Self-Supervised Spectral-Level Contrastive Learning for Hyperspectral Target Detection

Yulei Wang, *Member, IEEE*, Xi Chen, Enyu Zhao, and Meiping Song

***Abstract*— Deep learning-based hyperspectral target detection (HTD) methods are limited by the lack of prior information. Self-supervised learning is a kind of unsupervised learning, which mainly mines its own self-supervised information from unlabeled data. By training the model with such constructed valid posterior information, a valuable representation model can be learned and can get rid of the dependence of deep models on prior information. To this end, this article proposes a self-supervised spectral-level contrastive learning-based HTD (SCLHTD) method to train a model with spectral difference discrimination capability for HTD in a self-supervised manner. First, the hyperspectral images (HSIs) to be detected are sampled in odd and even bands, and the obtained band subsets are then used to train the corresponding adversarial convolutional autoencoders. Feature extraction part of the trained encoder is then used as the data augmentation function, where the positive and negative pairs are constructed through data augmentation, and the backbone is used to extract the representative vectors of the augmented samples. Second, the representative vectors are mapped to the spectral contrast space using spectral contrastive head, where the similarity and dissimilarity of spectra are learned by maximizing the similarity of positive pairs while minimizing the similarity of negative pairs, so that the backbone can discriminate spectral differences. Finally, aiming at suppressing the background, edge-preserving filters are used in conjunction with space information to process the detection results acquired by utilizing spectrum information via cosine similarity to generate the final detection results. Experimental results illustrate that the proposed SCLHTD method can achieve superior performances for HTD.**

***Index Terms*— Contrastive learning, deep learning, hyperspectral imagery, self-supervised learning, target detection.**

I. Introduction

A HYPERSPECTRAL image (HSI) is a 3-D image with rich spectrum and space information, with the spectral resolution up to nanometer level. Because of the abundant spectrum information, the HSI data can be used to distinguish detailed differences between different substances using spectral curves of each pixel. As a result, hyperspectral target detection (HTD) has emerged, which has been widely utilized in environmental detection [1], mineral surveying [2], medica diagnostics [3], and military camouflage target identification [4], performing an increasingly important role in both civil and military fields.

HTD separates target pixels from the background according to the known target spectrum, and many HTD algorithms have appeared in the past research literature. Traditional state-of-the-art spectral matching-based HTD methods include the adaptive coherence estimation (ACE) [5], the constrained energy minimization (CEM) [6], and the orthogonal subspace projection (OSP) [9], where the CEM detector has gained more attention due to its excellent performance. Some CEM-based algorithms are then proposed, such as the hierarchical CEM detector (hCEM) [7], the ensemble-based CEM (E-CEM) [8], and so on. Sparse representation-based HTD algorithms have also been proposed, such as the sparsity-based target detector (STD) [10] and the combined sparse and cooperative representation (CSCR) detector [11], where the STD reconstructs the spectrum of the pixel to be detected through a linear combination of the least number of dictionary atomic spectra by constructing a complete dictionary and calculates the residual value between the reconstructed pixel and the prior target pixel to obtain the detection result.

In recent years, deep learning has been gradually applied in HSI processing due to the powerful nonlinear feature extraction capability, such as classification [12], band selection [13], unmixing [14], and super-resolution reconstruction [15]. The application of deep learning technology to the field of HTD has become a hot research topic, with a slew of deep learning-based HTD methods being presented. The transfer learning-based HTD in [16] pairs pixels based on the label information from the source domain labeled HSI to expand the training samples and expects the network to learn the differences between spectra, and then transfers the model knowledge to the detection task in the target domain. To get rid of the sensor-dependent transferability, the few-shot learning model based on semisupervised domain adaptation in [17] adaptively transfers the model trained in the source domain to the target domain in an adversarial manner to improve the target detection accuracy. Due to the fact that the available labeled samples are extremely limited and insufficient to train the deep network, the HTD-Net method in [18] adopts the U-net idea to designed a modified autoencoder to generate target signatures, and then find background samples based on linear prediction; finally, the known target pixels are paired with both target and background pixels to augment

Manuscript received 7 February 2023; revised 29 March 2023; accepted 18 April 2023. Date of publication 25 April 2023; date of current version 12 May 2023. This work was supported in part by the National Nature Science Foundation of China under Grant 61801075 and Grant 42271355, in part by the Natural Science Foundation of Liaoning Province under Grant 2022-MS-160, in part by the China Postdoctoral Science Foundation under Grant 2020M670723, and in part by the Fundamental Research Funds for the Central Universities under Grant 3132023238. *(Corresponding author: Enyu Zhao.)*

The authors are with the Center of Hyperspectral Imaging in Remote Sensing (CHIRS), Information Science and Technology College, Dalian Maritime University, Dalian 116026, China (e-mail: wangyulei@dlmu.edu.cn; xi_chen@dlmu.edu.cn; zhaoenyu@dlmu.edu.cn; smping@163.com).

Digital Object Identifier 10.1109/TGRS.2023.3270324

the training samples. An HTD method with an auxiliary generative adversarial network (GAN) is proposed in [19] to generate simulated target and background spectra to expand the training samples. The hyperspectral target detector based on two-stream CNN [20] finds enough background pixels by hybrid sparse representation and classification-based pixel selection, and then blends a prior target spectrum with some typical background pixels to generate sufficient target samples; then, the generated target and background samples are, respectively, constructed with the prior target spectrum into positive and negative training samples to be expended and sent to the two-stream CNN to learn the spectral difference discrimination ability. The background learning based on target suppression constraint (BLTSC) [21] uses CEM to perform coarse detection to obtain background samples, which are fed into an adversarial autoencoder (AAE) with target suppression constraints imposed for training to reconstruct pure background, and finally achieves target detection by comparing the reconstructed background with the original HSI. To utilize the space information to improve the detection performance, a 3-D macro–micro-residual autoencoder is designed and used to extract macro- and micro-features, which are fused and sent to a hierarchical radial basis function (hRBF) detector for background suppression and target preservation [22]. A deep spatial–spectral network (DSSN) is proposed in [23], where a region of interest (ROI) map is obtained through CEM and an edge-preserving filter, and the HSI to be detected and the ROI map are fed into the constructed DSSN to extract spatial and spectral features of interest, and the detection results are obtained using the nearest neighbors (NNs) algorithm.

To liberate the HTD model from dependence on the quality of the priori information, inspired by self-supervised learning, a self-supervised spectral-level contrastive learning-based HTD (SCLHTD) method is proposed in this article. By constructing spectral-level contrastive learning, the backbone for feature extraction is used to obtain discrimination capabilities of spectral similarity and dissimilarity. To construct positive and negative sample pairs for spectral-level contrastive learning, the original HSI is sampled with odd and even bands, and the sampled sub-band images are augmented by adversarial convolutional autoencoder with spectral residual channel attention mechanism to obtain two augmented samples, and thus, the two augmented samples of the pixels at the same position can be considered as positive sample pairs, and the augmented samples of spectral pixels at other positions can be considered as negative sample pairs. Finally, background suppression is performed by an edge-preserving filter. The main contributions of this article are summarized as follows.

1) A pretext task for spectral difference discriminative ability learning is constructed, where the self-supervised signals are constructed by designing pretext tasks to assist the model in spectral discrimination ability learning.
2) Spectral-level contrastive learning is designed to perform spectral similarity and dissimilarity discrimination ability learning in a self-supervised manner. There is no need to use traditional methods to obtain target and background samples to train the model, which can liberate the HTD model performance from dependence on the quality of the priori information.

The remainder of this article is organized as follows. Section II gives a detailed description of the proposed SCLHTD method. The experimental studies and analysis to verify the proposed method are presented in Section III. Finally, the conclusions are drawn in Section IV.

II. Proposed Method

This section shows the details of the proposed self-supervised contrastive learning-based HSI target detector, consisting of three stages: data augmentation, spectral-level contrastive learning, and target detection, as shown in Fig. 1.

The data augmentation stage generates two augmentation samples by the designed data augmentation model, and two augmentation samples exist for any pixel spectrum in the HSI. Positive sample pairs are made up of two augmented samples from the same pixel spectrum, while negative sample pairs are made up of two augmented samples from different pixel spectra. In the stage of spectral-level contrastive learning, the backbone is used to extract the representations of the two augmented samples, and the spectral contrastive head is used to map the representation into the spectral contrast space to learn the spectral difference discriminative ability. In the stage of target detection, the representations of the prior target spectrum and the spectrum of each pixel to be detected are extracted using the trained backbone, and cosine similarity of the extracted representations can be used to obtain the detection result using spectral information, and finally, the space information of the HSI is combined with an edge-preserving filter to filter the spectral detection result to suppress the background, and the final target detection result of the HSI is obtained.

A. Spectral Residual Channel Attention Module

The attention mechanism has been widely used in areas, such as HSI classification [24], and has successfully shown its powerful role. To selectively emphasize informative features and suppress features that are less important for the target detection task [25], the spectral residual channel attention module (SRCAM) is designed to give adaptive weights to the feature obtained by different convolution kernels on the same spectral pixel, which could give more attention to the features that are beneficial to the optimization objective. The structure of SRCAM is shown in Fig. 2, which is used in the feature extraction part of the AAE1 encoder, the feature extraction part of the AAE2 encoder, and the backbone for spectral discriminative feature extraction of the flowchart in Fig. 1.

