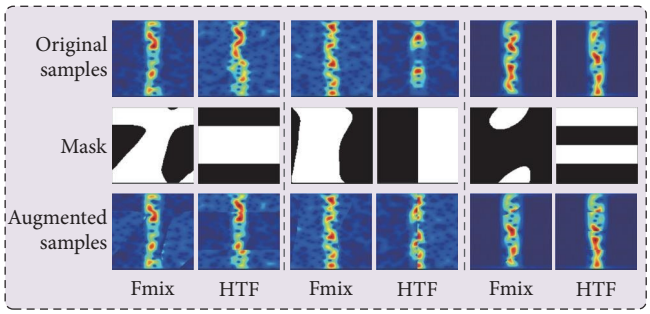

Figure 2: Augmented examples generated by Fmix and HTF mask. Every two columns is a group, there are three groups of augmented examples generated by Fmix and HTF mask, respectively. For each group, the first row is the original two samples. The second row is the binary mask. From left to right is the Fmix mask and HTF mask, respectively. And the third row is the augmented sample generated by Fmix and HTF mask from left to right.

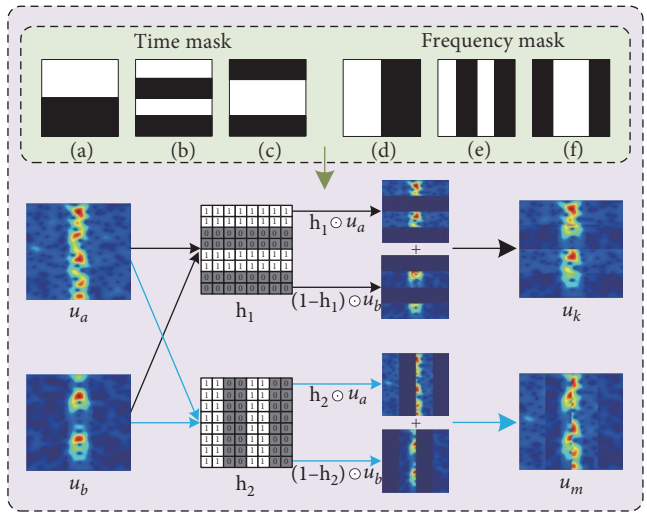

Figure 3: Overview of the HTF mask augmentation policy. The new sample pair (u_k, u_m) is augmented from original unlabeled sample pair (u_a, u_b). To generate new sample u_k (see the black line), applying a Haar mask $\mathbf{h_1}$ to u_a, inverted mask $\mathbf{1} - \mathbf{h_1}$ to u_b, and summation of them. To generate new sample u_m (see the blue line), applying another Haar mask $\mathbf{h_2}$ to u_a, inverted mask $\mathbf{1} - \mathbf{h_2}$ to u_b, and summation of them.

direction of the network gradient descent. For the large amount of unlabeled STFT spectrogram, we apply the data augmentation of HTF Mask on u_i, especially we use sample pair (u_a, u_b) to obtain augmented data pair (u_k, u_m) (described in 4.2). Consequently, for the sample pair (u_a, u_b) and its augmented data u_k, we introduce a pseudo label [44] to label the sample pair (u_a, u_b) (described in 4.4.2) and design the unsupervised cross-entropy loss L_{uce} base on entropy regularization to correct the learning direction of f_θ with the confidence. For augmented data pair (u_k, u_m), we design the unsupervised normalized temperature-scaled cross entropy (NT-Xent) [45] loss L_{ntx} for consistent regularization. The whole training process of HTF-PSA-SSL can be summarized in Algorithm 1.

4.2. Augmentation Policy. Considering that insufficient training data potentially leading to inadequate network training, selecting an effective data augmentation policy is of utmost importance. As shown in Figure 2, Fmix [39] augments data with irregular masks, but these irregular masks are not suitable for physically meaningful STFT spectrogram. Inspired by this, we propose a novel radio data augmentation approach based on HTF mask. It is not only specifically designed for dealing with STFT spectrogram, but also for the unlabeled data. Simultaneously, we incorporate a pseudo label [44] into the augmented unlabeled data. More details of pseudo label is discussed in 4.4.2.

We aim to construct an augmentation policy that acts on the STFT spectrogram directly, which helps the network learn useful features. Motivated by the goal that these features should be robust to deformations in the time direction, deformations in the frequency information, and partial replacement of small segments of the radio signal, we have chosen the following deformations to make up a policy:

(1) Time masking is applied so that t consecutive time frames $[t_0, t_0 + t)$ are masked, where t is first chosen from a uniform distribution from 0 to the time bin T, and t_0 is chosen from $[0, \tau - t)$.

(2) Frequency masking is applied so that f consecutive STFT frequency channels $[f_0, f_0 + f)$ are masked, where f is first chosen from a uniform distribution from 0 to the frequency mask parameter F, and f_0 is chosen from $[0, \nu - f)$.

According to the above policy, we designed three time masks and three frequency masks based on Haar feature template, seen in Figure 3. For each mixing operation, we randomly select two integer values from a uniform distribution ranging from 1 to 6 and generate a pair of masks based on the selected indices. An example of two augmentations applied to a pair of inputs describes the augmentation process in details. Given a pair of mask $\mathbf{h_1}$ and $\mathbf{h_2}$, we can generate the augmented data pair (u_k, u_m) by the below strategy

$$\begin{cases} u_k = \mathbf{h_1} \odot u_a + (\mathbf{1} - \mathbf{h_1}) \odot u_b \\ u_m = \mathbf{h_2} \odot u_a + (\mathbf{1} - \mathbf{h_2}) \odot u_b \end{cases}, \tag{3}$$

where $\mathbf{h_1}, \mathbf{h_2} \in \{0, 1\}^{T \times F}$ denotes a binary mask indicating regions for dropout and replacement within two STFT spectrogram, $\mathbf{1}$ is a binary mask filled with ones, and $\odot$ is element-wise multiplication. Each HTF mask has a mean value of 0 and different shapes work with the different extraction functions.

4.3. Attention Mechanism. Different from the classic attention mechanism in the CV field, PSA adapts itself to the signal strip shape of the STFT spectrogram and helps the network focus more on the important signal features.

Since a convolutional layer has difficulty capturing this strip-shaped relationship due to its limited receptive field, PSA performs two 1D pooling operations along the horizontal and vertical axes of the feature map. Furthermore, the

