electronics

Article

Specific Emitter Identification Model Based on Improved BYOL Self-Supervised Learning

Dongxing Zhao, Junan Yang *, Hui Liu and Keju Huang

College of Electronic Engineering, National University of Defense Technology, Hefei 230000, China
* Correspondence: yangjunan@ustc.edu

Abstract: Specific emitter identification (SEI) is extracting the features of the received radio signals and determining the emitter individuals that generate the signals. Although deep learning-based methods have been effectively applied for SEI, their performance declines dramatically with the smaller number of labeled training samples and in the presence of significant noise. To address this issue, we propose an improved Bootstrap Your Own Late (BYOL) self-supervised learning scheme to fully exploit the unlabeled samples, which comprises the pretext task adopting contrastive learning conception and the downstream task. We designed three optimized data augmentation methods for communication signals in the former task to serve the contrastive concept. We built two neural networks, online and target networks, which interact and learn from each other. The proposed scheme demonstrates the generality of handling the small and sufficient sample cases across a wide range from 10 to 400, being labeled in each group. The experiment also shows promising accuracy and robustness where the recognition results increase at 3-8% from 3 to 7 signal-to-noise ratio (SNR). Our scheme can accurately identify the individual emitter in a complicated electromagnetic environment.

Keywords: specific emitter identification; self-supervised learning; small samples; deep learning; signal processing

Citation: Zhao, D.; Yang, J.; Liu, H.; Huang, K. Specific Emitter Identification Model Based on Improved BYOL Self-Supervised Learning. *Electronics* **2022**, *11*, 3485. https://doi.org/10.3390/electronics11213485

Academic Editor: Osvaldo Gervasi

Received: 27 September 2022
Accepted: 25 October 2022
Published: 27 October 2022

1. Introduction

The radiofrequency (RF) fingerprint is an inherent feature of the transmitter, mainly caused by hardware defects in the manufacturing process [1]. The characteristics of electronic circuits and RF components determined by the production and manufacturing processes make the specific emitter identification (SEI) of communication equipment achievable [2]. SEI identifies an emitter individual by extracting RF fingerprint features. This technology is widely used in cognitive radio, communication band management, and military communication scenarios [3].

The traditional SEI scheme is mainly based on manual feature extraction, which classifies the signal by extracting the RF fingerprint feature of the signal [4]. As shown in Figure 1, the signals emitted by the emitters can show different characteristics at different stages, namely steady-state and transient signals [5]. Specifically, the transient signal refers to the signal segment sent by the transmitter power from 0 to stable power. It has a short duration but rich characteristics [6–10]. A steady-state signal refers to the signal segment sent after the transmitter reaches the rated power. Its duration is long and it is easier to extract steady-state features [11–15]. However, these traditional SEI schemes rely heavily on expert knowledge and prior knowledge [5].

The limitations of existing SEI schemes can be overcome by recently proposed deep learning algorithms, which can extract useful features without relying on expert knowledge [16]. For example, Zha et al. [17] used a baseband signal as the input of a complex-valued Fourier neural network and introduced time-domain and frequency-domain attention mechanisms. Compared with several state-of-the-art SEI schemes, this scheme

performs better. Yang et al. [18] used a convolutional neural network (CNN) for RF fingerprint extraction. They used many labeled and unlabeled training data to improve the network's generalization ability. The numerical results show that the classification accuracy is approximately 90% at 10 dB. However, as the electromagnetic environment becomes increasing complicated and the signal style becomes more and more changeable, the cost of manually labeling samples is increasing. As a result, most samples in the sample set are unlabeled. However, the manual feature extraction-based and deep learning-based methods require a large number of label samples, leading to significantly reduced recognition accuracy with small samples.

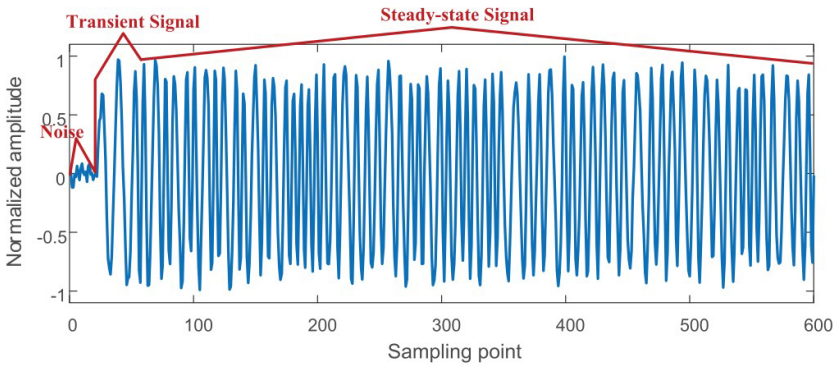

Figure 1. Different parts of the signal sent by the emitter.

Self-supervised learning is a method to improve the model's feature extraction ability by designing auxiliary tasks to improve the model's feature extraction ability. It uses many unlabeled samples to pretrain the model, which gives it a better feature representation ability. In recent years, as a kind of unsupervised learning, self-supervised learning has made significant breakthroughs in natural language processing [19], such as bidirectional encoder representation from Transformers (BERT) [20] and generative pretraining (GPT) [21]. In the past, in the field of computer vision, because the original signal was in a continuous high-dimensional space, unsupervised learning could not effectively reduce the dimension of features, which means the supervised network is still in the dominant position. Recently, some self-supervised learning models (such as the simple framework for the contrastive learning of visual representations (SimCLR) [22], swapping assignments between multiple views (SwAV) [23], and momentum contrast (MoCo) [24]) have been put forward successively, proving that self-supervised learning can achieve the same effect as supervised learning [25]. Therefore, we propose a scheme of SEI based on self-supervised learning, trying to solve the problem of the low accuracy of emitter identification with small samples.