Given a set of features $\mathbf{Q} \in \mathbb{R}^{B\times C\times L}$, where B, C, and L represent the batch size, the number of channels, and the depth of the feature, respectively. A set of feature tensors is obtained and denoted as $\mathbf{U} \in \mathbb{R}^{B\times C\times L}$ sequentially through CONV1, batch normalization, rectified linear unit (ReLU), and CONV2. The CONV1 and CONV2 represent the 1-D convolution layer, and the parameters k_n, k_s, s, and p represent the number of convolution kernels, the size of the convolution

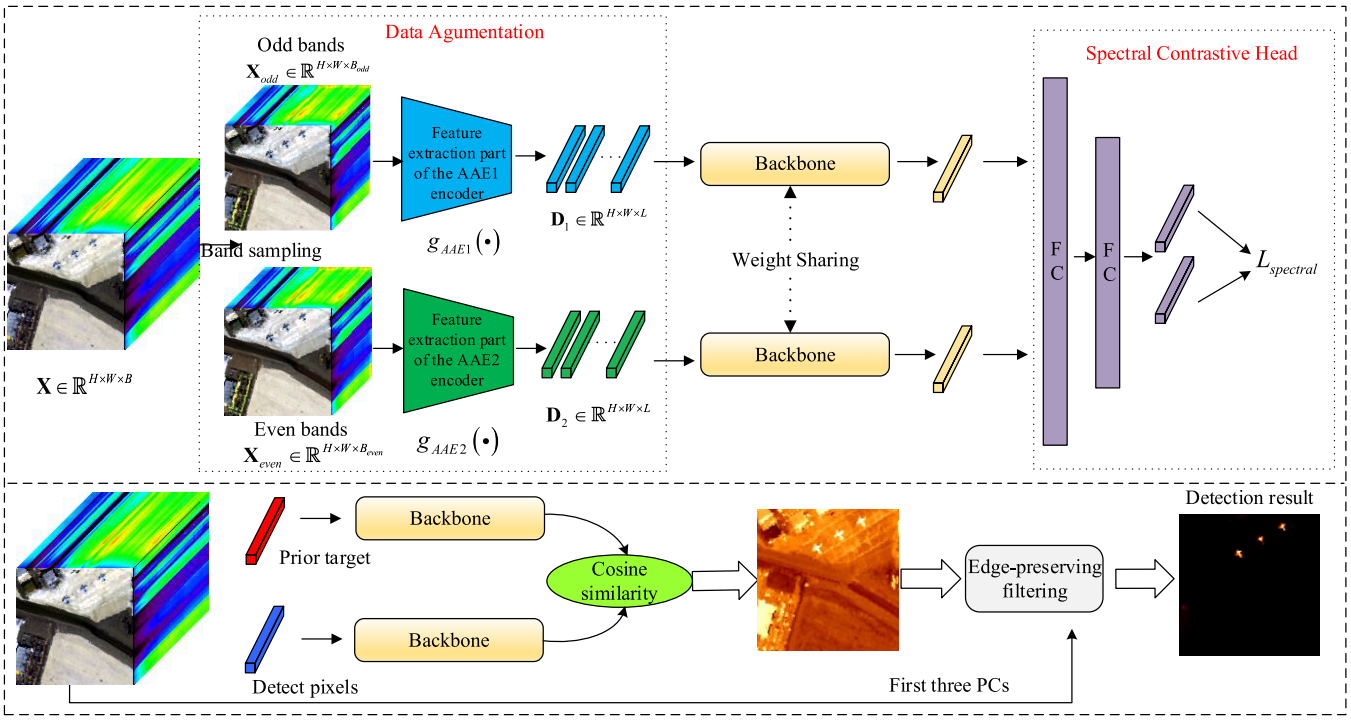

Fig. 1. Flowchart of the proposed SCLHTD method.

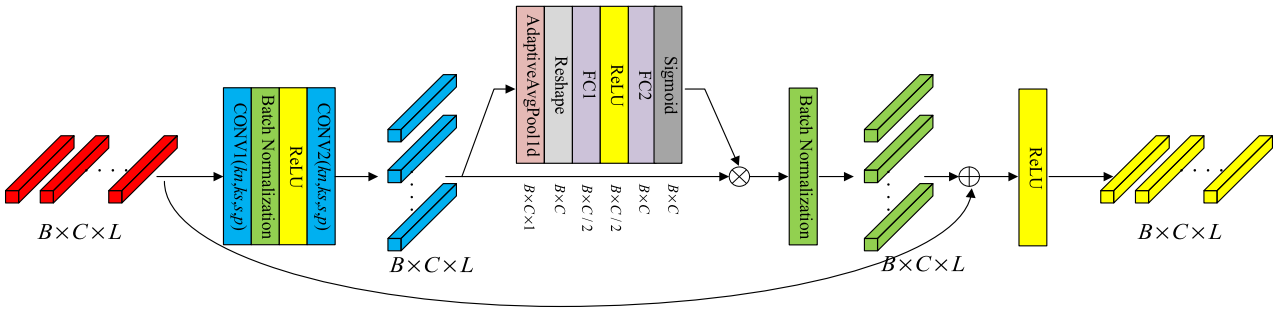

Fig. 2. SRCAM.

kernel, the convolution stride, and padding, respectively. The 1-D convolution operation can be formalized as follows:

$$q_j^l = b_j^l + \sum_{i=1}^{k_s} q_i^{l-1} \times w_{ij}^l \tag{1}$$

where q_j^l and q_i^{l-1} are the values in the features output from the lth and $l-1$th convolutional layers, respectively, w_{ij}^l is the convolutional kernel, and b_j^l is the bias. Then, the batch normalization is performed on the feature obtained after convolution to speed up the model convergency and reduce the internal covariate transition. However, the features learned by the network might be affected when the features are normalized directly, making the network less expressive. As a result, the batch normalization results are dynamically adjusted when learnable parameters a and b are introduced, and the process can be represented as follows:

$$\hat{\mathbf{q}}^l = \gamma^l \frac{\mathbf{q}^l - E[\mathbf{q}^l]}{\sqrt{\mathrm{Var}[\mathbf{q}^l]}} + \beta^l \tag{2}$$

where $E[\cdot]$ and $\mathrm{Var}[\cdot]$ denote the expectation and variance, respectively, $\mathbf{q}^l$ is the output feature of the lth layer, and $\hat{\mathbf{q}}^l$ is the normalization result of the lth layer. The features are batch normalized and then nonlinearly mapped using an activation function. The ReLU is used as the activation function, which can be expressed as follows:

$$\sigma(q_j^l) = \max[0, q_j^l]. \tag{3}$$

Next, in order to obtain the weights of the features between different spectral channels, the feature $\mathbf{U} \in \mathbb{R}^{B\times C\times L}$ output from the CONV2 layer is subjected to adaptive global averaging pooling to compress the spectral depth dimension to obtain $\mathbf{Z} \in \mathbb{R}^{B\times C\times 1}$, which can be described as follows:

$$\mathbf{Z}_k = \frac{1}{L}\sum_{i=1}^{L} \mathbf{U}_k(i) \tag{4}$$

where $k = \{1, 2, \ldots, B \times C\}$. The spectral descriptor vector $\mathbf{Z}$ is then transformed into a 2-D matrix through the reshape layer and fed into the two-layer fully connected (FC) layer

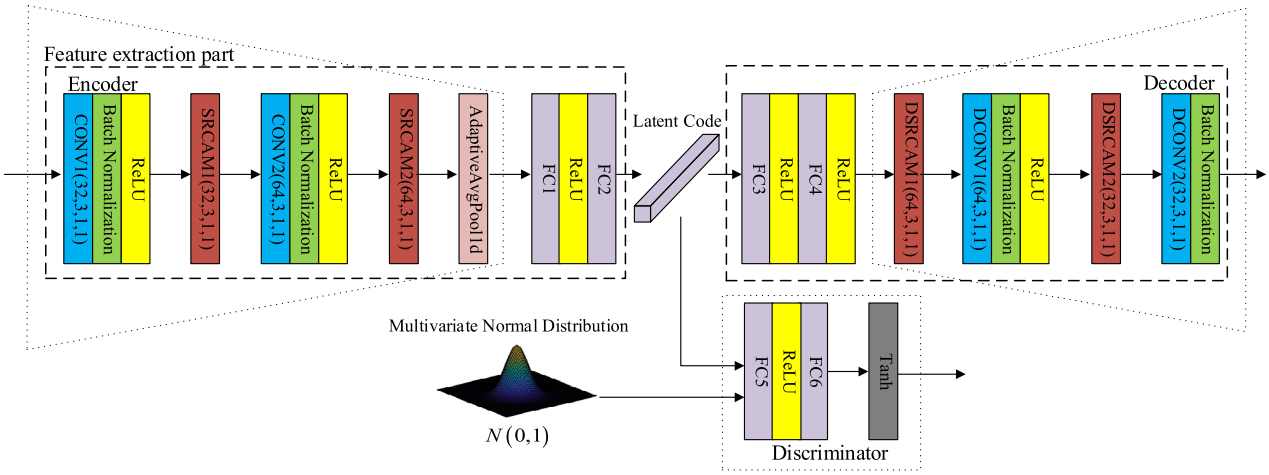

Fig. 3. AAE structure diagram.

to obtain the weights of the spectral channels, which can be formalized as follows:

$$\mathbf{s}_i = \delta(\mathbf{W}_2\sigma(\mathbf{W}_1\mathbf{e}_i)) \tag{5}$$

where $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_B] \in \mathbb{R}^{B\times C}$ is the obtained attention weight vector, and the sigmoid function δ is used to limit the value of the attention weight vector between 0 and 1. $\mathbf{W}_1 \in \mathbb{R}^{C/2}$ and $\mathbf{W}_2 \in \mathbb{R}^{C}$ are the parameters in FC1 and FC2, respectively. The obtained spectral channel attention weights are then multiplied by the features $\mathbf{U} \in \mathbb{R}^{B\times C\times L}$ output from the CONV2 layer, which can be described as follows:

$$\hat{\mathbf{U}}_{ij} = \mathbf{U}_{ij} \cdot \mathbf{s}_{ij} \tag{6}$$

where $\mathbf{U}_{ij}$ is the jth feature vector of the ith feature map in $\mathbf{U}$ and s_{ij} is the jth element of the ith row in $\mathbf{s}$, $i = [1, 2, \ldots, B]$, $j = [1, 2, \ldots, C]$. Finally, the output of SRCAM is

$$\hat{\mathbf{Q}} = \sigma\left(\mathbf{Q} + \hat{\mathbf{U}}\right). \tag{7}$$

B. Data Augmentation

Data augmentation can not only improve the generalization ability of the model, but also act as a regularization process to avoid overfitting, which is essential for learning good representations. For HTD, there is no prior information available other than the prior target spectrum. Therefore, the idea of self-supervised learning is adopted to learn the backbone with the ability to discriminate spectral similarity and dissimilarity through contrastive learning. Since contrastive learning requires positive and negative sample pairs for comparison, the two augmented samples can be regarded as positive sample pairs by performing two data augmentations on the spectra of pixels at the same location. The positive and negative sample pairs can be constructed by data augmentation, which, in turn, constitutes a training set for contrastive learning. The process of data augmentation is as follows.

First, due to the strong correlation between adjacent bands of HSIs [26], the band sampling [27] of the HSI $\mathbf{X} \in \mathbb{R}^{H\times W\times B}$ to be detected is first performed to obtain two HSIs consisting of odd and even bands, denoted as $\mathbf{X}_{\text{odd}} \in \mathbb{R}^{H\times W\times B_{\text{odd}}}$ and $\mathbf{X}_{\text{even}} \in \mathbb{R}^{H\times W\times B_{\text{even}}}$, respectively. Then, two trained encoders are obtained after training the AAE with $\mathbf{X}_{\text{odd}}$ and $\mathbf{X}_{\text{even}}$ separately. The feature extraction part of the two trained encoders $g_{\text{AAE1}}(\cdot)$ and $g_{\text{AAE2}}(\cdot)$ can be considered as a transformation function that plays a role in data augmentation. The feature extraction part in the encoder extracts the features of the pixel spectra, and then maps the features into the latent space by two FC layers. The features extracted by the above operation can well represent the main features of the original pixel spectra [28].