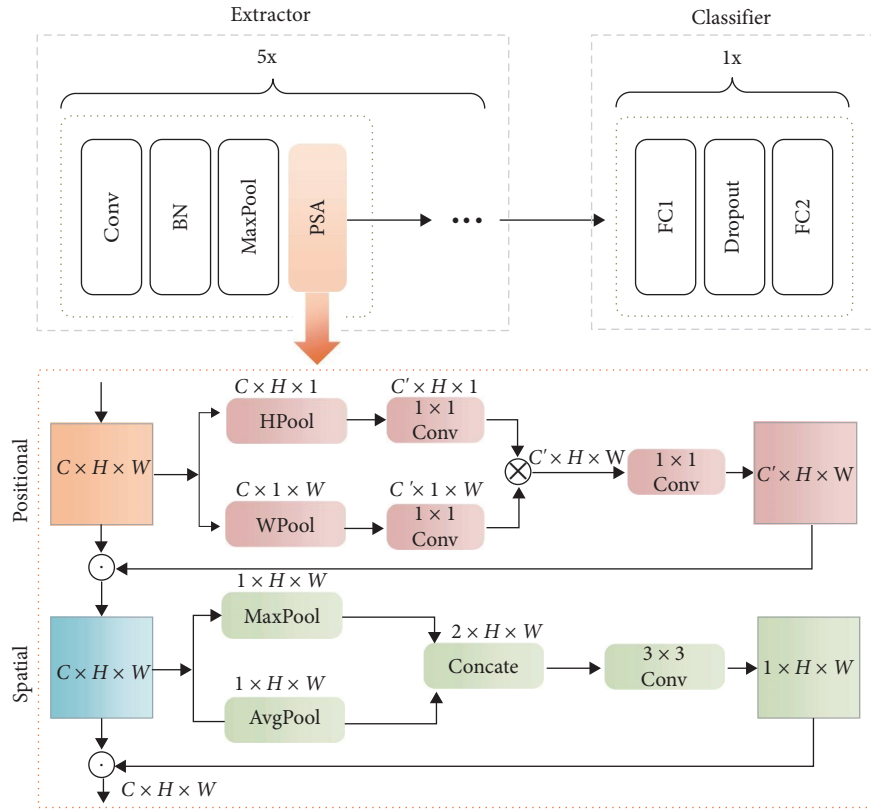

FIGURE 4: Specific structure of proposed attention mechanism PSA. Our network consists of extractor and classifier. Extractor is a stack of five convolutional layers. PSA can adapt itself to the signal strip shape of the STFT spectrogram. PSA performs positional step and spatial step in sequence to address the limited receptive field of convolutional layer.

network should be sensitive to the local position of this strip shape. Then, PSA performs a pooling operation along the channel axis of the feature map.

PSA is added after each max pooling layer. As illustrated in Figure 4, given the upper feature map $F \in \mathbb{R}^{C\times H\times W}$ as input, PSA sequentially generates attention maps $\mathbf{A_p} \in \mathbb{R}^{C\times H\times W}$ and $\mathbf{A_s} \in \mathbb{R}^{1\times H\times W}$. The overall process can be summarized as follows:

$$\begin{aligned} F_p &= A_p(F) \odot F \\ F_s &= A_s(F_p) \odot F_p, \end{aligned} \tag{4}$$

where $\odot$ refers to the element-wise multiplication operation. During multiplication, the values generated in the spatial step are broadcasted along the channel dimension.

4.3.1. Positional Step. Two 1D average pooling layers are applied to aggregate the feature relationships along the horizontal and vertical axes. Then, we utilize a 2D convolution layer with a kernel size of 1×1 to reduce the feature channel; thus, the computational resource required is appropriately lowered. After that, we calculate the correlation matrix between these axes. Finally, another 2D convolution with a kernel size of 1×1 is employed to reconstruct the attention map and multiply it with the input feature, eventually obtaining the final attention map. The above operations can be summarized as follows:

$$A_p(F) = \sigma(\mathrm{Conv}(O_H(F) \otimes O_W(F))), \tag{5}$$

with

$$\begin{aligned} O_H(F) &= \sigma\left(\mathrm{Conv}\left(F^w_{\mathrm{avg}}\right)\right) \\ O_W(F) &= \sigma\left(\mathrm{Conv}\left(F^h_{\mathrm{avg}}\right)\right), \end{aligned} \tag{6}$$

where $\otimes$ denotes matrix multiplication, σ denotes the sigmoid function, and Conv refers to the 2D convolutional layer.

4.3.2. Spatial Step. We first perform average pooling and max pooling operations along the channel axis and then concatenate the results to generate an efficient feature descriptor. Subsequently, we utilize the convolution layer with a kernel size of 1×1 to generate the final spatial attention map. Finally, we multiply this map with the input feature. The specific calculation process can be defined as follows:

$$\begin{aligned} A_s(F) &= \sigma(\mathrm{Conv}([O_A(F); O_M(F)])) \\ &= \sigma\left(\mathrm{Conv}\left(\left[F^s_{\mathrm{avg}}; F^{\mathrm{maxs}}\right]\right)\right), \end{aligned} \tag{7}$$

where $[\cdot;\cdot]$ denotes the concatenation operation.

4.4. Training Loss. To specifically illustrate the training procedure of HTF-PSA-SSL, we first clarify some notations. Let

$\mathscr{D}_L$ and $\mathscr{D}_U$ represent the labeled sample set and unlabeled sample set, respectively, and B_L and B_u are the batch sizes of the labeled sample and unlabeled sample during the training process. Moreover, the total number of signal classes to be classified is referred to as N_c.

4.4.1. Supervised Cross-Entropy Loss. To efficiently guide the learning direction of the whole network, the cross-entropy loss L_{ce} is introduced to calculate the total loss of the labeled data. We feed a labeled sample $(x_i, y_i) \sim \mathscr{D}_L$ into the student network f_θ. Specifically, (x_i, y_i) is first fed into the extractor to collect abundant useful features, and then these useful features are fed into the classifier to generate an output prediction, which is denoted as $p_\theta(y|x_i)$. After that, we use y_i and $p_\theta(y|x_i)$ to calculate the supervised loss L_{ce} as follows:

$$\begin{aligned} L_{ce} &= \frac{1}{B_L}\sum_{i=1}^{B_L} \ell_{ce}(y_i, p_\theta(y|x_i)) \\ &= -\frac{1}{B_L}\sum_{i=1}^{B_L}\sum_{j=1}^{N_c} y_i^j \log(p_\theta(y|x_i)^j). \end{aligned} \tag{8}$$

4.4.2. Unsupervised Cross-Entropy Loss. Although the unlabeled sample set is much larger than the labeled sample set in terms of quantity, we also attempt to apply the cross-entropy loss to the unlabeled sample set. Therefore, for an unlabeled sample $u_a \sim \mathscr{D}_U$, we need to construct a fake label, namely, a pseudo label, and suppose that it is the true label of u_a. More specifically, we feed u_a into the extractor and classifier in sequence and obtain its corresponding predicted output vector $p_\theta(y|u_a)$. To construct a pseudo label, we assume that the predicted vector $p_\theta(y|u_a)$ is credible.

After constructing pseudo labels for all unlabeled data, we sample $(u_a, u_b) \sim \mathscr{D}_U$ and apply the HTF mask to mix (u_a, u_b) into a pair of new samples (u_k, u_m) according to Equation (3). Then, we fetch the pre-preserved pseudo label $(p_\theta(y|u_a), p_\theta(y|u_b))$ according to the sample index (a, b) and generate the mixed pseudo label $\widehat{y}_{\text{mix}}$ as follows:

$$\widehat{y}_{\text{mix}} = \frac{1}{2}p_\theta(y|u_a) + \frac{1}{2}p_\theta(y|u_b). \tag{9}$$

The mixed sample u_k is then fed into the student network to generate the predicted vector $p_\theta(y|u_k)$. The prediction $p_\theta(y|u_k)$ is utilized to calculate the total loss of the unlabeled data. Since u_k is an augmented data sample from (u_a, u_b), when calculating the final loss, we introduce the mixed cross-entropy loss function. The mixed cross-entropy loss function can be denoted as follows:

$$\begin{aligned} L_{uce} &= L_{ce}(\widehat{y}_{\text{mix}}, p_\theta(y|u_k)) \\ &= \frac{1}{B_U}\sum_{i=1,k=1}^{B_U} \ell_{ce}((\widehat{y}_{\text{mix}})_i, p_\theta(y|u_k)) \\ &= -\frac{1}{B_U}\sum_{i=1,k=1}^{B_U}\sum_{j=1}^{N_c} (\widehat{y}_{\text{mix}})_i^j \log(p_\theta(y|u_k)^j). \end{aligned} \tag{10}$$