Self-supervised learning methods can be divided into three categories: context-based [26], temporal-based [27], and contrastive-based [28], which are generally divided into two stages: pretext tasks and downstream tasks. Context-based and temporal-based self-supervised learning methods are mainly used in text and video, while the scheme of SEI is mainly based on signal processing. Therefore, contrastive-based self-supervised learning is a better choice. State-of-the-art contrastive methods [22,24,29,30] are trained by reducing the distance between the representations of different augmented views of the same sample ('positive pairs') and increasing the distance between the representations of augmented views from different samples ('negative pairs') [31]. Based on the existing self-supervised learning method, we combined it with the complex-valued neural network and propose a complex-valued self-supervised learning-based scheme for SEI to improve the anti-noise performance of the network [32]. However, these methods need to rely on a large batch size [22] or memory banks [24] to retrieve negative pairs to improve the performance of contrastive learning. Grill et al. proposed the Bootstrap Your Own Latent (BYOL) self-supervised learning method [31], which achieved high recognition accuracy without using negative pairs. However, BYOL is a self-supervised learning method for image

representation whose data augmentation methods are all designed to obtain an enhanced view of the image. The main task of SEI is signal processing, in which the inherent continuity and timing of the signal are the most significant difference from the image processing. For BYOL to complete signal processing, this paper improves it to generate enhanced views of the signal.

We proposed the improved BYOL for SEI, a new scheme for the self-supervised learning of signal representations. The improved BYOL has two networks, the target and online network, which interact and learn from each other. The online network mainly trains a new and potentially enhanced representation by predicting the target signal representation. The target network primarily serves the online network and provides the target signal representation for the online network. The improved BYOL performs SEI in three steps. Firstly, many unlabeled samples are used to pretrain the network in the stage of the pretext task to obtain the encoder's initial weight. Secondly, a small number of labeled samples are used to transfer the encoder in the stage of the downstream task. Finally, the trained encoder is used for SEI. The main contributions of this paper are as follows.

- We first proposed an SEI model based on improved BYOL self-supervised learning. To the best of our knowledge know, this is the first scheme applying self-supervised knowledge to SEI. Compared with the traditional data augmentation and residual networks, the improved BYOL scheme can obtain better recognition accuracy and anti-noise performance with small samples.
- We designed three new data augmentation methods: phase rotation, random cropping, and jitter. Through different data enhancement methods, the network can obtain the positive and negative samples needed for contrastive learning to carry out self-supervised constraints.
- Recent self-supervised learning methods based on contrastive learning require negative samples. They need a careful treatment of negative pairs by relying on large batch sizes or memory banks, which significantly increases the network's computing resources, making it impossible to implement in small terminals. Our scheme removes negative samples, which can be implemented with minimal resources, and the recognition accuracy exceeds the latest self-supervised learning algorithm, significantly enhancing the algorithm's application.

The rest of this paper is organized as follows: the Section 2 briefly introduces the SEI system model. The Section 3 introduces self-supervised and contrastive learning. The Section 4 provides the design details of improved BYOL self-supervised learning for SEI. The Section 5 discusses the results of comparative experiments on real datasets. The Section 6 summarizes this paper.

2. System Model

There is a quadrature receiver and N identical emitters in an open space. We consider a signal receiver that performs SEI to determine which particular device the received radio signal comes from. It is a multi-classification problem. It is assumed that only one emitter is active in each period, which means that the signal of each emitter can be collected separately. The received signal is expressed as:

$$s(t) = f_k(r(t)) * h(t) + n(t), \tag{1}$$

where $h(t)$ is the channel response; f_k is the individual characteristics of the emitters; '$*$' represents convolution. $n(t)$ represents noise; $r(t)$ refers to the signal transmitted by the transmitter; and the formula is:

$$r(t) = Acos(2\pi ft + \varphi), \tag{2}$$

where A represents the amplitude of the signal; and φ represents signal's phase. The receiver samples the signal at the sampling frequency of F_s, and the discrete-time sample is expressed as:

$$s(n) = \sqrt{s_i^2(n) + s_q^2(n)}, \tag{3}$$

where $s_i(n)$ is the in-phase signal and $s_q(n)$ is the quadrature-phase signal. The complex electromagnetic environment often leads to the uneven distribution of open transmission data. In other words, the values of some dataset features are often non-Gaussian. For deep learning, data conforming to normal distribution can improve the learning efficiency of the network and accelerate the convergence speed of the network [33]. Therefore, we need to standardize the data. The formula is as follows:

$$x' = \frac{x - \mu}{\sigma}, \tag{4}$$

where μ is the mean of data x and σ is the standard deviation. After normalizing the data, we divide the sampled datasets into two categories: one is the unlabeled dataset $\mathcal{S} = \{s_1, s_2, \cdots, s_n\}$, which has a large number. The other is the labeled dataset $\mathcal{X} = \{x_1, x_2, \cdots, x_m\}$, whose number is small and labels are $\mathcal{L} = \{l_1, l_2, \cdots, l_m\}$.

SEI is generally considered a pattern recognition problem, which mainly includes four steps: signal acquisition, data pre-processing, feature extraction, and classification [18]. The SEI scheme based on deep learning can automatically extract the key RF fingerprint features in the signal without expert knowledge, as shown in Figure 2. As a kind of deep learning, self-supervised learning can effectively use unlabeled samples to increase the recognition accuracy with small samples.