The structure of the AAE for data augmentation is shown in Fig. 3, consisting of an encoder $G_1(\cdot)$, a decoder $G_2(\cdot)$, and a discriminator $D(\cdot)$. The encoder contains two parts: feature extraction and feature mapping. Feature extraction is formed by two convolutional layers, two SRCAMs, and one adaptive average pooling layer, where the specific values of the number of convolutional kernels in the convolutional layer and SRCAM k_n, the size of the convolutional kernel k_s, the convolution stride s, and padding p are shown in Fig. 3 for encoder feature extraction part. The adaptive global average pooling layer is used to fix the features to a specific shape, and fixed dimensional feature vectors can be obtained after the reshape operation. Feature mapping of the encoder is composed of FC layers FC1 and FC2, which are used to map the feature vectors extracted by the former feature extraction part of the encoder into the latent space. The decoder consists of two FC layers FC3 and FC4, two transposed convolutional layers DCONV1 and DCONV2, and two transposed SRCAMs (DSRCAMs). The DSRCAM is a module composed by replacing the convolutional layers in SRCAM with transposed convolutional layers. The FC layers FC3 and FC4 are used to map the latent encoding to a specific dimension for the subsequent part of the decoder to reconstruct the pixel spectrum. The specific parameters in the convolutional layer and DSRCAM are shown in the decoder part of Fig. 3. To reduce the internal covariate transition, batch normalization is used after each convolutional layer. All the layers except the output layer are combined with a ReLU. The discriminator consists of two FC

layers FC5 and FC6. ReLU is used as the activation function of FC5, and a hyperbolic tangent function (Tanh) is used in FC6 to restrict the output feature vector values to the range of [−1, 1].

The training of AAE includes two parts: the autoencoder network and the adversarial network. In the stage of the autoencoder network training, the network is composed of the encoder $G_1(\cdot)$ and the decoder $G_2(\cdot)$. For a given input pixel spectrum $\mathbf{x}$, the autoencoder is trained to optimize the parameters in the network by minimizing the reconstruction loss, which uses the mean square error loss defined as follows:

$$L_r = \|\mathbf{x}_i - G_2(G_1(\mathbf{x}_i))\|_2^2. \tag{8}$$

In the stage of the adversarial network training, the training idea of GAN is used, and the training can be seen as a game process between the generator $G_1(\cdot)$ (encoder) and the discriminator $D(\cdot)$. The goal of the adversarial training is to make the latent encoding result of the generator $G_1(\cdot)$ output closer and closer to the preset prior distribution $p(\mathbf{z})$, while enabling the discriminator $D(\cdot)$ to better distinguish whether the feature vectors are from the latent encoding of the generator output or the vectors sampled from the prior distribution. The prior distribution is a multivariate Gaussian distribution. The optimization objective of the adversarial training process can be expressed as follows:

$$\min_{G_1}\max_{D} E_{\mathbf{z}\sim p(\mathbf{z})}\left[\log D(\mathbf{z})\right]+E_{\mathbf{x}\sim p_{\text{data}}(\mathbf{x})}\left[\log(1-D(G_1(\mathbf{x})))\right]. \tag{9}$$

The adversarial training process first optimizes the discriminator D. The optimization objective of the discriminator D can be expressed as follows:

$$\min -\frac{1}{n}\sum_{i=1}^{n}\left[\log D(\mathbf{z}_i) + \log(1 - D(G_1(\mathbf{x}_i)))\right]. \tag{10}$$

Then, optimize the generator $G_1(\cdot)$, with the following objective expressed as:

$$\min \frac{1}{n}\sum_{i=1}^{n}\left[\log(1 - D(G_1(\mathbf{x}_i)))\right] \tag{11}$$

where $\mathbf{x}_i$ is the input pixel spectrum and $\mathbf{z}_i$ is a vector sampled from the prior distribution. The AAE is trained with $\mathbf{X}_{\text{odd}}$ and $\mathbf{X}_{\text{even}}$, respectively. When the training is completed, the corresponding two encoders can be obtained, and the feature extraction parts $g_{\text{AAE1}}(\cdot)$ and $g_{\text{AAE2}}(\cdot)$ are taken. The final data augmentation samples are then obtained for $\mathbf{X}_{\text{odd}}$ and $\mathbf{X}_{\text{even}}$ using the feature extraction part of the corresponding trained encoder, and the process can be described as follows:

$$\begin{aligned}\mathbf{D}^a &= g_{\text{AAE1}}(\mathbf{X}_{\text{odd}})\\ \mathbf{D}^b &= g_{\text{AAE2}}(\mathbf{X}_{\text{even}})\end{aligned} \tag{12}$$

where $\mathbf{D}^a = [\mathbf{d}_1^a, \mathbf{d}_2^a, \ldots, \mathbf{d}_{H\times W}^a] \in \mathbb{R}^{(H\times W)\times L}$ and $\mathbf{D}^b = [\mathbf{d}_1^b, \mathbf{d}_2^b, \ldots, \mathbf{d}_{H\times W}^b] \in \mathbb{R}^{(H\times W)\times L}$ are the final data augmentation samples. L is the size of the output feature vector of the feature extraction part of the encoder, fixed at 64. By the above method, two data augmentation samples of the pixel spectra at the same position are obtained and used for comparative learning. The procedure of data augmentation is shown in Algorithm 1.

Algorithm 1 Data Augmentation

Input: HSI to be detected $\mathbf{X} \in \mathbb{R}^{H\times W\times B}$, the encoder $G_1(\cdot)$, decoder $G_2(\cdot)$ and discriminator $D(\cdot)$ of the AAE, batch size N, epoch E, learning rate r.

Output: Augmentation samples $\mathbf{D}^a \in \mathbb{R}^{(H\times W)\times L}$ and $\mathbf{D}^b \in \mathbb{R}^{(H\times W)\times L}$.

1. Sample odd bands $\mathbf{X}_{odd} \in \mathbb{R}^{H\times W\times B_{odd}}$ and even bands $\mathbf{X}_{even} \in \mathbb{R}^{H\times W\times B_{even}}$ from $\mathbf{X} \in \mathbb{R}^{H\times W\times B}$ respectively.
2. for $\{\mathbf{X}_{odd}, \mathbf{X}_{even}\}$ do
3. for epoch = 1 to E do
4. for sample batch $\left\{\mathbf{x}_i^{odd}\right\}_{i=1}^{N}$ do
5. compute $L_r = \left\|\mathbf{x}_i - G_2\left(G_1\left(\mathbf{x}_i^{odd}\right)\right)\right\|_2^2$, update encoder $G_1(\cdot)$ and decoder $G_2(\cdot)$ parameters through Adam optimizer.
6. sample vector $\mathbf{z}_i$ from multivariate Gaussian distribution $p(\mathbf{z})$ and send it to discriminator $D(\cdot)$ to get the output $\mathbf{D}(\mathbf{z}_i)$
7. compute $L_D = \log D(\mathbf{z}_i)+\log\left(1 - D\left(G_1\left(\mathbf{x}_i^{odd}\right)\right)\right)$, update discriminator $D(\cdot)$ parameters through SGD optimizer.
8. compute $L_{G_1} = \log\left(1 - D\left(G_1\left(\mathbf{x}_i^{odd}\right)\right)\right)$, update encoder $G_1(\cdot)$ parameters through SGD optimizer.
9. end
10. end
11. obtain encoder $G_1(\cdot)$ of the trained AAE, denote the feature extraction part of encoder $G_1(\cdot)$ as $g_{AAE1}(\cdot)$, and then get the augmentation samples $\mathbf{D}^a$ by $\mathbf{D}^a = g_{AAE1}(\mathbf{X}_{odd})$.
12. end

C. Backbone

The backbone in Fig. 1 is a deep residual CNN with spectral residual channel attention, which is used to extract representative vectors from the augmented data samples in the stage of contrastive learning. The specific structure of the backbone is shown in Fig. 4, which consists of convolution layers, normalization layers, activation layers, SRCAMs, and a reshaping layer. The convolutional layer CONV4 uses a convolutional kernel with size of 1×1 to integrate the features in all channels into a feature vector, which is then transformed into a 1-D feature vector by a reshaping layer to obtain the final extracted representative vector for contrastive learning. The pooling operation is implemented by convolutional downsampling with stride 2. All convolutional layers use 1-D convolution, and the specific parameters of the convolutional layers and SRCAM are shown in Fig. 4.

For the augmented samples $\mathbf{D}^a$ and $\mathbf{D}^b$, the corresponding representative vectors are extracted through the backbone $f(\cdot)$, and the process can be described as follows:

$$\begin{aligned}\mathbf{H}^a &= f\left(\mathbf{D}^a\right) = \left[\mathbf{h}_1^a, \mathbf{h}_2^a, \ldots, \mathbf{h}_{H\times W}^a\right] \in \mathbb{R}^{(H\times W)\times L/4}\\ \mathbf{H}^b &= f\left(\mathbf{D}^b\right) = \left[\mathbf{h}_1^b, \mathbf{h}_2^b, \ldots, \mathbf{h}_{H\times W}^b\right] \in \mathbb{R}^{(H\times W)\times L/4}.\end{aligned} \tag{13}$$

D. SpectralLevel Contrastive Learning

The augmented samples are passed through the backbone $f(\cdot)$ to obtain the corresponding representative vectors, which are fed into the spectral contrastive head for spectral-level contrastive learning. Contrastive learning aims to maximize the similarity between pairs of positive samples while minimizing the similarity between pairs of negative samples [29].

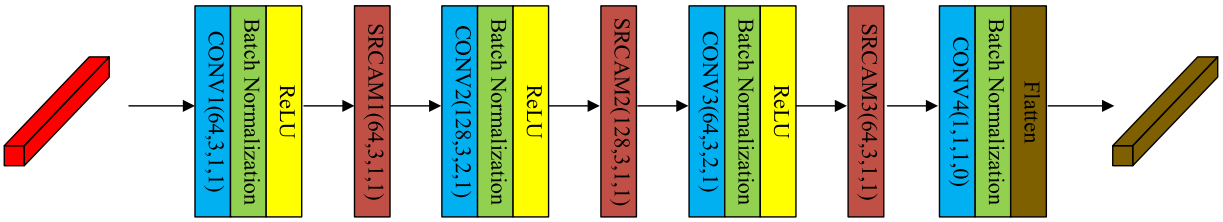

Fig. 4. Schematic of the backbone structure.