Require: f_θ: student model with trainable parameters θ
Require: $f_{\theta'}$: teacher model with parameters θ' equal to moving average of θ
Require: $\mathscr{D}_L(x, y)$: labeled samples set
Require: $\mathscr{D}_U(u)$: unlabeled samples set
Require: η: learning rate of student model
Require: α: rate of moving average
Require: λ_u: weight of unlabeled loss
Require: $\omega(t)$: Gaussian ramp–up curve function
Require: B_L: batch size of labeled data
Require: B_U: batch size of unlabeled data
Require: $\text{mix}(u_a, u_b, \mathbf{h}) = \mathbf{h} \odot u_a + (\mathbf{1} - \mathbf{h}) \odot u_b$.
1: **for** t = 1, 2, 3, … **do**
2: Sample $\{(x_i, y_i)\}_{i=1}^{B_L} \sim \mathscr{D}_L(x, y)$
3: Calculate L_{ce} via Equation (8).
4: Sample $\{u_a\}_{a=1}^{B_U} \sim \mathscr{D}_U(u)$, $\{u_b\}_{b=1}^{B_U} \sim \mathscr{D}_U(u)$
5: Generate pseudo label $\{p_\theta(y|u_a)\}_{a=1}^{B_U}$, $\{p_\theta(y|u_b)\}_{b=1}^{B_U}$
6: $u_k = \text{mix}(u_a, u_b, \mathbf{h}_1)$
7: Calculate L_{uce} via Equation (10).
8: $u_m = \text{mix}(u_a, u_b, \mathbf{h}_2)$
9: Calculate L_{ntx} via Equation (14).
10: $\mathscr{L} = L_{ce} + \omega(t) \times \lambda_u \times (L_{uce} + L_{ntx})$
11: $g_\theta \leftarrow \nabla_\theta \mathscr{L}$
12: $\theta'_t = \alpha\theta'_{t-1} + (1 - \alpha)\theta_t$
13: $\theta \leftarrow \theta - \eta g_\theta$
14: **end for**
15: **return** θ, θ'

Algorithm 1: The Training Procedure of HTF-PSA-SSL.

4.4.3. Unsupervised NT-Xent Loss. When a percept is slightly changed, the smaller the angles between these high-dimensional features, the closer these classes are. Motivated by this, we apply the contrastive loss to maximize the agreement between different examples augmented from the same signal sample. Specifically, we obtain the latent high-dimensional feature representation $(v_\theta(y|u_k), v_{\theta'}(y|u_m))$ generated from the extractor and calculate the NT-Xent loss between them. To minimize NT-Xent, we use cosine similarity to measure the similarity between two augmented samples u_k and u_m. The cosine similarity measure is defined as follows:

$$\text{sim}(v_\theta(y|u_k), v_{\theta'}(y|u_m)) = \frac{v_\theta(y|u_k)^T v_{\theta'}(y|u_m)}{\|v_\theta(y|u_k)\| \|v_{\theta'}(y|u_m)\|}, \tag{11}$$

where $\|v_\theta(y|u_k)\|$ denotes the ℓ_2 norm of $v_\theta(y|u_k)$. Then, the loss for a positive pair of samples (k, m) is defined as follows:

$$\ell(k, m) = -\log \frac{\exp(\text{sim}(v_\theta(y|u_k), v_{\theta'}(y|u_m))/\tau)}{\text{Neg}}, \tag{12}$$

with

$$\mathrm{Neg} = \sum_{j=1}^{2B_U} \mathbb{1}_{[j\neq k]} \exp\left(\mathrm{sim}\left(v_\theta(y|u_k), v_{\theta'}(y|u_j)\right)/\tau\right), \quad (13)$$

where τ represents the temperature when calculating the cosine similarity value and $\mathbb{1}_{[j\neq k]} \in \{0, 1\}$ is an indicator function that is equal to 1 iff $j \neq k$. To obtain the final loss, the average loss values of all positive sample pairs, including (m, k), are calculated, which can be denoted as follows:

$$L_{ntx} = \frac{1}{2B_U} \sum_{i=1}^{B_U} [\ell(2i-1, 2i) + \ell(2i, 2i-1)]. \quad (14)$$

4.4.4. Final Loss. The final total loss function of HTF-PSA-SSL can be defined as follows:

$$\mathscr{L} = L_{ce} + \omega(t) \times \lambda_u \times (L_{uce} + L_{ntx}). \quad (15)$$

where the weight λ_u is a hyperparameter that balances the labeled loss and unlabeled loss. The $\omega(t)$ function is the Gaussian ramp-up curve function [46], which can be defined as follows:

$$\omega(t) = \begin{cases} \exp^{-5\left(1-\frac{t}{T}\right)}, & 0 \leq t < T \\ 1, & T \leq t \end{cases}. \quad (16)$$

where t is the number of current epochs and T refers to the starting epoch when the unlabeled weight is equal to λ_u. The application of $\omega(t)$ ensures that the training process with the labeled data is not disturbed even with the existence of an unlabeled loss. Moreover, its slow increase helps the pseudo labels of unlabeled data become closer to the true labels.

The whole training process of HTF-PSA-SSL can be summarized in Algorithm 1.

5. Experiments

We evaluate the effectiveness of the proposed HTF-PSA-SSL approach and compare it with the other existing SSL-based methods on the public RML2016.10a dataset. To further demonstrate the robustness of the proposed HTF-PSA-SSL method, we also examine its performance on the other public datasets, RML2016.10b and RML2016.04c.

5.1. Datasets and Training Environment

5.1.1. Dataset Descriptions. RML2016.10a is a synthetic dataset consisting of 11 modulation types (8 digital and 3 analog), which are 8PSK, AM-DSB, AM-SSB, BPSK, CPFSK, GFSK, PAM4, QAM16, QAM64, QPSK, and WBFM. For each modulation type, there are 20 different SNRs varying from −20 to +18 dB with an interval of 2 dB. For each SNR, there are 1,000 signals. Each signal is composed of I and Q parts, and the size is 2×128. In this paper, for each raw IQ signal, we apply the STFT function to generate a spectrogram with a size of 128×128 and feed it into the network as our input.

Table 1: Comparisons between supervised (100%), HTF-PSA-SSL and supervised (1%).

Type	Supervised (100%)	HTF-PSA-SSL (%)	Supervised (1%)
8PSK	98.00	97.00	86.00
AM-DSB	80.00	78.00	66.00
AM-SSB	99.00	100.00	34.00
BPSK	99.00	99.00	97.00
CPFSK	100.00	100.00	100.00
GFSK	100.00	100.00	100.00
PAM4	100.00	100.00	100.00
QAM16	100.00	100.00	45.00
QAM64	99.00	99.00	65.00
QPSK	98.00	98.00	97.00
WBFM	50.00	42.00	47.00
Average	93.00	**92.09**	76.09

Note. Bold value indicates the total average result of HTF-PSA-SSL.

5.1.2. Implementation Details. We split the dataset into three parts, training, validation, and testing sets, according to the ratio of 8 : 1 : 1. Then, we select a certain percentage of samples from the training set as the labeled dataset, while we select the remaining samples of the training set as the unlabeled dataset. Specifically, we randomly divide 1,000 signals into three groups: 800 signals for training, 100 signals for validation, and 100 signals for testing. Then, the training set is divided into two parts: 88 signals for the labeled dataset and 712 signals for the unlabeled dataset. The other dataset is divided in the same way.