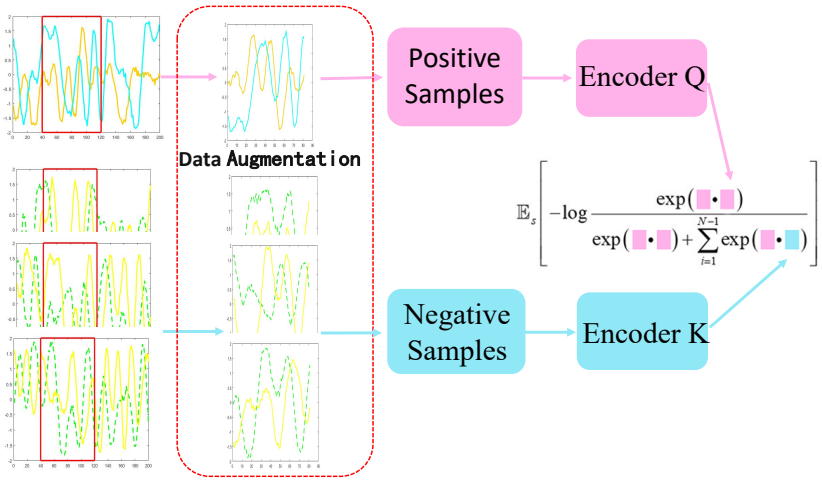

Figure 2. SEI method based on deep learning.

3. Related Work

Self-supervised learning methods can be divided into three categories: context-based, temporal-based, and contrastive-based, which have demonstrated strong capabilities in the natural language processing [34,35]. Among them, state-of-the-art contrastive methods are trained by reducing the distance between representations of different augmented views of the same image ('positive pairs') and increasing the distance between the representations of augmented views of different images ('negative pairs') [31].

3.1. Context-Based Self-Supervised Learning

The context-based method mainly trains the encoder by mining the context information of the data itself. In natural language processing, context-based self-supervised learning methods always use the context relationship in sentences to train the network [26], such as predicting the middle word with the front and back words [36] or the front and back words with the middle word [37].

3.2. Temporal-Based Self-Supervised Learning

The temporal-based method is mainly used in the field of video. Because the adjacent frames in the video have identical features, while the video frames far apart are different [38], we can construct similar (positive) and non-similar (negative) samples to achieve self-supervised constraints.

3.3. Contrastive-Based Self-Supervised Learning

MoCo [24] and SimCLR [22] first proposed a self-supervised learning scheme based on contrastive learning in image processing. As shown in Figure 3, we applied the self-supervised learning method based on contrastive learning to SEI. The method adopts a twin network structure, one of which is used to generate positive samples, and the other is used to generate negative samples. In a batch picture, different augmentation views of the same picture are positive sample pairs, while the other pictures in the batch are negative samples. By defining the loss function, the network makes the representations of positive samples more similar and the representations of positive and negative samples more separated to complete the contrastive learning. However, the number of negative samples will directly determine the effect of contrastive learning [39]. To improve the recognition accuracy, SimCLR [22] ensures a sufficient number of negative samples using a large batch size, and MoCo [24] increases the number of negative samples by storing negative samples from different batches into a queue. These schemes improved the effect of contrastive learning, but they need a lot of computing resources, which prompts the question as to whether using negative pairs is necessary.

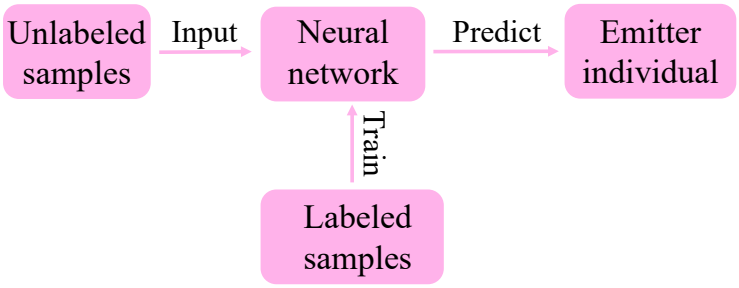

Figure 3. Contrastive-based self-supervised learning method for SEI.

A simple removal of negative samples is often not feasible. The key to contrastive learning lies in contrastive loss, which is in the following forms:

$$L_{contrast} = \mathbb{E}\left[-log\frac{e^{f_x^T f_y}}{e^{f_x^T f_y} + \sum_i e^{f_x^T f_{y_i^-}}}\right]. \quad (5)$$

This loss can be decomposed into two parts: $\mathbb{E}\left[-f_x^T f_y\right]$ and $\mathbb{E}\left[log\left(e^1 + \sum_i e^{f_x^T f_{y_i^-}}\right)\right]$. The former part is called alignment, hoping that the features of positive sample pairs are close to each other. The other part is called uniformity, which means that all eigenvectors are evenly distributed on the unit sphere. Adding no negative sample pairs is equivalent to the lack of uniformity in loss, making it easy for the network to output a fixed value for all inputs. As such, the characteristic difference is 0, which perfectly conforms to the optimization goal of the so-called training collapse.

Deepcluster [40] proposed a method that does not use negative pairs. It uses a priori representation to cluster data points and uses the clustering index of each sample as the classification target of the new representation. However, this method requires high-cost clustering stages and specific preventive measures.

Grill et al. proposed the BYOL self-supervised learning scheme, a self-supervised representation learning technology for reinforcement learning that can effectively prevent

training collapse [31]. It has two encoder networks; one is the online network, and the other is the target network. The network can avoid training collapse through the asymmetric network structure and the strategy of slow momentum update. Although the BYOL scheme performs well on images, its data augmentation methods are image-based and unsuitable for signals.

The self-supervised learning scheme can effectively use unlabeled samples to pretrain the network so that the network can obtain high recognition accuracy with small samples. However, most of the self-supervised learning schemes at this stage are based on images and unsuitable for signals. Therefore, this paper can try to improve them so that they can be applied to SEI.

4. Methodology

This section introduces our scheme in two parts. The first is to introduce the three methods of data augmentation. The second is to introduce the implementation details of the improved BYOL.

4.1. Data Augmentation

We designed three data augmentation methods for communication signals: phase rotation, random cropping, and jitter. We can obtain different data enhancement views of the same signal segment through these three data-augmentation methods to realize contrastive learning.