The spectral contrastive head is a two-layer nonlinear multilayer perceptron (MLP), denoted as $g_S(\cdot)$. The number of neurons in each layer of the MLP is 16, and its structure is shown in the spectral contrastive head section of Fig. 1. Since no prior label information available, it is impossible to construct positive sample and negative sample pairs based on the label information. The positive sample pairs are constructed based on the augmented samples obtained by data augmentation at the same location, and the negative sample pairs are constructed between the augmented samples obtained at different locations. Formally, a batch containing N pixel spectra is randomly sampled from the HSI $\mathbf{X}$ to be detected, and the designed data augmentation method is performed on each pixel spectrum $\mathbf{x}_i$ to obtain $2N$ data augmentation samples $\{\mathbf{d}_1^a, \ldots, \mathbf{d}_N^a, \mathbf{d}_1^b, \ldots, \mathbf{d}_N^b\}$. For an augmented sample $\mathbf{d}_i^a$ of a specific pixel spectrum $\mathbf{x}_i$, $2N-1$ pairs can be formed between it and the batch of augmented samples, where this augmented sample $\mathbf{d}_i^a$ forms the positive sample pair $\{\mathbf{d}_i^a, \mathbf{d}_i^b\}$ with another augmented sample $\mathbf{d}_i^b$ of the specific pixel spectrum x_i, and the negative sample pairs are formed with the remaining $2N-2$ samples. The backbone extracted representation was mapped to the spectral contrast loss space by $\mathbf{z}_i^a = g_s(\mathbf{h}_i^a)$ using the spectral contrastive head $g_S(\cdot)$. The similarity between sample pairs is measured by the cosine distance, which can be expressed as follows:

$$s\left(\mathbf{z}_i^{c_1}, \mathbf{z}_j^{c_2}\right) = \frac{\left(\mathbf{z}_i^{c_1}\right)\left(\mathbf{z}_j^{c_2}\right)^T}{\left\|\mathbf{z}_i^{c_1}\right\|\left\|\mathbf{z}_j^{c_2}\right\|} \tag{14}$$

where $c_1, c_2 \in \{a, b\}$ and $i, j \in [1, N]$. The spectral contrast loss for the given augmented sample $\mathbf{d}_i^a$ can be defined as follows:

$$l_i^a = -\log \frac{\exp\left(s\left(\mathbf{z}_i^a, \mathbf{z}_i^b\right)/\tau_s\right)}{\sum_{j=1}^{N}\left[\exp\left(s\left(\mathbf{z}_i^a, \mathbf{z}_j^a\right)/\tau_s\right) + \exp\left(s\left(\mathbf{z}_i^a, \mathbf{z}_j^b\right)/\tau_s\right)\right]} \tag{15}$$

where τ_s is the spectral temperature parameter to control the softness. To learn the similarity between all positive sample pairs from the spectral level, the spectral contrast loss is calculated on each augmented sample and can be formalized as follows:

$$L_{\text{spectral}} = \frac{1}{2N}\sum_{i=1}^{N}\left(l_i^a + l_i^b\right). \tag{16}$$

The spectral-level contrastive learning starts from the perspective of discriminating spectral similarities and differences, so that spectral similarity and dissimilarity can be distinguished by the representative vectors extracted by the backbone. The training process of the spectral-level contrastive learning phase is shown in Algorithm 2.

Algorithm 2 Contrastive Training Procedure

Input: augmentation samples $\mathbf{D}^a \in \mathbb{R}^{(H\times W)\times L}$ and $\mathbf{D}^b \in \mathbb{R}^{(H\times W)\times L}$, epoch E, batch size N, learning rate r, temperature parameter τ_s and τ_c, backbone $f(\cdot)$, spectrum contrastive head $g_s(\cdot)$.
Output: backbone $f(\cdot)$.
1. for epoch $= 1$ to E do
2. take N samples from the same position in $\mathbf{D}^a \in \mathbb{R}^{(H\times W)\times L}$ and $\mathbf{D}^b \in \mathbb{R}^{(H\times W)\times L}$ respectively, expressed as:
$\left[\mathbf{d}_1^a, \ldots, \mathbf{d}_N^a\right] \in \mathbb{R}^{N\times L}$ and $\left[\mathbf{d}_1^b, \ldots, \mathbf{d}_N^b\right] \in \mathbb{R}^{N\times L}$
3. for $\left\{\mathbf{d}_i^a\right\}_{i=1}^N$ and $\left\{\mathbf{d}_i^b\right\}_{i=1}^N$ do
4. extract representative vectors through backbone:
$\mathbf{h}_i^a = f\left(\mathbf{d}_i^a\right)$, $\mathbf{h}_i^b = f\left(\mathbf{d}_i^b\right)$.
5. compute spectrum contrastive head's output by
$\mathbf{z}_i^a = g_s\left(\mathbf{h}_i^a\right)$, $\mathbf{z}_i^b = g_s\left(\mathbf{h}_i^b\right)$.
6. end
8. compute spectrum contrastive loss $L_{spectral}$ through Eq. 14-16.
9. update $f(\cdot)$, $g_s(\cdot)$ to minimize $L_{spectral}$.
10. end

E. Spectral–Spatial Target Detection

After completing the spectral-level contrastive training, the representative vectors of each pixel spectrum in the HSI to be detected $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots \mathbf{x}_{H\times W}] \in \mathbb{R}^{(H\times W)\times B}$ and the prior target spectrum $\mathbf{x}_*$ are extracted separately using the backbone, and the similarity between the pixel spectrum to be detected and the prior target spectrum is measured by cosine similarity, which can be formalized as follows:

$$u_i = \frac{f(\mathbf{x}_i)\cdot f(\mathbf{x}_*)^T}{\|f(\mathbf{x}_i)\|\cdot\|f(\mathbf{x}_*)\|} \tag{17}$$

where u_i is the similarity score between the pixel spectrum to be detected and the prior target spectrum. The detection result $\mathbf{U} = [u_1, u_2, \ldots u_{H\times W}]$ is obtained after calculating cosine similarity of each pixel spectrum to be detected with the prior target spectrum.

Due to the low spatial resolution of HSIs, the pixels on the edge of the target may be the pixels where the target spectra are mixed with the surrounding background spectra, and these mixed pixels may be incorrectly identified as the background by relying solely on the spectral information.

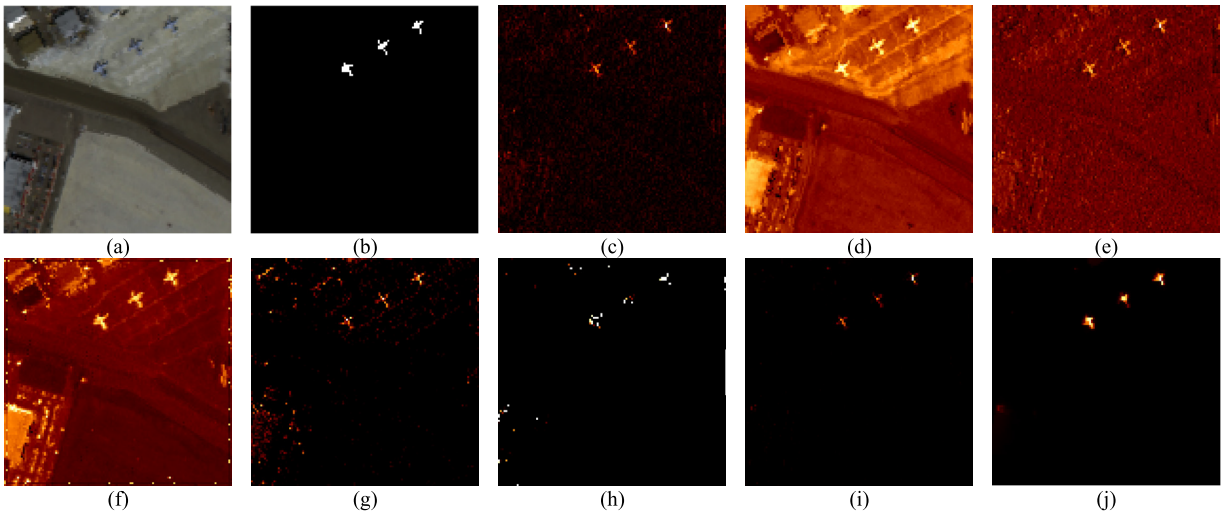

Fig. 5. Detection maps for the AVIRIS1 dataset. (a) Pseudo-color image. (b) Ground truth. (c) CEM. (d) OSP. (e) hCEM. (f) CSCR. (g) DM-BDL. (h) CNND. (i) BLTSC. (j) SCLHTD.

Therefore, to preserve the edges of the target and suppress the background, a 2-D transform domain recursive filter is used to process the above-obtained detection result $\mathbf{U}$. The 1-D transform domain recursive filtering [30] can be expressed as follows:

$$q_i = \left(1 - s^t\right)u_i + s^t q_{i-1} \tag{18}$$

where q_i is the result after filtering, s is the feedback coefficient, and t is the distance between adjacent pixels in the transform domain. s is calculated by the following formula:

$$s = \exp\left(\frac{-\sqrt{2}}{\delta_s}\right) \tag{19}$$

where $s = [0, 1]$. t can be calculated by the following procedure:

$$v_i = g_0 + \sum_{j=1}^{i}\left(1 + \frac{\delta_s}{\delta_r}\left|g_j - g_{j-1}\right|\right) \tag{20}$$

$$t = v_i - v_{i-1} \tag{21}$$

where δ_s and δ_r are two additional parameters to adjust the amount of smoothness in filtering and g_i represents the value of the ith pixel in the guide image.

The 2-D transform domain recursive filter is implemented by performing a sequence of horizontal and vertical 1-D transform domain recursive filtering on the spectral detection map $\mathbf{U}$. After several iterations, the final filtered output result is obtained, marked as the final spectral–spatial joint detection result $\mathbf{Q} = [q_1, q_2, \ldots, q_{H\times W}]$.

III. Experimental Results and Analysis

A. Hyperspectral Dataset

1) San Diego Dataset: The San Diego dataset was collected by the Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) at the San Diego Airport area CA, USA. The San Diego dataset has 400 × 400 pixels with a spatial resolution of 3.5 m and a spectral resolution of 10 nm, with 224 bands and a wavelength range of 370–2510 nm. In the experiments, two regions of size 120 × 120 and 100 × 100 were cropped from the San Diego dataset, named AVIRIS1 and AVIRIS2, respectively. After removing low SNR and water absorption bands (1–6, 33–35, 97, 107–113, 153–166, and 221–224), 189 bands were reserved for target detection. The airplanes in the AVIRIS1 and AVIRIS2 scenes are considered as targets for detection, with 58 target pixels in AVIRIS1 and 134 target pixels in AVIRIS2. The pseudo-color images of AVIRIS1 and AVIRIS2 with ground truth are shown in Figs. 5(a) and (b) and 6(a) and (b), respectively.