The adaptive moment estimation (Adam) optimizer is applied for all of our experiments, and each experiment runs for 150 epochs. The initial learning rate is set to 0.001, which is then adjusted with cosine decay. The moving average rate α of the EMA is set to 0.99. The batch size is set to 64 for both the labeled dataset and unlabeled dataset. The weight of the unlabeled loss is set to 20, and the number of epochs for the ramp-up function is set to 30. Our experiments are implemented in PyTorch with Python 3.7 using two Nvidia 3090 graphics processors.

5.2. Comparison with Supervised Methods. We consider two supervised scenarios: supervised (100%) and supervised (1%). Supervised (100%) means that we train a network using the full training set with labels. Supervised (1%) refers to training the network using only 1% of the whole training set. The comparison results obtained under 6 dB signals are shown in Table 1. We can determine that the performance of HTF-PSA-SSL is basically between those of the two

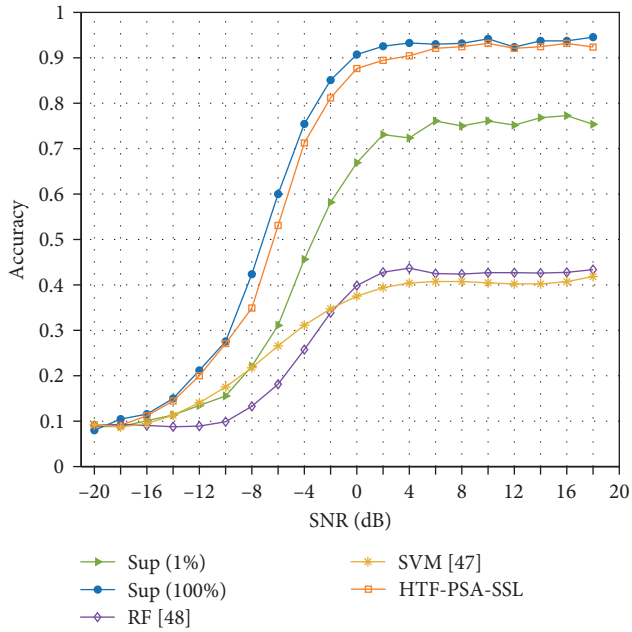

Figure 5: Comparison with Sup (1%), Sup (100%), SVM [47], and RF [48].

supervised methods, which is 0.91% lower than supervised (100%) but 16% higher than supervised (1%). This shows that HTF-PSA-SSL can effectively make use of unlabeled data to improve its recognition performance.

Then, we evaluate the performance of the three methods from −20 to +18 dB, and the comparison results are shown in Figure 5. It is obvious that HTF-PSA-SSL keeps approaching the supervised (100%) method and outperforms the supervised (1%) method by 13.88% on average. This further verifies the powerful feature extraction ability of HTF-PSA-SSL, which can extract sufficient reliable features from an unlabeled dataset and continuously improve its model performance. We also compare the performance of HTF-PSA-SSL with some Machine Learning (ML) methods. These ML methods are trained using the full training set with labels. As shown in Figure 5, at a SNR of 16 dB, HTF-PSA-SSL outperforms support vector machine (SVM) [47] by 52% and random forest (RF) [48] by 50% in terms of recognition accuracy. Moreover, we show the feature visualization of instantaneous statistical features [47], entropy features [48], and high-dimensional features (HTF-PSA-SSL) using t-distributed stochastic neighbor embedding (t-SNE) [49]. The feature distribution under 12 dB signals is shown in Figure 6. It is obvious that the features of HTF-PSA-SSL are well-aggregated, while the instantaneous statistical features and entropy features are scattered. Since these manual features are more severely confused than the high-dimensional features, the performance obtained by SVM and RF is also lower than that of HTF-PSA-SSL. However, HTF-PSA-SSL has a higher computation complexity. The comparison of computational complexity is presented in Table 2.

From the confusion matrix obtained under −6, 0, and 12 dB signals, drawn in Figure 7, we observe that with increasing SNR, the recognition accuracies achieved for most modulation types are improved. However, AM-DSB and WBFM are heavily confused even at 12 dB, which indicates that this pair of modulation classes is difficult to correctly recognize.

5.3. Efficiency of the HTF Mask and PSA. As shown in Figure 8, the HTF mask augmentation method achieves the best classification accuracy compared with the Mixup method [37] and the Fmix method [39] from −20 to +18 dB. More specifically, the HTF mask augmentation method has a higher accuracy than Mixup method due to the application of mask augmentation form and it has a better performance than Fmix method because HTF mask method has taken the temporal and frequency correspondence of STFT spectrogram into account. For instance, the HTF mask method outperforms Mixup by nearly 12% and surpasses Fmix by almost 4% on the average of −20 to 18 dB. Under 16 dB, HTF mask achieves a highest accuracy of 93.18%.

Then, we simply replace PSA with the other famous attention methods, SE [40], the CBAM [42], and Fca [41], and evaluate their performance under the same experimental settings. The experimental results are shown in Figure 9. The recognition accuracy of PSA is higher than that of the other three attention mechanisms by 2.67%, 5.13%, and 7.75% on average, especially when the SNR ≥ -6 dB. This is due to the strong detail extraction ability of PSA, which establishes the horizontal and vertical long-term dependencies of features to effectively capture the detailed information contained in signals.

5.4. Ablation Study. To showcase the efficiency of the proposed HTF mask and PSA method, we perform a range of ablation experiments at 10 different SNRs. Furthermore, we evaluate the performance of training loss, the corresponding results are all listed in Table 3.

5.4.1. HTF Mask. As shown in Table 3, augmentation method with only frequency mask or time mask reach higher classification accuracy compared to no augmentation method Eps13 [44]. Furthermore, by augmenting data in both the temporal and frequency domains, HTF-PSA-SSL achieves superior results, as demonstrated in Table 3. For example, HTF-PSA-SSL outperforms Eps13 by 22.31%, Eps13 with frequency mask by 16.18%, and Eps13 with time mask by 10.63% under −4 dB. These results show the effectiveness of the HTF mask in augmenting spectrogram data, such as the STFT spectrogram.

5.4.2. PSA Method. From Table 3, we can figure out that model with spatial attention or positional attention achieve higher classification accuracy than no attention model CNN5. However, when both the positional attention and the spatial attention are performed, HTF-PSA-SSL reaches the highest accuracy. For instance, HTF-PSA-SSL outperforms CNN5 by 13.68%, CNN5 with spatial attention by 6.28%, and CNN5 with positional attention by 3.73%, respectively, under 0 dB. These experiments indicate that the PSA can efficiently enhance the strip-shaped features of STFT spectrogram.