After the receiver samples the signal, the in-phase and quadrature signals (I/Q) are output. Phase rotation aims to multiply the I/Q signal by a random phase to achieve the purpose of data augmentation. Its formula is as follows:

$$s(t)' = s(t)e^{i\theta}, \tag{6}$$

where the original signal $s(t)$ is a complex signal. i represents imaginary units. $s(t)'$ is the signal after phase rotation. θ is a constant that obeys the uniform distribution of $[0, 2\pi]$. Through phase reversal, we can change the phase of the I/Q signals at the same time to simulate the impact of the channel on its phase in the process of signal transmission.

In the field of images, random cropping is a standard data augmentation method. They randomly erased a part of the image and construct an auxiliary task to restore the image or cut the image randomly into pieces of the same size, mark them, and build auxiliary tasks to make the network perform the puzzle correctly. Image cropping is usually random and discontinuous, and fragmented images can still retain image features. However, too many signal segments will lead to the severe loss of fingerprint features, resulting in reduced recognition accuracy. In other words, the specific RF fingerprint features can be only preserved by preserving the continuity of the signal. Therefore, in contrast to the random cropping in the image field, we randomly set cropping points at the head and tail of the signal to clip out a continuous section of the signal in the middle, thus preserving the continuity of the signal. We will cut the signal to a fixed length to further meet the signal processing requirements.

The signal will inevitably be affected by noise in the transmission process. The inherent noise of the channel, power amplifier, and transmitter will be automatically superimposed on the signal. Therefore, the last method, 'jitter', is to artificially add additive Gaussian white noise to augment the data to simulate the influence of additive noise in the process of the signal legend to achieve the purpose of data expansion. The jitter expression is as follows:

$$s(t)' = s(t) + N(t), \tag{7}$$

where $N(t)$ is the additive Gaussian white noise. Its mean value is 0, and its variance satisfies the uniform distribution of $(0, 1)$.

4.2. Improved BYOL

Figure 4 shows the improved BYOL method. Like most self-supervised learning methods, our method is mainly divided into two parts: the pretext and downstream task. In the front task stage, there are two networks: the target network and the online network. The online network mainly trains a new and potentially enhanced representation by predicting the target representation. The target network aims to provide the target representation for the online network. Many unlabeled samples are input into the two networks for contrastive learning after data augmentation to complete the update iteration of online network model parameters. In the stage of the downstream task, we transfer the encoder to the online network and use a small number of labeled samples to fine-tune the model parameters to perform SEI.

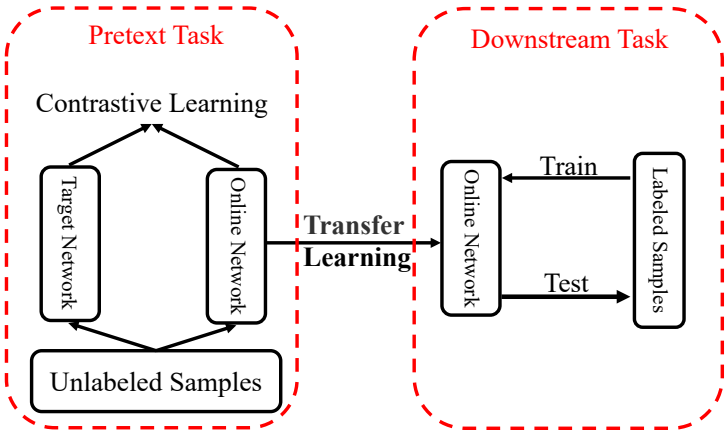

Figure 4. Improved BYOL method.

Figure 5 shows the stage of the pretext task. Our method aims to let the encoder learn a good feature representation to obtain high recognition accuracy with small samples. As mentioned above, the improved BYOL uses two neural networks for learning: the target and online networks. The online network consists of three parts—encoder, projector, and predictor—While the target network only consists of two parts: the encoder and projector. Because the architecture of the online network and the target network is asymmetric, it can effectively prevent the output of the two networks from being the same so that the characteristics collapse into one point [31]. The initialization weight of the target network is the same as that of the online network, but the update method is different. The online network adopts the gradient descent back propagation algorithm, while the target network adopts the momentum update algorithm [24].

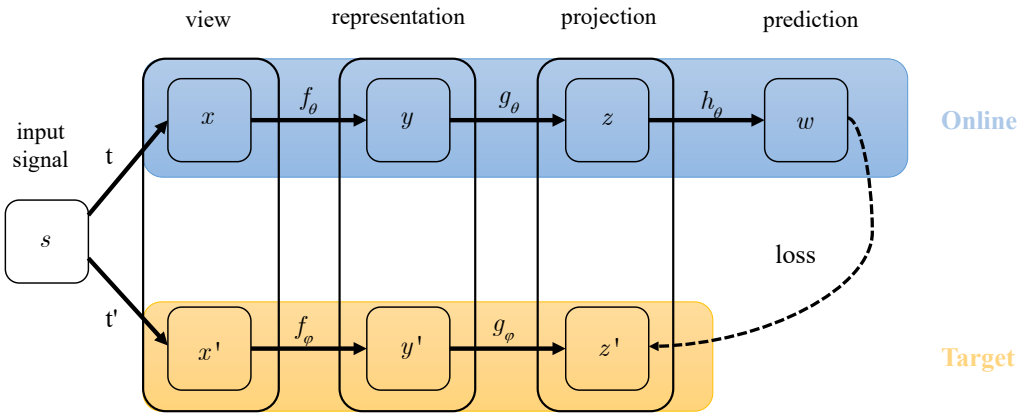

Figure 5. Pretext task.