2) Urban Dataset: The Urban dataset was captured by AVIRIS sensors off the coast of TX, USA, with a spatial resolution of 17.2 m per pixel. The Urban dataset has 100 × 100 pixels, and after removing the low signal-to-noise band the remaining 204 bands, a total of 67 pixels are considered as targets for detection. The pseudo-color image and ground truth are shown in Fig. 7(a) and (b).

3) MUUFL Gulfport Dataset: The MUUFL Gulfport dataset [31], [32] was collected at the University of Southern Mississippi Gulf Park Campus Long Beach, MS, USA. After removing the noise bands (1–4 and 69–72) and the invalid region, it has a size of 325 × 220 × 64. The cloth panel in the scene is regarded as the target, and there are 269 target pixels for HTD. The pseudo-color image and ground truth are shown in Fig. 8(a) and (b).

B. Experimental Details

1) Evaluation Criteria: To evaluate the performance of the proposed SCLHTD method, the receiver operating characteristic (ROC) curve and the area under the curve (AUC) were used for quantitative analysis. The ROC curves have been widely used as an evaluation tool for target detection in HSI [33]. The ROC curve is obtained by changing the threshold τ to obtain different detection probabilities P_D and false alarm probabilities P_F. The detection probability P_D and the false alarm probability P_F can be calculated by the

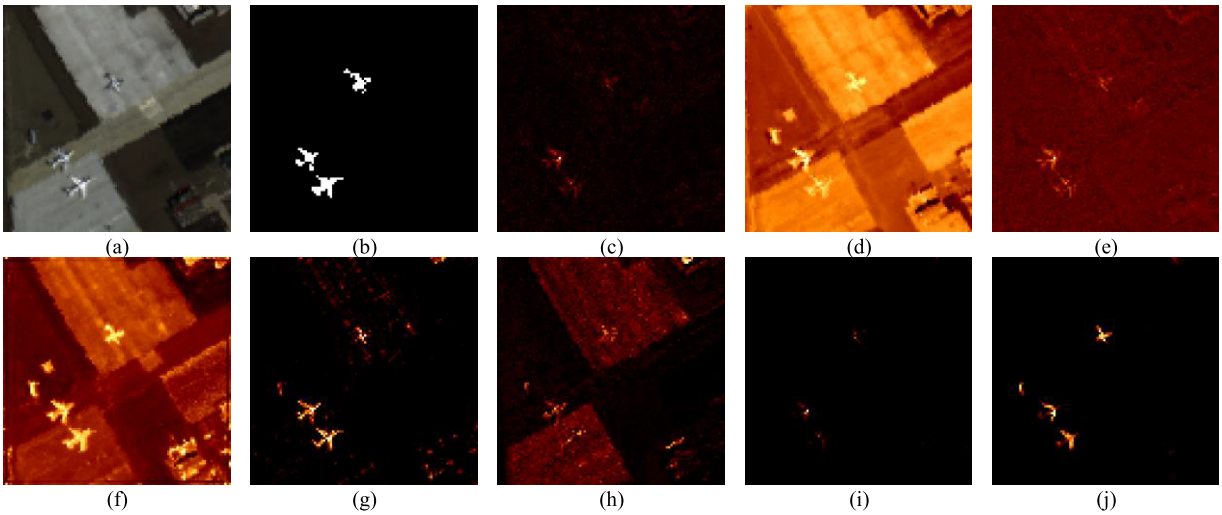

Fig. 6. Detection maps for the AVIRIS2 dataset. (a) Pseudo-color image. (b) Ground truth. (c) CEM. (d) OSP. (e) hCEM. (f) CSCR. (g) DM-BDL. (h) CNND. (i) BLTSC. (j) SCLHTD.

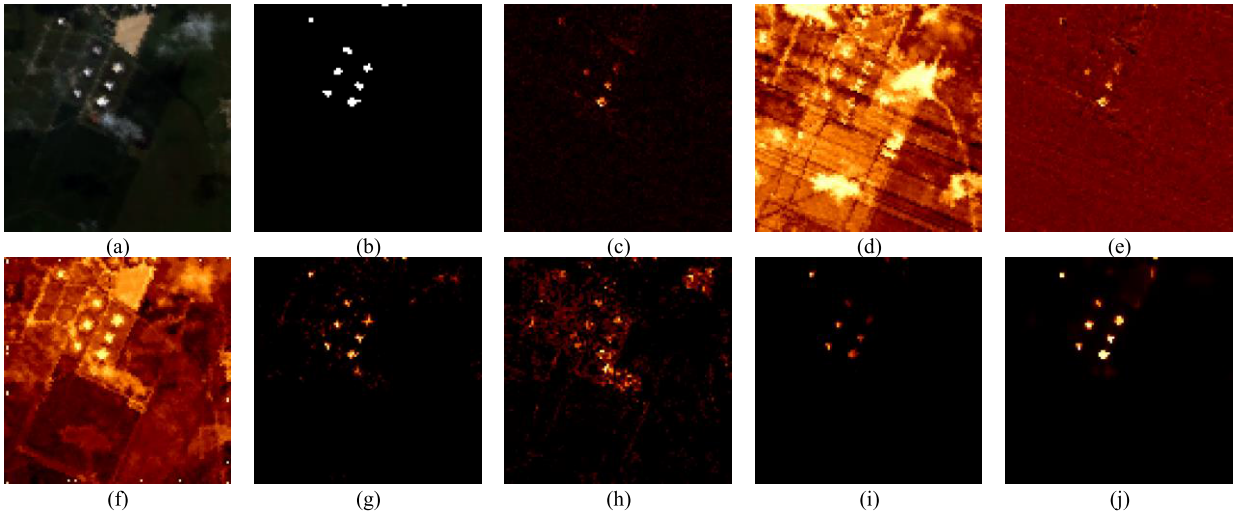

Fig. 7. Detection maps for the Urban dataset. (a) Pseudo-color image. (b) Ground truth. (c) CEM. (d) OSP. (e) hCEM. (f) CSCR. (g) DM-BDL. (h) CNND. (i) BLTSC. (j) SCLHTD.

following procedure:

$$P_{\mathrm{D}}(\tau) = \frac{n_{\mathrm{D},\tau}}{n_{\mathrm{D},\tau} + n_{\mathrm{FN},\tau}} \tag{22}$$

$$P_{\mathrm{F}}(\eta) = \frac{n_{\mathrm{F},\tau}}{n_{\mathrm{F},\tau} + n_{\mathrm{TN},\tau}} \tag{23}$$

where $n_{\mathrm{D},\tau}$, $n_{\mathrm{FN},\tau}$, $n_{\mathrm{F},\tau}$, and $n_{\mathrm{TN},\tau}$ denote the number of correctly detected target pixels, the number of pixels that are indeed targets but not detected as targets, the number of false detections of background pixels as target pixels, and the number of correctly detected background pixels under the threshold τ, respectively.

Due to the interaction between the detection probability P_{D} and the false alarm probability P_{F}, a high AUC value of the ROC curve of (P_{D}, P_{F}) does not necessarily mean that the detector has a high target detection probability or good background suppression capability [33]. To evaluate the detector performance more precisely, 3-D ROC curves are used, and three 2-D ROC curves of (P_{D}, P_{F}), (P_{D}, τ), and (P_{F}, τ) are generated to evaluate the detector effectiveness, target detection ability, and background suppression ability, respectively.

The AUC value is the area under the ROC curve and is used to quantitatively evaluate the performance of the detector. For 2-D ROC curves of (P_{D}, P_{F}), the value of AUC (P_{D}, P_{F}) between 1 and 0.5 indicates that the detector is effective, and the closer the AUC (P_{D}, P_{F}) is to 1, the better the performance of the detector. The value of AUC (P_{D}, τ) is the area under 2-D ROC curve of (P_{D}, τ), which can quantitatively measure the target detection ability of the detector. The value of AUC (P_{F}, τ) is the area under the 2-D ROC curve (P_{F}, τ), which can measure the ability of the detector to suppress the background. In general, the smaller the AUC (P_{F}, τ), the better the detector suppresses the background. A new quantitative detection metric designed in [33] is also used to consider the three AUC values as a whole to measure the performance of the target detection method, which was defined as follows:

$$\mathrm{AUC}_{\mathrm{OD}} = \mathrm{AUC}(P_{\mathrm{D}}, P_{\mathrm{F}}) + \mathrm{AUC}(P_{\mathrm{D}}, \tau) - \mathrm{AUC}(P_{\mathrm{F}}, \tau) \tag{24}$$

where $\mathrm{AUC}_{\mathrm{OD}} \in [-1, 2]$.

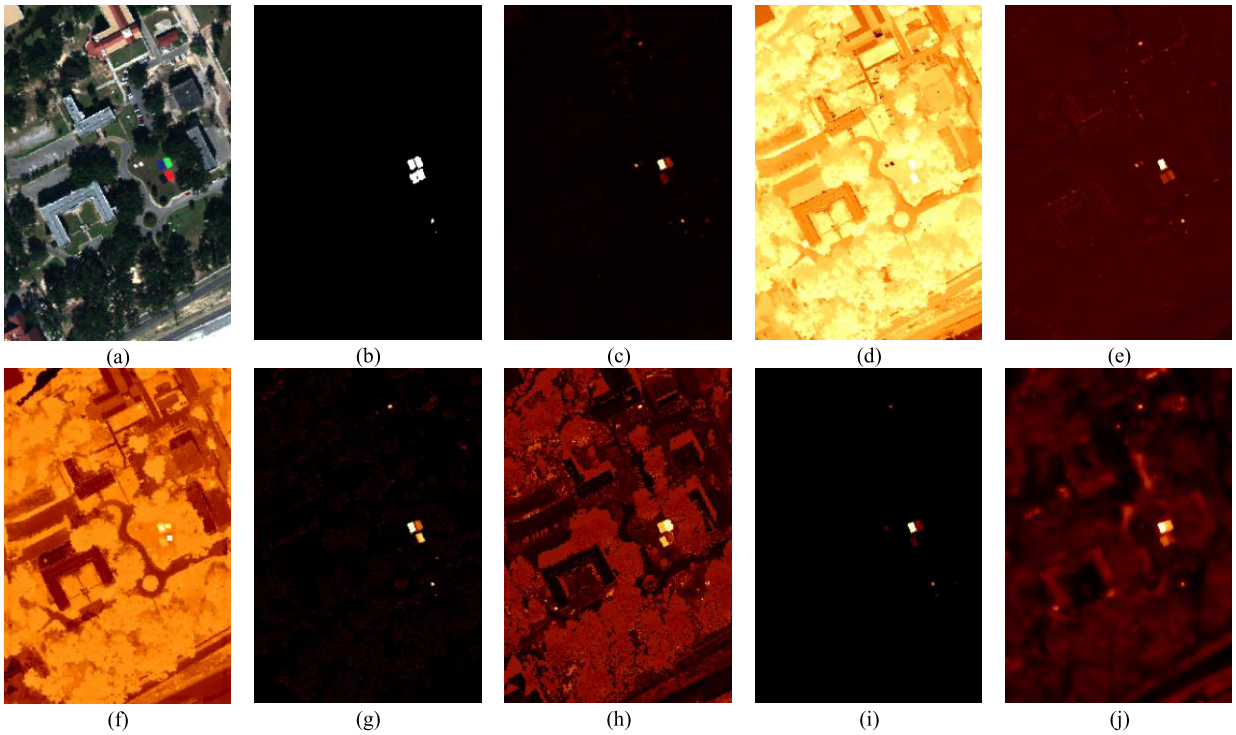

Fig. 8. Detection maps for the MUUFL Gulfport dataset. (a) Pseudo-color image. (b) Ground truth. (c) CEM. (d) OSP. (e) hCEM. (f) CSCR. (g) DM-BDL. (h) CNND. (i) BLTSC. (j) SCLHTD.