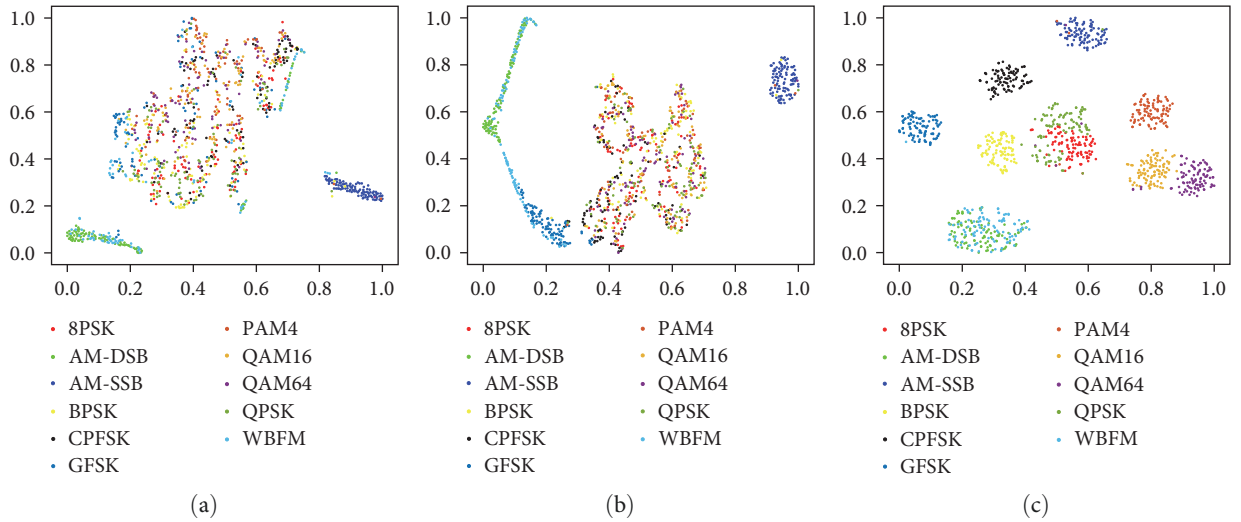

Figure 6: Sample distribution of (a) instantaneous statistical features[47], (b) entropy features [48], and (c) high-dimensional features (HTF-PSA-SSL) after using t-SNE to visualize features under 12 dB signals. The high-dimensional features of HTF-PSA-SSL are well-aggregated, while the instantaneous statistical features and entropy features are heavily scattered.

Table 2: Computation complexity.

Network	FLOPs	Parameters	Memory
SVM [47]	—	54.2 K	1.5 M
RF [48]	—	63.7 K	3.5 M
SSRCNN [19]	390 K	52.8 K	227.7 K
EDCT [31]	9 M	291 K	1.2 M
SimAMC [33]	25 M	620 K	2.5 M
CNN5	2.5 G	1.66 M	6.7 M
CNN5 + spatial	2.5 G	1.66 M	6.7 M
SE [40]	2.5 G	1.77 M	7.1 M
Fca [41]	2.5 G	**1.85 M**	**11.2 M**
CBAM [42]	2.5 G	1.77 M	7.1 M
CNN5 + positional	2.53 G	1.82 M	7.4 M
HTF-PSA-SSL	**2.53 G**	1.82 M	7.4 M

Note. Bold values indicate the highest value of FLOPs, Paramters and Memory.

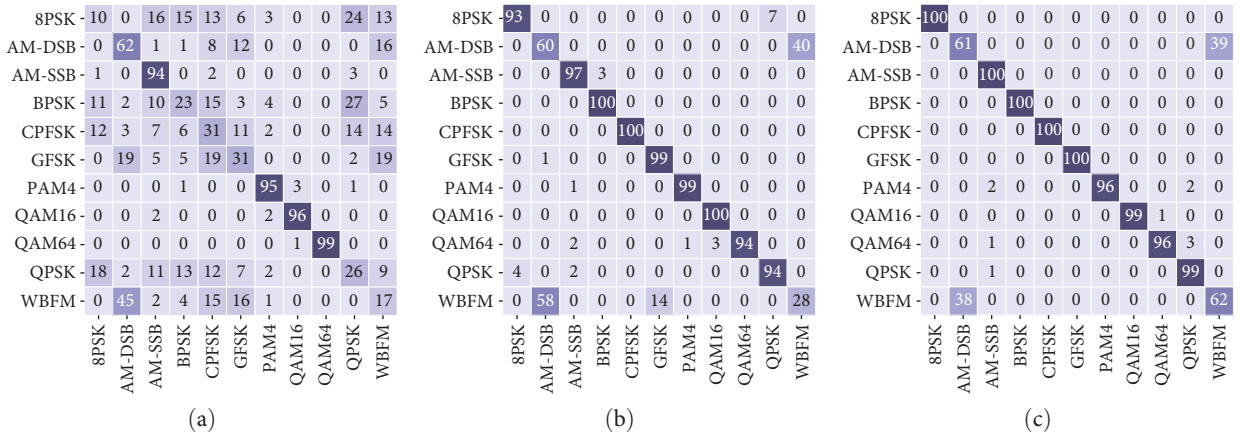

Figure 7: Confusion matrix of HTF-PSA-SSL under (a) −6, (b) 0, and (c) 12 dB signal. The horizontal axis is the predicted modulation type of network. The vertical axis is the true modulation type of signals. Each number on the leading diagonal shows the total number of modulation correct prediction.

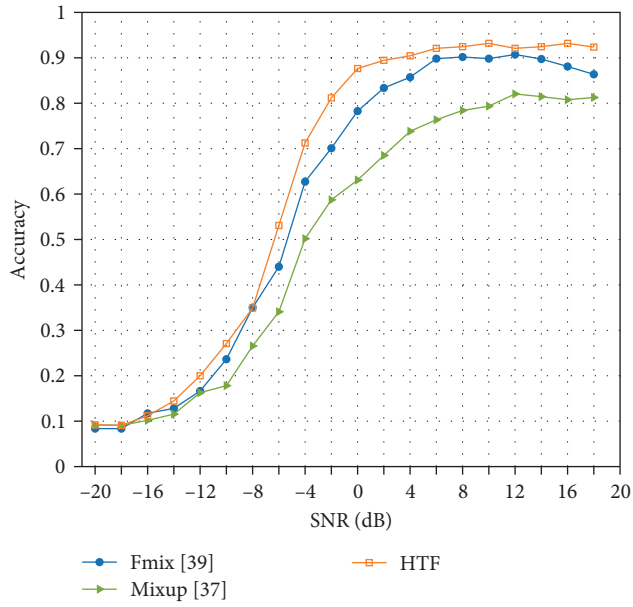

Figure 8: Recognition accuracy of different augmentation methods.

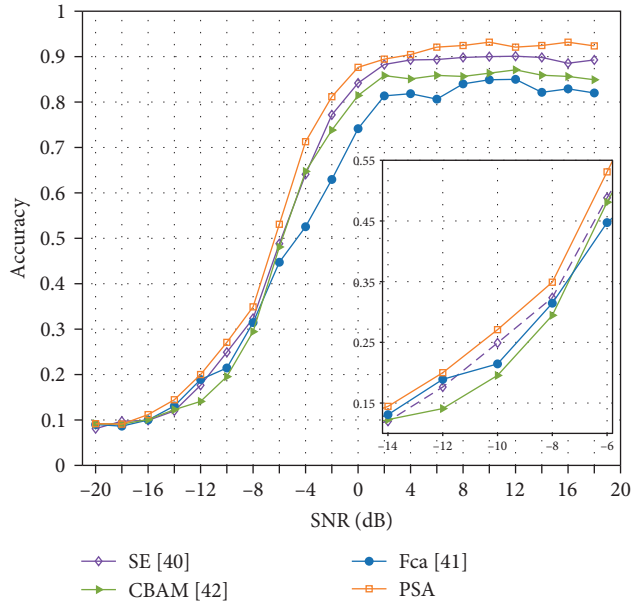

Figure 9: Recognition accuracy of different attention mechanisms.