Give a signal set $\mathcal{S}$, and uniformly sample $s \in \mathcal{S}$. Perform data augmentation t and $t^{'}$ for s twice, respectively, to obtain two augmented views $x = t(s)$ and $x^{'} = t^{'}(s)$. Then, the output prediction values of the two networks, $w = h_\theta(g_\theta(f_\theta(t(s))))$ and $z^{'} = g_\varphi(f_\varphi(t^{'}(s)))$, are obtained from the two enhanced views through the encoder, projector and predictor in turn. We take the mean square error as the loss function of the model training, and the formula is as follows:

$$\mathcal{L} = \left\| \overline{w} - \overline{z^{'}} \right\|_2^2 = 2 - 2 \cdot \frac{\left\langle w, z^{'} \right\rangle}{\|w\|_2 \cdot \|z^{'}\|_2} \tag{8}$$

To symmetrize the parameters θ and φ, we input different data augmentation views into each other's network. That is to say, each data augmented view should not only pass through the target network but also through the online network. Then, we will get two sets of output predictions: $w_1 = h_\theta(g_\theta(f_\theta(t(s))))$, $z_1^{'} = g_\varphi(f_\varphi(t^{'}(s)))$ and $w_2 = h_\theta(g_\theta(f_\theta(t^{'}(s))))$, $z_2^{'} = g_\varphi(f_\varphi(t(s)))$. Therefore, the loss function will also be adjusted to:

$$\mathcal{L} = \left\| \overline{w_1} - \overline{z_1^{'}} \right\|_2^2 + \left\| \overline{w_2} - \overline{z_2^{'}} \right\|_2^2 \tag{9}$$

After completing the pretext task training, we obtained the encoder f_θ with great data representation ability. Then, in the downstream task, a small number of labeled samples are used to transfer the encoder and fine-tune its parameters.

Figure 6 shows the neural network structure of the pretext task. The residual network (ResNet) is used as the encoder and MLP as the projector and predictor. We replace the full connection layer and the softmax classifier of the encoder with an MLP. These MLPs include a linear layer with a size of 4096, a batch normalization layer, an activation function ReLU layer, and a linear layer with a final input size of 256. For the encoder and projector of the target network, except that the initialization is precisely the same as the weight of the online network, the subsequent parameter update is entirely based on the momentum update algorithm to make it change slowly and prevent training collapse.

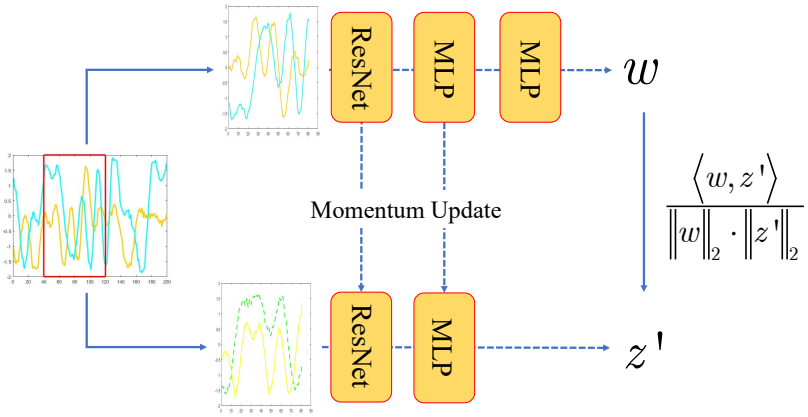

Figure 6. Neural network structure of pretext task.

Algorithm 1 provides the improved BYOL pseudo-code for the pretext task. For the current batch, we input two different data augmentation views of the same graph into the online network and the target network, respectively, which are positive sample pairs.

Algorithm 1 Pseudocode of improved BYOL

Require:

$f_\theta, g_\theta, h_\theta$: online encoder, projector, and predictor
f_φ, g_φ: target encoder, projector
$f_\theta.params$=$f_\varphi.params$
$g_\theta.params$=$g_\varphi.params$
for s in loader **do** # load a batch s
    $x = aug(s)$ # a randomly augmented version
    $x^{'} = aug(s)$ # another randomly augmented version
    $w_1 = h_\theta(g_\theta(f_\theta(t(s))))$
    $w_2 = h_\theta(g_\theta(f_\theta(t^{'}(s))))$
    $z_1^{'} = g_\varphi(f_\varphi(t(s)))$
    $z_2^{'} = g_\varphi(f_\varphi(t^{'}(s)))$
    $z_{1,2}^{'} = z_{1,2}^{'}$.detach() # no gradient to target network
    $loss \leftarrow -2 \cdot (\frac{\langle w_1, z_1^{'} \rangle}{\|w_1\|_2 \cdot \|z_1^{'}\|_2} + \frac{\langle w_2, z_2^{'} \rangle}{\|w_2\|_2 \cdot \|z_2^{'}\|_2})$
    update($f_\theta.params, g_\theta.params$)
    $f_\varphi.params = m \cdot f_\varphi.params + (1-m) \cdot f_\theta.params$
    $g_\varphi.params = m \cdot g_\varphi.params + (1-m) \cdot g_\theta.params$
end for

Ensure: encoder f_θ

5. Experiments

In this section, we first introduce the datasets and the model parameters. Second, we discuss the superior performance of our method compared with the existing MoCo self-supervised learning and traditional data augmentation methods. Because the BYOL self-supervised learning method only applies to images, it cannot complete the SEI task.

5.1. Datasets and Parameter Settings

Our experimental system is Windows 10. The graphics card is an RTX3060. The deep learning framework is Pytorch 1.2, and the compilation environment is Python 3.8.

The data of the pretext task and downstream task w4re collected in the same environment, which comes from 8 emitte. The carrier frequency of the signal is 500 MHz. The sampling frequency of the receiver is 50 MHz. The number of sampling points is 16,384, of which the first 8192 points are in-phase signals, and the last 8192 points are quadrature signals.