2) Experimental Setup: The proposed SCLHTD is implemented in three steps, namely data augmentation, spectral-level contrastive learning and spectral–spatial target detection, respectively. For data augmentation, band sampling is first performed on the HSI to be detected to obtain HSIs consisting of odd and even bands, respectively. The AAE is then trained separately using HSIs consisting of odd and even bands. For the four real HSI datasets, when training the AAE, the encoder and decoder are first optimized by the Adam optimizer, and the learning rate is set to $1e-3$ for both, and then the generator and discriminator are optimized by the SGD, and the learning rate is set to $1e-4$ when optimizing the generator and $1e-5$ when optimizing the discriminator. The AAE was trained with 20 epochs. The batch sizes of AVIRIS1, AVIRIS2, Urban and MUUFL Gulfport datasets are set to 240, 200, 200 and 500, respectively. The dimension of the encoder output latent encoding of AAE is 32, and the dimension of the feature vector extracted by the feature extraction part of the encoder is fixed to 64. In spectral-level contrastive learning, epoch, learning rate and temperature parameter are all set to 100, 0.05 and 0.1 for the four HSI datasets. The batch size is set to 240, 200, 200 and 500 for AVIRIS1, AVIRIS2, Urban and MUUFL Gulfport datasets respectively. For the four HSI datasets, the parameter δ_s used to control the window size of the filter, the parameter δ_r used to control the ambiguity of the filter and the number of iterations performed in the 2-D transform domain recursive filter are set to (5, 2, 3), (5, 0.5, 3), (5, 2, 3) and (5, 4, 3), respectively.

To evaluate the performance of the proposed SCLHTD method in the experiments, the following detection methods are compared with the proposed SCLHTD method: the classical detection method CEM [6], the subspace model-based detection method OSP [9], the improved method hCEM [7] for CEM, the representation-based target detectors CSCR [11] and decomposition model with background dictionary learning (DM-BDL) [34], and two deep learning-based methods—the CNN-based method [convolution neural network based detector (CNND)] [16] and the BLTSC [21]. CEM do not have any parameters that need to be set artificially. For the hCEM detector, two parameters λ and ε that need to be adjusted are set to 200 and 0.01 for all datasets in the experiment. For the CSCR detector, the outer and inner windows sizes are (7, 3), (7, 5), (9, 5), and (33, 15) for the AVIRIS1, AVIRIS2, Urban, and MUUFL Gulfport datasets, respectively. The regularization parameters λ_1 and λ_2 are set to 10^{-1} and 10^{-2} for all datasets in the experiment. The decay parameter in the DM-BDL detector was set to 0.982 for all datasets in the experiment, and the other parameters followed the settings in the original literature. For the transfer learning-based CNND detection method, the training set is constructed by subtracting the spectra of similar pixels and subtracting the spectra of different classes of pixels using the Salinas and MUUFL Gulfport dataset with known labels captured by the corresponding sensor when training the deep CNN. For all datasets in the experiment, the learning rate, batch size, and epoch of the CNND method during training are set to 10^{-3}, 256, and 50, respectively. For BLTSC, coarse detection is performed using the classical CEM method with a learning rate and epoch set to $1e-4$ and 500 during training for the four HSI datasets in the experiment, respectively.

The experimental hardware environment consists of an Intel Core i7-10875h eight-core CPU and an NVIDIA GeForce RTX 2080 graphics card. In terms of software environment, CNND based on transfer learning and the proposed SCLHTD

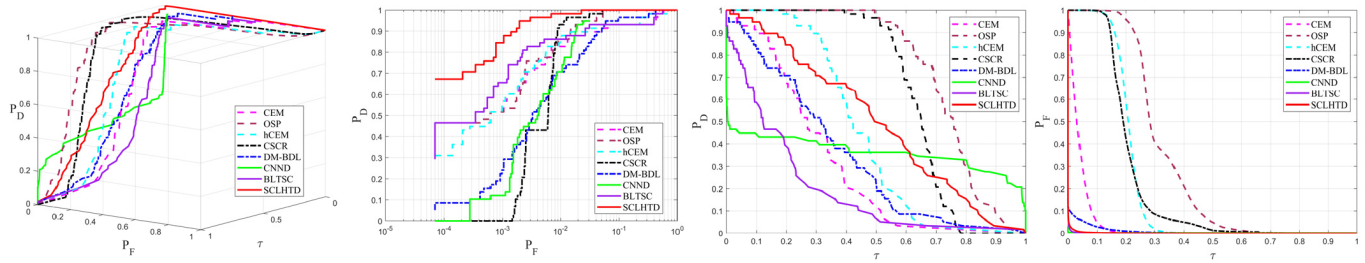

Fig. 9. ROC curves comparison on AVIRIS1. (a) 3-D ROC curve. (b) 2-D ROC curve of (P_D, P_F). (c) 2-D ROC curve of (P_D, τ). (d) 2-D ROC curve of (P_F, τ).

method are implemented using Python 3.8.3 and PyTorch 1.60. The BLTSC method uses Python 3.6 and TensorFlow 1.80. CEM, OSP, hCEM, CSCR, and DM-BDL detection methods are implemented using MATLABR 2017b.

C. Results and Analysis

1) Compared With State-of-the-Art Detection Methods: For performance evaluation of the proposed SCLHTD method, seven different state-of-the-art detection methods are used for comparison, which are the classical detection method CEM, the subspace model-based detection method OSP, the improved method hCEM for CEM, the representation-based target detectors CSCR and DM-BDL, and the two deep learning-based detectors CNND and BLTSC. Figs. 5–8 show the detection maps by the above eight methods for the AVIRIS1, AVIRIS2, Urban and MUUFL Gulfport datasets.

It can be seen from the detection maps that CEM, hCEM, CNND, and BLTSC miss many target pixels, and they have very low tolerance for target spectral variations. CEM is designed based on the constrained least squares-based regression method. However, hyperspectral data in real scenes exhibit usually show strong non-Gaussianity and nonlinearity, leading to a decrease in target detection accuracy. The hCEM suppresses the background and preserves the target through a layer-by-layer filtering process, but does not perform stably in practice when CEM detection is not good. CNND expands the training samples for training deep CNN by pairing pixels of the same class with pixels of the same class and pairing pixels of different classes based on known label information from known labeled HSIs of the corresponding sensor type, which enables the deep CNN to learn spectral difference discrimination ability for target detection. Since the spectral pairing is performed by pixel spectral subtraction, it leads to the loss of detailed spectral information of the original HSI, and the transfer knowledge is not so well adapted to the detection task in the target domain, which makes many target pixels are not detected. BLTSC used CEM to perform a coarse detection of the HSI to be detected and found reliable background samples for training AAE. After reconstructing the original HSI using the trained AAE, the background of the reconstructed HSI was reconstructed relatively accurately, and the target was reconstructed poorly. The difference between the reconstructed and original HSI was considered the target. The detection performance of BLTSC will be affected when CEM is not good enough to detect HSI. OSP and CSCR can detect the most of targets, but there is poor background suppression and small separation between target and background, resulting in the inability to visually identify targets, and the detection performance decreases when the background of the detection scene becomes complex. The target in the detection map of DM-BDL is primarily detectable and has good background suppression, but it requires more prior target spectra. The proposed SCLHTD shows excellent detection performance using one prior target spectrum with high target detection accuracy, good background suppression, and visually obvious identification of the target in the detection maps obtained on the four real HSI datasets.

Subjective evaluation of the detection maps visually has limitations, and to quantitatively evaluate the performance of the SCLHTD detector, 3-D ROC curves and their corresponding 2-D ROC curves (P_D, P_F), (P_D, τ), and (P_F, τ) with the AUCs of (P_D, P_F), (P_D, τ), and (P_F, τ) are used for quantitative evaluation. The 2-D ROC curve of (P_D, P_F) is used to demonstrate the effectiveness of detectors, as shown in Figs. 9(b)–12(b). For the four real HSI datasets in the experiment, the red curve is the ROC curve of the proposed SCLHTD, which outperforms the curves of other detectors. The 2-D ROC curve of (P_D, τ) is used to evaluate the preservation ability of the detector for the target, as shown in Figs. 9(c)–12(c). SCLHTD outperforms CEM and BLTSC, but OSP performs better than SCLHTD. However, for the 2-D ROC curve of (P_F, τ), which evaluates the detector background suppression ability, SCLHTD has significantly better background suppression ability than CSCR and OSP, and SCLHTD shows a strong background suppression ability on AVIRIS1, AVIRIS2, and Urban datasets.

The specific values of AUC (P_D, P_F), AUC (P_D, τ), AUC (P_F, τ), and AUC_{OD} for different detectors on the AVIRIS1, AVIRIS2, Urban, and MUUFL Gulfport datasets are given in Tables I–IV.