5.4.3. Traing Loss. As shown in Table 3, the absence of unsupervised cross-entropy loss L_{uce} leads to the significant decline in classification accuracy. This indicates that the L_{uce} plays an important role when training network. Moreover, when both the unsupervised cross-entropy loss L_{uce} and the unsupervised NT-Xent loss L_{ntx} are added, HTF-PSA-SSL achieves the best classification accuracy. For example, under 4 dB, HTF-PSA-SSL outperforms supervised cross-entropy loss L_{ce} by 18.09%, $L_{ce} + L_{uce}$ by 1.18%, respectively. This further shows that the loss function of HTF-PSA-SSL is indeed useful.

5.5. Comparison with Other SSL-Based Methods. In this experiment, we evaluate the performance of three SSL methods applied in the signal field: SSRCNN [19], SimAMC [33], and EDCT [31]. As shown in Figure 10, HTF-PSA-SSL reaches higher recognition accuracy than the other three SSL-based methods, it outperforms SSR by 35.84%, SimAMC by 35.28%, and EDCT by 13.01%. This fully demonstrates the strong performance of the proposed HTF-PSA-SSL technique. It can extract more critical information from spectrograms and screen out the practical features from this information.

Table 3: Ablation study on public dataset RML2016.10a.

Description	−20 dB (%)	−16 dB (%)	−12 dB (%)	−8 dB (%)	−4 dB (%)	0 dB (%)	4 dB (%)	8 dB (%)	12 dB (%)	16 dB (%)
L_{ce}	8.82	10.09	13.45	22.09	45.64	66.91	72.36	75.00	75.18	77.27
$L_{ce} + L_{uce}$	8.82	10.45	17.82	32.91	70.91	87.36	89.27	91.96	92.00	92.64
CNN5	7.91	9.55	14.09	29.00	48.64	73.96	80.37	82.87	83.27	82.96
CNN5 + spatial	8.09	10.36	15.00	29.60	60.64	81.36	83.73	84.32	85.96	85.27
CNN5 + positional	9.09	11.00	18.45	33.05	69.36	83.91	88.82	89.73	89.55	89.03
Eps13 [44]	8.73	9.18	15.27	25.73	48.96	68.18	72.46	77.68	81.55	81.36
Eps13 + frequency mask	9.00	10.64	15.91	29.77	55.09	75.00	81.82	82.33	84.82	83.45
Eps13 + time mask	8.27	10.46	17.64	32.82	60.64	76.32	84.48	84.82	85.03	83.36
HTF-PSA-SSL	**9.18**	**11.18**	**20.00**	**34.91**	**71.27**	**87.64**	**90.45**	**92.45**	**92.09**	**93.18**

Note. Bold values indicate the result of HTF-PSA-SSL.

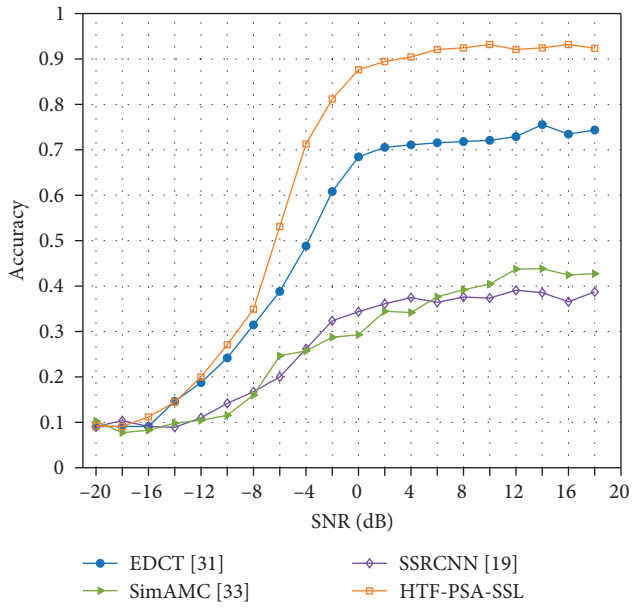

Figure 10: Comparison with other SSL-based signal recognition methods.

5.6. Robustness of the Proposed HTF-PSA-SSL Method. In this part, we study the robustness of HTF-PSA-SSL by evaluating its recognition accuracy on three public datasets, RML2016.10a, RML2016.10b, and RML2016.04c, and the specific recognition accuracies are shown in Table 4. RML2016.10b contains 10 modulation classes, but the total size of the dataset is much larger than that of RML2016.10a. For each SNR, each modulation type has 6,000 signal samples. RML2016.04c has the same modulation classes as RML2016.10a, but the number of samples for each modulation type is different, ranging from 207 to 1,248. For each SNR, there are 8,103 signal samples in total, including all modulation classes. We can determine that HTF-PSA-SSL achieves the best recognition accuracy on these three public datasets, which indicates that HTF-PSA-SSL is robust and can achieve stable recognition performance, regardless of whether the given dataset is large or small and whether the numbers of samples in different classes are balanced or not.

Table 4: The recognition accuracy of HTF-PSA-SSL on public datasets RML2016.10a, RML2016.10b and RML2016.04c with 1% (88) labeled samples.

SNR (dB)	RML2016.10a (%)	RML2016.10b (%)	RML2016.04c (%)
−20	9.18	10.27	8.91
−18	9.09	11.05	8.68
−16	11.18	14.43	8.87
−14	14.45	18.78	9.09
−12	20.00	23.35	12.10
−10	27.09	30.23	17.41
−8	34.91	40.62	21.11
−6	53.09	51.27	36.54
−4	71.27	71.98	67.41
−2	81.18	83.28	79.14
0	87.64	87.90	86.05
2	89.45	89.58	89.88
4	90.45	91.13	91.23
6	92.09	90.02	90.74
8	92.45	90.62	89.63
10	93.18	91.87	91.23
12	92.09	91.75	91.73
14	92.45	91.72	88.27
16	93.18	92.72	89.38
18	92.36	93.25	91.48

As shown in Table 5, we evaluate the recognition accuracy of the label rate 1% (88), 5% (440), and 10% (880) at some SNRs. The results show that with the increase in the amount of labeled data, the recognition accuracy of HTP-PSA-SSL is also gradually improved, but the improvement fluctuates at 1%. This indicates that increasing the amount of labeled data can improve the recognition accuracy of HTP-PSA-SSL but the improvement effect is not obvious. This is because the HTF mask improves the robustness of the network. Even with only a small amount of labeled data, the recognition accuracy is close to that of supervised learning using the whole training set.

Table 5: The recognition accuracy of HTF-PSA-SSL under different label rate: 1% (88), 5% (440) and 10% (880).