Table 1 shows the main parameters of the network model we set in the pretext task. We used a smaller batch size(64), which significantly reduces the computing resources and improves the practicality of the model. A state-of-the-art self-supervised learning method, such as SimCLR, requires a larger batch size (8192), or MoCo needs a large memory bank, which leads to the fact that network training can only be implemented on large servers, reducing the universality of the method. A total of 80,000 unlabeled samples were used to pretrain the model. They all came from the above eight emitters, but there were no labels.

Table 1. Main parameter settings of pretext task.

Signal Parameter	Parameter Value
Learning rate	0.0001
Batch size	64
Epoch	1000
Momentum values	0.99
Optimizer	Adam
Loss function	Mean square

Table 2 shows the main parameters of the network model in the downstream task. There are 800 labeled samples for each type of emitter. We used 400 samples as training samples to fine-tune the weight parameters of the online network encoder. The remaining 400 samples were test samples. It can be seen from the table that we used a significant learning rate in the downstream tasks, which will cause the accuracy of the network to fluctuate severely in the learning process. Therefore, we used a dynamic learning rate. Through experiments, it was found that it is most appropriate to reduce every 100 epochs to the original 10%. The following figures show the change in network learning accuracy with an epoch under different learning rate decays.

Table 2. Main parameter settings of downstream task.

Signal Parameter	Parameter Value
Learning rate	0.01
Batch size	64
Epoch	1000
Optimizer	Adam
Loss function	Cross entropy

From Figure 7, we can see that, if the learning rate decay is not adopted, the network will easily fall into the overfitting state because of the excessively large learning rate. If the decay is 10% for every 200 epochs, the overfitting of the network is indeed prevented to some extent, but it can be seen from the figure that the network still has inevitable fluctuations and is not exceptionally stable. In addition, if the learning rate decays too much, it have a negative impact. It can be seen from the figure that when the learning rate decays by 50% for every 100 epochs, the recognition accuracy of the network cannot reach the optimal state.

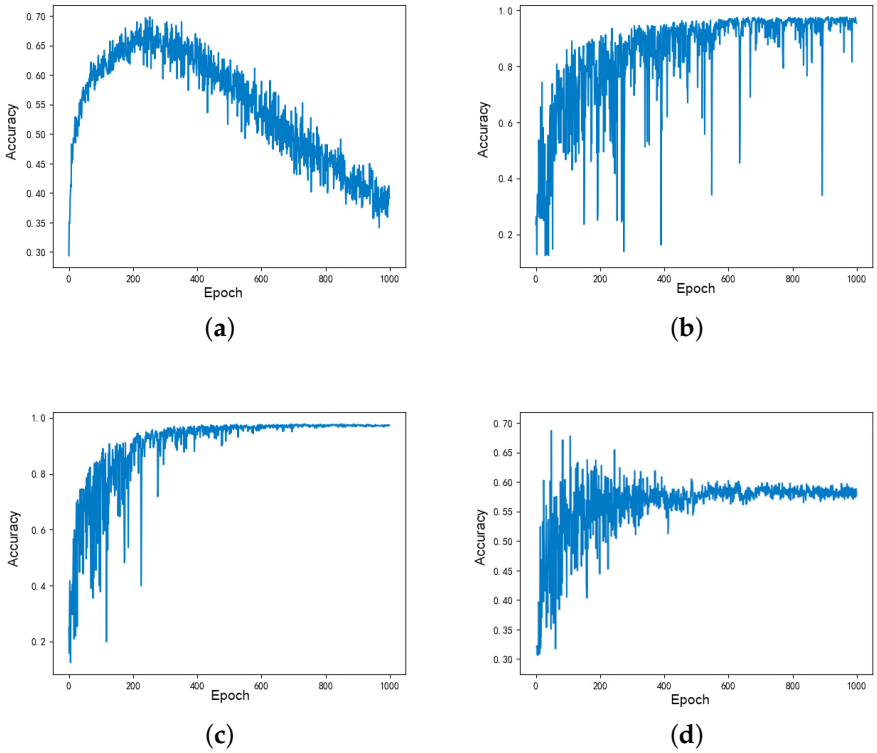

Figure 7. Recognition accuracy under different learning rates: (**a**) Without learning rate attenuation; (**b**) decay 10% for every 200 epochs; (**c**) decay 10% for every 100 epochs; and (**d**) decay 50% for every 100 epochs.

5.2. Performs vs. Sample Number

We test our model from two aspects. First, under the sufficient signal-to-noise ratio(SNR), we compare the method with the MoCo self-supervised learning and the traditional data augmentation network and ResNet. The other is to artificially add Gaussian white noise to the test samples to reduce the signal-to-noise ratio and compare our method with the three above methods.

First, under the condition of a sufficient signal-to-noise ratio (SNR > 15 dB), without adding Gaussian white noise to the test samples, we tested the individual recognition accuracy of the emitter with different numbers, especially under the condition of small samples. We take the average of the five experimental results as the final value, and the results are shown in the table below ('DA' means data augmentation).

As can be seen from Table 3, compared with the MoCo self-supervised learning, our method has a specific improvement in recognition accuracy with different samples across a wide range from 10 to 400. Results show that our method improves the recognition accuracy after removing negative samples. When our method is compared with the traditional data augmentation network and residual network, it can be seen that under the condition of small samples, we can increase the recognition accuracy by nearly 10%. With the increase in the number of samples, the improved BYOL method is more effective than other algorithms, indicating that the rise in the number of samples can significantly improve the recognition performance of this model. In the pretext task stage, we use many unlabeled samples to pretrain the model so that the online network encoder can obtain a better signal representation ability. In the stage of the downstream task, only a small number of labeled samples are required to obtain a high recognition accuracy, which solves the problem of low recognition accuracy with small samples.

Table 3. Recognition accuracy under different sample numbers.