The optimal results are shown in bold, and the suboptimal results are underlined. As can be seen from the tables, BLTSC performs the best in background suppression but the worst in target preservation. OSP and CSCR perform good in target preservation, and most of the target pixels can be detected, but its background suppression ability is much weaker than SCLHTD. The AUC (P_D, P_F) values of the proposed SCLHTD remain optimal on the HSI datasets in the experiment, and the AUC (P_F, τ) values remain suboptimal and only weakly

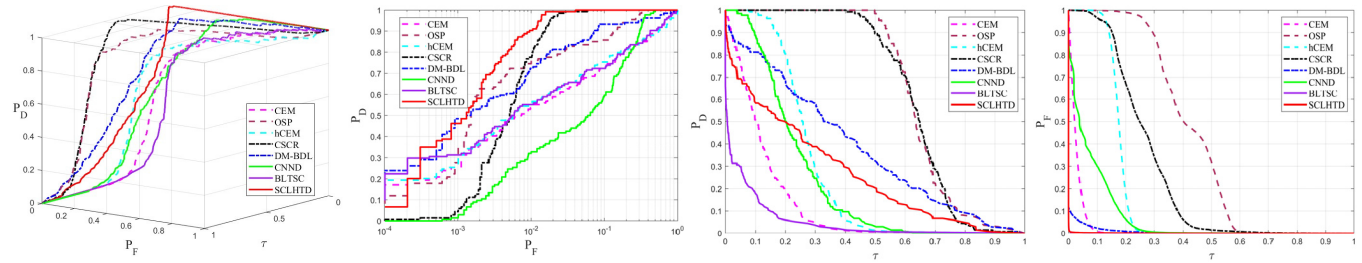

Fig. 10. ROC curves comparison on AVIRIS2. (a) 3-D ROC curve. (b) 2-D ROC curve of (P_D, P_F). (c) 2-D ROC curve of (P_D, τ). (d) 2-D ROC curve of (P_F, τ).

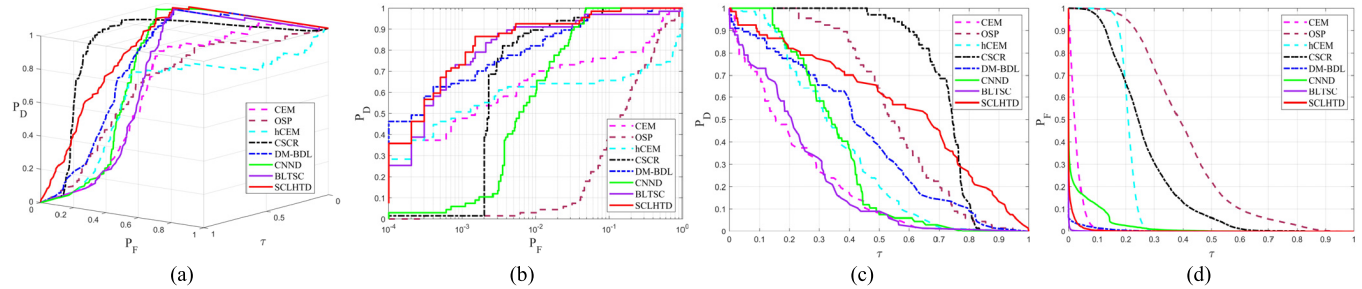

Fig. 11. ROC curves comparison on Urban. (a) 3-D ROC curve. (b) 2-D ROC curve of (P_D, P_F). (c) 2-D ROC curve of (P_D, τ). (d) 2-D ROC curve of (P_F, τ).

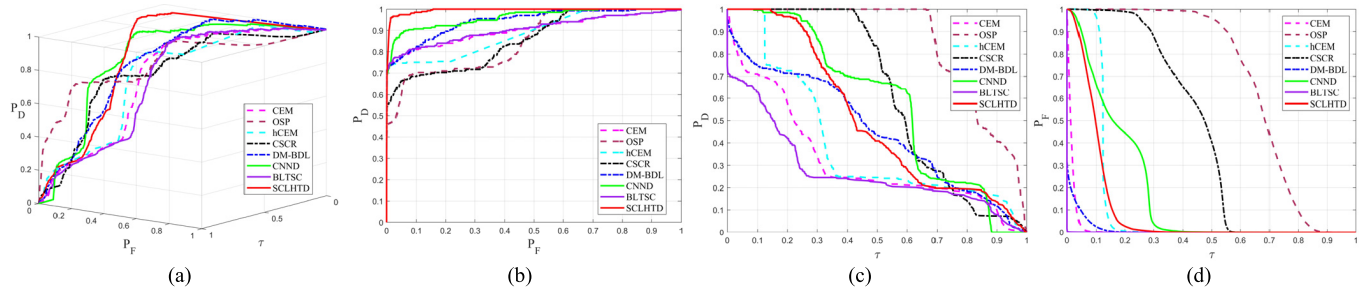

Fig. 12. ROC curves comparison on MUUFL. (a) 3-D ROC curve. (b) 2-D ROC curve of (P_D, P_F). (c) 2-D ROC curve of (P_D, τ). (d) 2-D ROC curve of (P_F, τ).

TABLE I

ACCURACY COMPARISON OF DIFFERENT METHODS FOR THE AVIRIS1 DATASET. BOLDFACE HIGHLIGHTS THE BEST RESULT, WHILE UNDERLINE THE SECOND

Method	CEM	OSP	hCEM	CSCR	DM-BDL	CNND	BLTSC	SCLHTD
$AUC_{(P_D,P_F)}$	0.9629	<u>0.9948</u>	0.9706	0.9937	0.9759	0.9675	0.9669	**0.9992**
$AUC_{(P_D,\tau)}$	0.2973	**0.7402**	0.4419	<u>0.6409</u>	0.3306	0.3621	0.1914	0.4979
$AUC_{(P_F,\tau)}$	0.0385	0.3205	0.2091	0.2113	0.0087	0.0018	**0.0006**	<u>0.0014</u>
AUC_{OD}	1.2217	<u>1.4145</u>	1.2034	1.4233	1.2978	1.3278	1.1577	**1.4957**

inferior to BLTSC on AVIRIS1 and AVIRIS2 datasets. AUC (P_D, τ) values remain suboptimal on the Urban datasets. The proposed SCLHTD method is much higher than the comparison methods for the AUC_{OD} values that exhibit comprehensive detection ability on the AVIRIS1, Urban, and MUUFL Gulfport datasets, only weakly with CSCR and DM-BDL on the AVIRIS2 dataset.

To evaluate the effectiveness of SCLHTD in separating target from background, the target–background separability boxplot [35] is used to show the separation degree of target and background. Fig. 13 shows the target–background separability boxplot for the seven compared methods and the proposed SCLHTD method on the four real HSI datasets. The boxes in the target–background separability boxplot represent pixels with statistically distributed values, removing the highest and lowest 10% of data in the target and background. The red box and green box represent the target and background, respectively. The horizontal line in the middle of each box

TABLE II
ACCURACY COMPARISON OF DIFFERENT METHODS FOR THE AVIRIS2 DATASET. BOLDFACE HIGHLIGHTS THE BEST RESULT, WHILE UNDERLINE THE SECOND

Method	CEM	OSP	hCEM	CSCR	DM-BDL	CNND	BLTSC	SCLHTD
$AUC_{(P_D,P_F)}$	0.8725	0.9665	0.8711	0.9931	0.9545	0.8895	0.8802	**0.9969**
$AUC_{(P_D,\tau)}$	0.1224	**0.6537**	0.2691	0.6423	0.3767	0.2276	0.0505	0.2625
$AUC_{(P_F,\tau)}$	0.0264	0.4192	0.1753	0.2602	0.0074	0.0691	**0.0002**	0.0007
AUC_{OD}	0.9685	1.2010	0.9649	**1.3752**	1.3238	1.0480	0.9305	1.2587

TABLE III
ACCURACY COMPARISON OF DIFFERENT METHODS FOR THE URBAN DATASET. BOLDFACE HIGHLIGHTS THE BEST RESULT, WHILE UNDERLINE THE SECOND

Method	CEM	OSP	hCEM	CSCR	DM-BDL	CNND	BLTSC	SCLHTD
$AUC_{(P_D,P_F)}$	0.8941	0.7647	0.7439	0.9933	0.9840	0.9873	0.9757	**0.9943**
$AUC_{(P_D,\tau)}$	0.2142	0.5553	0.3502	**0.7200**	0.4175	0.3428	0.2229	0.5703
$AUC_{(P_F,\tau)}$	0.0264	0.4056	0.2090	0.2636	0.0044	0.0265	**0.0006**	0.0046
AUC_{OD}	1.0819	0.9144	0.8851	1.4497	1.3971	1.3036	1.1980	**1.5600**

TABLE IV
ACCURACY COMPARISON OF DIFFERENT METHODS FOR THE MUUFL GULFPORT DATASET. BOLDFACE HIGHLIGHTS THE BEST RESULT, WHILE UNDERLINE THE SECOND

Method	CEM	OSP	hCEM	CSCR	DM-BDL	CNND	BLTSC	SCLHTD
$AUC_{(P_D,P_F)}$	0.9027	0.8538	0.8878	0.8576	0.9445	0.9613	0.9069	**0.9947**
$AUC_{(P_D,\tau)}$	0.3192	**0.8503**	0.3974	0.6224	0.4566	0.5748	0.2707	0.5058
$AUC_{(P_F,\tau)}$	0.0162	0.6780	0.1241	0.4378	0.0119	0.1728	**0.0002**	0.0986
AUC_{OD}	1.2057	1.0261	1.1611	1.0422	1.3892	1.3633	1.1774	**1.4019**

TABLE V
EFFECT OF THE SRCAM ON DETECTION ACCURACY ON FOUR DATASETS

SCLHTD / $AUC(P_D,P_F)$	AVIRIS1	AVIRIS2	Urban	MUUFL Gulfport
Without SRCAM	0.99661	0.99464	0.99227	0.90278
With SRCAM	**0.99727**	**0.99571**	**0.99384**	**0.95437**

TABLE VI
EFFECT OF THE DESIGNED DATA AUGMENTATION METHODS ON DETECTION ACCURACY ON FOUR DATASETS

SCLHTD / $AUC(P_D,P_F)$	AVIRIS1	AVIRIS2	Urban	MUUFL Gulfport
Without Data Augmentation	0.88573	0.78386	0.84931	0.83123
With Data Augmentation	**0.99727**	**0.99571**	**0.99384**	**0.95437**

indicates the median value, and the upper and lower horizontal lines indicate the maximum and minimum values. SCLHTD displays good background suppression performance for the HSI datasets in the experiment and can better separate the target from the background. The excellent target–background separability indicates that spectral-level contrastive learning enables the model to effectively learn the ability to discriminate spectral differences.

2) Module Ablation Experiments of SRCAM: To investigate the effect of SRCAM on the detection accuracy of HTD, the SRCAM is removed from the backbone and adversarial convolutional autoencoder for data augmentation to demonstrate the effect of SRCAM on HTD accuracy. Table V illustrates the effect of SRCAM on the detection accuracy of HTD. The AUC (P_D, P_F) values in Table V are a direct measure of the similarity between the representation of the pixel spectrum to be detected and the representation of the prior target spectrum through the cosine similarity. The detection accuracy obtained by cosine similarity intuitively reflects the impact of SRCAM on the model detection accuracy. It can be seen from Table V that the detection accuracy of the model with the SRCAM module is higher than that without SRCAM on all four real HSI datasets.