SNR (dB)	1% (88)	5% (440)	10% (880)
−18	9.09%	9.55%	9.73%
−12	20.00%	20.36%	20.73%
−6	53.09%	54.73%	55.36%
0	87.64%	89.82%	89.73%
6	92.09%	92.64%	93.09%
12	92.09%	93.00%	93.55%
18	92.36%	93.82%	94.09%

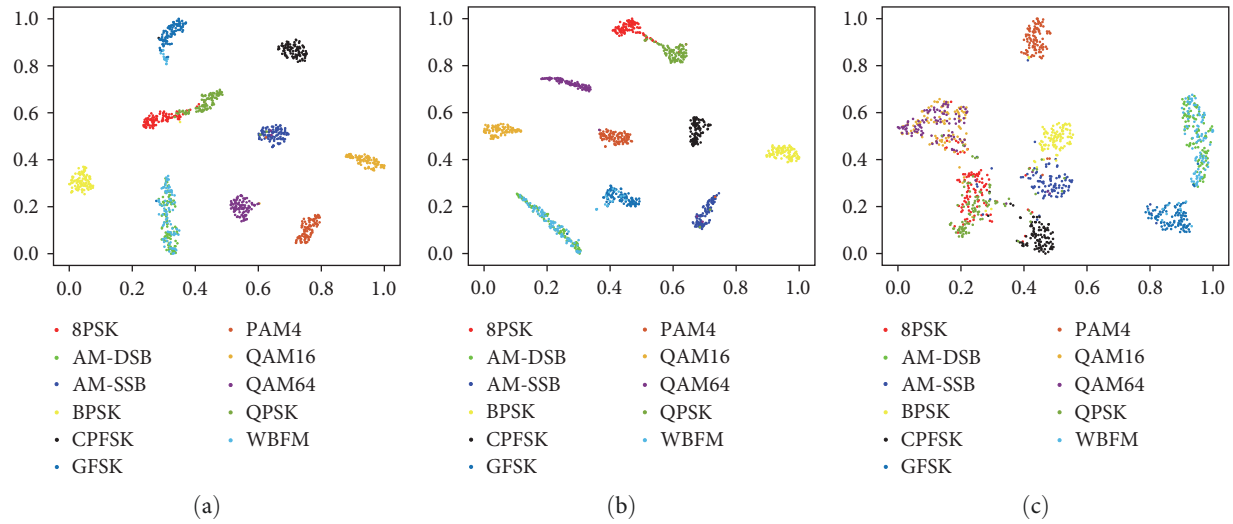

Figure 11: Test sample points of two supervised methods and HTF-PSA-SSL after dimensionality reduction using t-SNE under 0 dB signals. (a) Supervised (100%) and (b) HTF-PSA-SSL are well-aggregated. (c) Supervised (1%) is heavily scattered.

To visualize the features of the test signal samples, we obtain the intermediate features of the classifier and utilize t-SNE for dimensionality reduction. The sample point distribution under 0 dB signals is shown in Figure 11. It is obvious that the results supervised (100%) and HTF-PSA-SSL are well-aggregated, while those of supervised (1%) are scattered and heavily confused with QAM16 and QAM64, QPSK and 8PSK, further indicating that HTF-PSA-SSL is highly reliable. From these figures, we can additionally conclude that WBFM and AM-DSB are difficult to recognize for HTF-PSA-SSL as well as EDCT, SSRCNN, and SimAMC.

5.7. Computation Complexity. In Table 2, the FLOPs, the parameters volume and the memory usage are compared.

5.7.1. FLOPs. Compared with other methods, HTF-SSL-PSA has the highest FLOPs which is 2.53 G. This is because we filter redundant information from the time domain, frequency domain, and global time–frequency domain, which requires a certain amount of calculation. We also compare the time complexity of SVM, RF, and HTF-PSA-SSL. From Table 6, we can figure out that HTF-PSA-SSL has the highest time complexity. In our forthcoming research, strategies to optimize network computational overhead are our research directions, such as binary neural networks.

Table 6: Time complexity of machine learning methods and HTF-PSA-SSL.

Network	Time complexity
SVM [47]	$O(n^3)$
RF [48]	$O(T(n \cdot f \cdot \log(f)))$
HTF-PSA-SSL	$O(\sum_{l=1}^{D} M_l^2 \cdot K_l^2 \cdot C_{l-1} \cdot C_l)$

n is the number of samples, f is the number of features used in each tree, T is the number of trees in the forest. D is number of convolutional layers, M is the side length of the feature map output by each convolution kernel, K is the kernel size of convolutions, and C is the channels of convolutions.

5.7.2. Parameters. In terms of parameters, Fca has the highest value, which is 1.85 M, while HTF-PSA-SSL closely follows as the second highest with 1.82 M. This improvement can be attributed to the PSA's powerful filtering function of key information in both the time and frequency domains. Compared to SE and CBAM, the HTF-PSA-SSL is slightly higher. This increase can be attributed to the fact that HTF-PSA-SSL also filters global time–frequency information. These domains collaboratively eliminate redundant information. While compared to the rest methods, HTF-PSA-SSL is much higher. This difference can be attributed to the model's necessary complexity, which allows HTF-SSL-PSA to extract features from large

amounts of unlabeled data and contribute to correcting the network's learning direction. In our future work, we will continue to investigate methods for significantly reducing the computational complexity of the network while maintaining the higher recognition performance.

5.7.3. Memory. Similar to the model parameters, the memory usage of HTF-PSA-SSL also ranks second at 7.4 M. This is because the number of parameters of the network itself is large. This is also a direction for our future improvement.

6. Conclusion

In this paper, an AMR framework based on SSL is proposed to achieve improved modulation recognition accuracy by effectively utilizing large amounts of unlabeled data. While using only a small amount of labeled data, the framework can significantly improve its recognition performance. The proposed HTF mask data augmentation method can effectively mix unlabeled data and expand the total amount of unlabeled data to improve the overall generalization performance of the convolutional network. The designed attention mechanism named PSA can plug and play into any convolutional layer to compensate for the limited receptive field of the convolutional layer and enhance the feature extraction ability of the convolutional network. Compared with SSR, SimAMC, and EDCT, HTF-PSA-SSL achieves 35.84%, 35.28%, and 13.01% higher accuracy on average. Compared with supervised (1%), HTF-PSA-SSL improves the recognition accuracy by 13.88% on average. Extensive experiments and comparisons conducted on public datasets show that the proposed framework can effectively use a large amount of unlabeled data and accurately predict the modulation types of unknown signals with very little labeled data.

Data Availability

The data used to support this study are public datasets. They can be downloaded from http://radioml.com [14].

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

[1] X. Hong, J. Wang, C.-X. Wang, and J. Shi, "Cognitive radio in 5G: a perspective on energy-spectral efficiency trade-off," *IEEE Communications Magazine*, vol. 52, no. 7, pp. 46–53, 2014.

[2] Z. Zhu and A. K. Nandi, *Automatic Modulation Classification: Principles, Algorithms and Applications*, John Wiley & Sons, 2015.

[3] O. A. Dobre, A. Abdi, Y. Bar-Ness, and W. Su, "Survey of automatic modulation classification techniques: classical approaches and new trends," *IET Communications*, vol. 1, no. 2, pp. 137–156, 2007.

[4] A. Polydoros and K. Kim, "On the detection and classification of quadrature digital modulations in broad-band noise," *IEEE Transactions on Communications*, vol. 38, no. 8, pp. 1199–1211, 1990.

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

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

[7] J. Reichert, "Automatic classification of communication signals using higher order statistics," in *ICASSP-92: 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 221–224, IEEE, San Francisco, CA, USA, 1992.

[8] Y. Yang and S. S. Soliman, "Statistical moments based classifier for MPSK signals," in *IEEE Global Telecommunications Conference GLOBECOM '91: Countdown to the New Millennium. Conference Record*, pp. 72–76, IEEE, Phoenix, AZ, USA, 1991.

[9] B. Schölkopf, K. Tsuda, and J.-P. Vert, "Advanced application of support vector machines," in *Kernel Methods in Computational Biology*, p. 275, MIT Press, 2004.

[10] S. Zheng, P. Qi, S. Chen, and X. Yang, "Fusion methods for CNN-based automatic modulation classification," *IEEE Access*, vol. 7, pp. 66496–66504, 2019.