Number / Acc (%)	10	15	20	25	200	400
Residual network	23.65	25.18	26.28	27.50	48.18	65.65
DA+residual network	55.09	60.62	67.09	76.84	89.84	91.00
MoCo	68.12	72.00	75.21	79.62	90.91	93.06
Improved BYOL	70.56	76.84	78.87	80.96	92.13	96.25

5.3. Performs vs. SNR

To further simulate the influence of various additive noises on signal transmission, we artificially add additive Gaussian white noise to the test samples and reduce its signal-to-noise ratio to less than 10dB. In addition, the number of training and test samples for each emitter type is 400. We take the average of the five experimental results as the final value, and the results are shown in the table below (where 'DA' means data augmentation).

As can be seen from Table 4, compared with MoCo, our method can improve the recognition accuracy by nearly 5% in the range of 3 dB–7 dB signal-to-noise ratio. With the reduction in the signal-to-noise ratio (from 7 dB to 3 dB), our method decreased by approximately 5%, while the MoCo method only decreased by approximately 2%, which shows that our method can adapt to the signal with extreme noise. With the increased SNR, the improved BYOL method is more effective than other algorithms, which shows that this method has more vital recognition ability in samples with a high SNR.

Table 4. Recognition accuracy under different signal-to-noise ratios.

SNR / Acc (%)	3	4	5	6	7
Residual network	16.43	24.62	30.65	32.22	34.65
DA+residual network	81.37	82.43	83.71	83.84	86.34
MoCo	83.15	83.49	84.18	84.72	85.78
Improved BYOL	85.18	86.75	89.03	90.78	91.34

The residual network is susceptible to noise, which leads to low recognition accuracy. The data augmentation method can still have strong anti-noise performance with sufficient samples, and the recognition accuracy does not decline significantly with the signal-to-noise ratio. When our method is compared with the data augmentation and residual networks, it can be seen that when the number of samples is sufficient, our method can be significantly improved compared with the traditional emitter recognition method.

6. Conclusions

In this paper, we first propose the SEI model based on improved BYOL self-supervised learning, which relies on two neural networks: online and target. The encoder of the online network is trained to predict the target network representation of the same sample under a different augmented view. Although state-of-the-art self-supervised learning methods rely on negative samples, the scheme achieves a new state-of-the-art without them. By removing negative samples, the algorithm significantly reduces the computational resources and improves the model's practicability. By using many unlabeled samples to pretrain the encoder in pretext tasks, we achieved a massive improvement in identification accuracy with small samples. The results show that our method can improve recognition by 3–8% compared to other existing methods.

Author Contributions: Conceptualization, D.Z. and K.H.; methodology, D.Z.; software, D.Z.; validation, J.Y. and H.L.; formal analysis, D.Z.; investigation, J.Y.; resources, H.L.; data curation, H.L.; writing—original draft preparation, D.Z.; writing—review and editing, J.Y.; visualization, D.Z.; supervision, J.Y.; project administration, J.Y.; funding acquisition, H.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Baldini, G.; Steri, G.; Giuliani, R. Identification of wireless devices from their physical layer radio-frequency fingerprints. In *Encyclopedia of Information Science and Technology*, 4th ed.; IGI Global: Hershey, PA, USA, 2018; pp. 6136–6146.
2. Qu, L.; Yang, J.; Huang, K.; Liu, H. Specific emitter identification based on one-dimensional complex-valued residual networks with an attention mechanism. *Bull. Pol. Acad. Sci. Tech. Sci.* **2021**, *69*, e138814.
3. Huang, K.; Yang, J.; Liu, H.; Hu, P. Deep adversarial neural network for specific emitter identification under varying frequency. *Bull. Pol. Acad. Sci. Tech. Sci.* **2021**, *69*, e136737.
4. Talbot, K.I.; Duley, P.R.; Hyatt, M.H. Specific emitter identification and verification. *Technol. Rev.* **2003**, *113*, 133.
5. Qian, Y.; Qi, J.; Kuai, X.; Han, G.; Sun, H.; Hong, S. Specific emitter identification based on multi-level sparse representation in automatic identification system. *IEEE Trans. Inf. Forensics Secur.* **2021**, *16*, 2872–2884. [CrossRef]
6. Ezuma, M.; Erden, F.; Anjinappa, C.K.; Ozdemir, O.; Guvenc, I. Detection and classification of UAVs using RF fingerprints in the presence of Wi-Fi and Bluetooth interference. *IEEE Open J. Commun. Soc.* **2019**, *1*, 60–76. [CrossRef]
7. Ezuma, M.; Erden, F.; Anjinappa, C.K.; Ozdemir, O.; Guvenc, I. Micro-UAV detection and classification from RF fingerprints using machine learning techniques. In Proceedings of the 2019 IEEE Aerospace Conference, Big Sky, MT, USA, 2–9 March 2019; pp. 1–13.
8. Ali, A.M.; Uzundurukan, E.; Kara, A. Improvements on transient signal detection for RF fingerprinting. In Proceedings of the 2017 25th Signal Processing and Communications Applications Conference (SIU), Antalya, Turkey, 15–18 May 2017; pp. 1–4.
9. Serinken, N.; Ureten, O. Generalised dimension characterisation of radio transmitter turn-on transients. *Electron. Lett.* **2000**, *36*, 1064–1066. [CrossRef]