[11] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: pre-training of deep bidirectional transformers for language understanding," arXiv preprint arXiv: 1810.04805, 2018.

[12] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[13] G. Hinton, L. Deng, D. Yu et al., "Deep neural networks for acoustic modeling in speech recognition: the shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[14] T. J. O'Shea, J. Corgan, and T. C. Clancy, "Convolutional radio modulation recognition networks," in *Engineering Applications of Neural Networks. EANN 2016*, vol. 629 of *Communications in Computer and Information Science*, pp. 213–226, Springer, Cham, 2016.

[15] 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.

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

[17] K. Yashashwi, A. Sethi, and P. Chaporkar, "A learnable distortion correction module for modulation recognition," *IEEE Wireless Communications Letters*, vol. 8, no. 1, pp. 77–80, 2019.

[18] T. J. O'Shea, N. West, M. Vondal, and T. C. Clancy, "Semi-supervised radio signal identification," in *2017 19th International Conference on Advanced Communication Technology (ICACT)*, pp. 33–38, IEEE, PyeongChang, Korea (South), 2017.

[19] Y. Dong, X. Jiang, L. Cheng, and Q. Shi, "SSRCNN: a semi-supervised learning framework for signal recognition," *IEEE Transactions on Cognitive Communications and Networking*, vol. 7, no. 3, pp. 780–789, 2021.

[20] T. J. O'shea and N. West, "Radio machine learning dataset generation with gnu radio," *Proceedings of the GNU Radio Conference*, vol. 1, no. 1, pp. 1–6, 2016.

[21] X. Li, Y. Xu, N. Li, B. Yang, and Y. Lei, "Remaining useful life prediction with partial sensor malfunctions using deep

adversarial networks," *IEEE/CAA Journal of Automatica Sinica*, vol. 10, no. 1, pp. 121–134, 2023.

[22] W. Zhang, Z. Wang, and X. Li, "Blockchain-based decentralized federated transfer learning methodology for collaborative machinery fault diagnosis," *Reliability Engineering & System Safety*, vol. 229, Article ID 108885, 2023.

[23] J. Xu, C. Luo, G. Parr, and Y. Luo, "A spatiotemporal multichannel learning framework for automatic modulation recognition," *IEEE Wireless Communications Letters*, vol. 9, no. 10, pp. 1629–1632, 2020.

[24] L. Li, J. Huang, Q. Cheng, H. Meng, and Z. Han, "Automatic modulation recognition: a few-shot learning method based on the capsule network," *IEEE Wireless Communications Letters*, vol. 10, no. 3, pp. 474–477, 2020.

[25] 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.

[26] 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.

[27] Y. Wang, M. Liu, J. Yang, and G. Gui, "Data-driven deep learning for automatic modulation recognition in cognitive radios," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 4, pp. 4074–4077, 2019.

[28] T. Huynh-The, C.-H. Hua, Q.-V. Pham, and D.-S. Kim, "MCNet: an efficient CNN architecture for robust automatic modulation classification," *IEEE Communications Letters*, vol. 24, no. 4, pp. 811–815, 2020.

[29] Z. Zhang, H. Luo, C. Wang, C. Gan, and Y. Xiang, "Automatic modulation classification using CNN-LSTM based dual-stream structure," *IEEE Transactions on Vehicular Technology*, vol. 69, no. 11, pp. 13521–13531, 2020.

[30] F. Zhang, C. Luo, J. Xu, and Y. Luo, "An efficient deep learning model for automatic modulation recognition based on parameter estimation and transformation," *IEEE Communications Letters*, vol. 25, no. 10, pp. 3287–3290, 2021.

[31] C. Luo, W. Wang, and L. Gan, "Modulation recognition based on deep co-training," in *2021 International Conference on Digital Society and Intelligent Systems (DSInS)*, pp. 287–291, IEEE, 2021.

[32] N. E. West and T. O'shea, "Deep architectures for modulation recognition," in *2017 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pp. 1–6, IEEE, Baltimore, MD, USA, 2017.

[33] D. Liu, P. Wang, T. Wang, and T. Abdelzaher, "Self-contrastive learning based semi-supervised radio modulation classification," in *MILCOM 2021-2021 IEEE Military Communications Conference (MILCOM)*, pp. 777–782, IEEE, 2021.

[34] M. Li, G. Liu, S. Li, and Y. Wu, "Radio classify generative adversarial networks: a semi-supervised method for modulation recognition," in *2018 IEEE 18th International Conference on Communication Technology (ICCT)*, pp. 669–672, IEEE, Chongqing, China, 2018.

[35] M. Li, O. Li, G. Liu, and C. Zhang, "Generative adversarial networks-based semi-supervised automatic modulation recognition for cognitive radio networks," *Sensors*, vol. 18, no. 11, Article ID 3913, 2018.

[36] H. Kim, A. Shahid, J. Fontaine, E. De Poorter, I. Moerman, and H. Nam, "Automatic modulation classification using relation network with denoising autoencoder," in *2022 13th International Conference on Information and Communication Technology Convergence (ICTC)*, pp. 485–488, IEEE, 2022.

[37] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: beyond empirical risk minimization," arXiv preprint arXiv: 1710.09412, 2017.

[38] S. Yun, D. Han, S. J. Oh, S. Chun, J. Choe, and Y. Yoo, "Cutmix: regularization strategy to train strong classifiers with localizable features," in *Proceedings of the IEEE/CVF international conference on computer vision (ICCV)*, pp. 6023–6032, IEEE, 2019.

[39] E. Harris, A. Marcu, M. Painter, M. Niranjan, A. Prügel-Bennett, and J. Hare, "Fmix: enhancing mixed sample data augmentation," arXiv preprint arXiv: 2002.12047, 2020.

[40] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 7132–7141, Salt Lake City, UT, USA, 2018.

[41] Z. Qin, P. Zhang, F. Wu, and X. Li, "Fcanet: frequency channel attention networks," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 783–792, IEEE, 2021.

[42] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: convolutional block attention module," in *Proceedings of the European conference on computer vision (ECCV)*, pp. 3–19, IEEE, 2018.

[43] D. Linsley, D. Shiebler, S. Eberhardt, and T. Serre, "Learning what and where to attend," arXiv preprint arXiv: 1805.08819, 2018.

[44] D.-H. Lee, "Pseudo-label: the simple and efficient semi-supervised learning method for deep neural networks," in *Workshop on Challenges in Representation Learning*, p. 896, ICML, Atlanta, Georgia, USA, 2013.

[45] K. Sohn, "Improved deep metric learning with multi-class N-pair loss objective," in *Advances in Neural Information Processing Systems*, pp. 1857–1865, 2016.

[46] S. Laine and T. Aila, "Temporal ensembling for semi-supervised learning," arXiv preprint arXiv: 1610.02242, 2016.

[47] X. Zhang, T. Ge, and Z. Chen, "Automatic modulation recognition of communication signals based on instantaneous statistical characteristics and SVM classifier," in *2018 IEEE Asia-Pacific Conference on Antennas and Propagation (APCAP)*, pp. 344–346, IEEE, Auckland, New Zealand, 2018.

[48] Z. Zhang, Y. Li, X. Zhu, and Y. Lin, "A method for modulation recognition based on entropy features and random forest," in *2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pp. 243–246, IEEE, Prague, Czech Republic, 2017.

[49] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *Journal of Machine Learning Research*, vol. 9, no. 11, pp. 2579–2605, 2008.