10. Choe, H.C.; Poole, C.E.; Andrea, M.Y.; Szu, H.H. Novel identification of intercepted signals from unknown radio transmitters. In Proceedings of the Wavelet Applications II, SPIE, Orlando, FL, USA,17–21 April 1995; Volume 2491, pp. 504–517.
11. Zhang, X.D.; Shi, Y.; Bao, Z. A new feature vector using selected bispectra for signal classification with application in radar target recognition. *IEEE Trans. Signal Process.* **2001**, *49*, 1875–1885. [CrossRef]
12. Aubry, A.; Bazzoni, A.; Carotenuto, V.; De Maio, A.; Failla, P. Cumulants-based radar specific emitter identification. In Proceedings of the 2011 IEEE International Workshop on Information Forensics and Security, Iguacu Falls, Brazil, 29 November–2 December 2011; pp. 1–6.
13. López-Risueño, G.; Grajal, J.; Sanz-Osorio, A. Digital channelized receiver based on time-frequency analysis for signal interception. *IEEE Trans. Aerosp. Electron. Syst.* **2005**, *41*, 879–898. [CrossRef]
14. Zhang, J.; Wang, F.; Dobre, O.A.; Zhong, Z. Specific emitter identification via Hilbert—Huang transform in single-hop and relaying scenarios. *IEEE Trans. Inf. Forensics Secur.* **2016**, *11*, 1192–1205. [CrossRef]
15. Lundén, J.; Koivunen, V. Automatic radar waveform recognition. *IEEE J. Sel. Top. Signal Process.* **2007**, *1*, 124–136. [CrossRef]
16. O'Shea, T.J.; Corgan, J.; Clancy, T.C. Convolutional radio modulation recognition networks. In Proceedings of the International Conference on Engineering Applications of Neural Networks, Halkidiki, Greece, 5–7 June 2016; pp. 213–226.
17. Zha, X.; Chen, H.; Li, T.; Qiu, Z.; Feng, Y. Specific Emitter Identification Based on Complex Fourier Neural Network. *IEEE Commun. Lett.* **2021**, *26*, 592–596. [CrossRef]
18. Yang, N.; Zhang, B.; Ding, G.; Wei, Y.; Wei, G.; Wang, J.; Guo, D. Specific emitter identification with limited samples: A model-agnostic meta-learning approach. *IEEE Commun. Lett.* **2021**, *26*, 345–349. [CrossRef]
19. Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; Bengio, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv* **2014**, arXiv:1406.1078.
20. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 84–90. [CrossRef]
21. Graves, A.; Mohamed, A.r.; Hinton, G. Speech recognition with deep recurrent neural networks. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; pp. 6645–6649.
22. Karpathy, A.; Toderici, G.; Shetty, S.; Leung, T.; Sukthankar, R.; Fei-Fei, L. Large-scale video classification with convolutional neural networks. In Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 23–28 June 2014; pp. 1725–1732.
23. Radford, A.; Narasimhan, K.; Salimans, T.; Sutskever, I. Improving Language Understanding by Generative Pre-Training. *cs.ubc.ca*. 2018. Available online: https://www.cs.ubc.ca/~amuham01/LING530/papers/radford2018improving.pdf (accessed on 24 October 2022).
24. He, K.; Fan, H.; Wu, Y.; Xie, S.; Girshick, R. Momentum contrast for unsupervised visual representation learning. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 9729–9738.
25. Radford, A.; Wu, J.; Child, R.; Luan, D.; Amodei, D.; Sutskever, I.; et al. Language models are unsupervised multitask learners. *OpenAI blog* **2019**, *1*, 9.
26. Ferriyan, A.; Thamrin, A.H.; Takeda, K.; Murai, J. Encrypted Malicious Traffic Detection Based on Word2Vec. *Electronics* **2022**, *11*, 679. [CrossRef]
27. Zhang, Z.; Guo, T.; Chen, M. Dialoguebert: A self-supervised learning based dialogue pre-training encoder. In Proceedings of the 30th ACM International Conference on Information & Knowledge Management, Virtual, 1–5 November 2021; pp. 3647–3651.
28. Zhou, Z.; Hu, Y.; Zhang, Y.; Chen, J.; Cai, H. Multiview Deep Graph Infomax to Achieve Unsupervised Graph Embedding. *IEEE Trans. Cybern.* **2022**, 1–11. [CrossRef]
29. Oord, A.v.d.; Li, Y.; Vinyals, O. Representation learning with contrastive predictive coding. *arXiv* **2018**, arXiv:1807.03748.
30. Tian, Y.; Krishnan, D.; Isola, P. Contrastive multiview coding. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; pp. 776–794.
31. Grill, J.B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. Bootstrap your own latent-a new approach to self-supervised learning. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 21271–21284.
32. Zhao, D.; Yang, J.; Liu, H.; Huang, K. A Complex-Valued Self-Supervised Learning-Based Method for Specific Emitter Identification. *Entropy* **2022**, *24*, 851. [CrossRef]
33. Bjorck, N.; Gomes, C.P.; Selman, B.; Weinberger, K.Q. Understanding batch normalization. *arXiv* **2018**, arXiv:1806.02375.
34. Noroozi, M.; Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 69–84.
35. Wu, J.; Wang, X.; Wang, W.Y. Self-supervised dialogue learning. *arXiv* **2019**, arXiv:1907.00448.
36. Xiong, Z.; Shen, Q.; Xiong, Y.; Wang, Y.; Li, W. New generation model of word vector representation based on CBOW or skip-gram. *Comput. Mater. Contin.* **2019**, *60*, 259. [CrossRef]
37. Du, X.; Yan, J.; Zhang, R.; Zha, H. Cross-network skip-gram embedding for joint network alignment and link prediction. *IEEE Trans. Knowl. Data Eng.* **2020**, *34*, 1080–1095. [CrossRef]
38. Sermanet, P.; Lynch, C.; Hsu, J.; Levine, S. Time-contrastive networks: Self-supervised learning from multi-view observation. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Honolulu, HI, USA, 21–26 July 2017; pp. 486–487.

39. Chen, T.; Kornblith, S.; Norouzi, M.; Hinton, G. A simple framework for contrastive learning of visual representations. In Proceedings of the International Conference on Machine Learning, Virtual Event, 13–18 July 2020; pp. 1597–1607.
40. Caron, M.; Bojanowski, P.; Joulin, A.; Douze, M. Deep clustering for unsupervised learning of visual features. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 132–149.