3) Module Ablation Experiments of Data Augmentation: An ablation study of data augmentation was conducted to verify whether the data augmentation module has any effect on the detection accuracy of the model output. Table VI shows the accuracy of the detection results for the direct

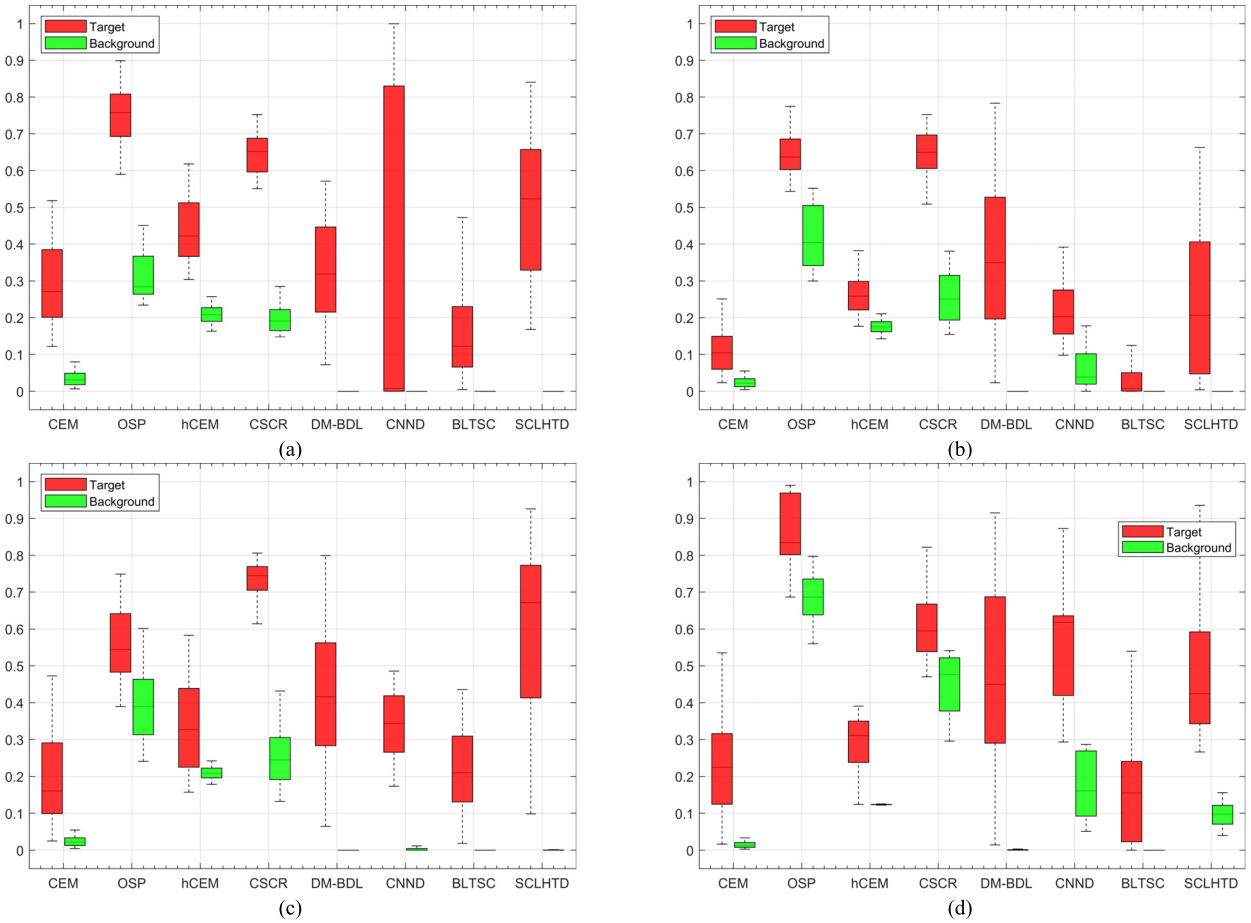

Fig. 13. Target-background separability boxplots for four HSI datasets. (a) AVIRIS1. (b) AVIRIS2. (c) Urban. (d) MUUFL Gulfport.

TABLE VII
TIME CONSUMPTION OF DIFFERENT METHODS FOR DIFFERENT HYPERSPECTRAL DATASETS

Method		AVIRIS1	AVIRIS2	Urban	MUUFL Gulfport
CEM		0.0436	0.0279	0.0238	0.2974
OSP		0.4765	0.0804	0.6680	0.1785
hCEM		0.2180	0.1551	0.1581	0.2219
CSCR		4.1392	2.6419	3.9638	839.3378
DM-BDL		4.2762	3.5070	3.4891	17.4299
CNND	Train	353.2291	350.2004	351.9937	143.5464
	Detect	35.0636	24.9714	26.8009	170.8501
BLTSC	Train	1071.7673	695.5097	786.4740	3576.1699
	Detect	8.2916	8.1337	8.5614	6.4931
SCLHTD	Train	319.6209	247.3505	252.8787	1117.1488
	Detect	3.9193	3.3156	3.0419	5.1406

output of the model with and without data augmentation in the proposed method. As can be seen in Table VI, containing data augmentation can significantly improve the target detection accuracy of the SCLHTD.

4) Time Consumption: Table VII lists the time consumption of the seven compared methods and the proposed SCLHTD method. As can be seen in Table VII, the time consumptions of the classical HTD method and the machine learning-based HTD method are much less than those of the deep learning-based HTD method. This is reasonable since the deep learning-based methods need to be trained to obtain the parameters of the networks. Among three deep learning-based algorithms, the training time for BLTSC includes the time to find reliable background samples using coarse detection and the time to train the AAE using the background samples; the training time for SCLHTD includes the time for data augmentation and performing spectral-level contrastive learning, and the training time for CNND is independent of the size of the detection scene and is related to the size of the source domain dataset since it is transfer learning-based algorithm. In the experiments of CNND, the AVIRIS1, AVIRIS2 and Urban datasets are collected with AVIRIS sensors and the train data used are the Salinas data with the same sensor, while the MUUFL Gulfport is trained using HSI with smaller data samples of the same sensor as MUUFL, making it trains with less time consumption (we cannot find a large labeled dataset from the same sensor). As a result, the time consumption of CNND for MUUFL seems to be less than the proposed SCLHTD since the SCLHTD trains the model in a self-supervised manner by mining self-supervised information in the largescale MUUFL image with more pixels. In terms of training time of the deep learning-based HTD method, SCLHTD consumes less training time than BLTSC, and if the training dataset is of the same size, SCLHTD also consumes less training time than CNND and furthermore achieves better target detection accuracy than CNND and BLTSC. Once the model has been trained well, the

detective efficiency relies on the detection time. The detection time of the deep learning-based detection methods starts with loading the model and ends with the detection results, where as shown in Table VII, the detection time of the proposed SCLHTD is less than that of the other two deep learning-based methods (CNND and BLTSC) using the same HSI datasets.

IV. Conclusion

To liberate the HTD model from dependence on the quality of the priori information, a self-supervised spectral-level contrastive learning is proposed in this article. Data augmentation procedure is proposed to mine the supervision information of HSIs to be detected, and spectral-level contrastive learning is then designed to make the model with the capability of identifying the similarities and differences between spectra in a self-supervised manner. Specifically, an adversarial convolutional autoencoder with spectral residual channel attention mechanism is first designed for data augmentation, where the HSI to be detected is sampled into odd and even band subsets and sent to the corresponding adversarial convolutional autoencoders for training, respectively. The feature extraction part of the two trained adversarial convolutional autoencoders is regarded as the data augmentation function, and two kinds of data augmentation samples are obtained by using the corresponding data augmentation functions. Two augmented samples of pixels at the same position can be regarded as a positive sample pair, and the augmented samples of the pixels at different positions can be regarded as negative sample pairs. Second, in the stage of spectral-level contrastive learning, the backbone is used to extract the representative vectors of positive and negative sample pairs, and the representative vectors are mapped from the spectral contrastive head to the spectral contrast space to learn the similarities and dissimilarities between spectra, which has the ability to distinguish spectral similarity and dissimilarity to extract the representative vectors of the prior target spectrum and the pixel spectra to be detected, and then obtain the detection map using the spectral information by measuring the similarity through the cosine distance. Finally, the space information is combined, and the first three principal components of the HSI to be detected are used to perform edge-preserving filtering on the above detection map to suppress the background and obtain the final target detection results. The results of comprehensive experiments show that the proposed SCLHTD method using only one prior target spectrum outperforms other comparative detectors with more prior information.

References

[1] H. Huang et al., "Underwater hyperspectral imaging for in situ underwater microplastic detection," *Sci. Total Environ.*, vol. 776, Jul. 2021, Art. no. 145960.

[2] X. Sun, H. Zhang, F. Xu, Y. Zhu, and X. Fu, "Constrained-target band selection with subspace partition for hyperspectral target detection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 9147–9161, 2021.

[3] R. Fakhrullin, L. Nigamatzyanova, and G. Fakhrullina, "Dark-field/hyperspectral microscopy for detecting nanoscale particles in environmental nanotoxicology research," *Sci. Total Environ.*, vol. 772, Jun. 2021, Art. no. 145478.

[4] X. Zhao, Z. Hou, X. Wu, W. Li, P. Ma, and R. Tao, "Hyperspectral target detection based on transform domain adaptive constrained energy minimization," *Int. J. Appl. Earth Observ. Geoinf.*, vol. 103, Dec. 2021, Art. no. 102461.

[5] X. Jin, S. Paswaters, and H. Cline, "A comparative study of target detection algorithms for hyperspectral imagery," in *Proc. SPIE*, vol. 7334, pp. 682–693, Apr. 2009.

[6] C.-I. Chang and D. Heinz, "Constrained subpixel target detection for remotely sensed imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 38, no. 3, pp. 1144–1159, May 2000.

[7] Z. Zou and Z. Shi, "Hierarchical suppression method for hyperspectral target detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 1, pp. 330–342, Jan. 2016.

[8] R. Zhao, Z. Shi, Z. Zou, and Z. Zhang, "Ensemble-based cascaded constrained energy minimization for hyperspectral target detection," *Remote Sens.*, vol. 11, no. 11, p. 1310, Jun. 2019.

[9] Q. Du, H. Ren, and C.-I. Chang, "A comparative study for orthogonal subspace projection and constrained energy minimization," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 6, pp. 1525–1529, Jun. 2003.

[10] Y. Chen, N. M. Nasrabadi, and T. D. Tran, "Sparse representation for target detection in hyperspectral imagery," *IEEE J. Sel. Topics Signal Process.*, vol. 5, no. 3, pp. 629–640, Jun. 2011.

[11] W. Li, Q. Du, and B. Zhang, "Combined sparse and collaborative representation for hyperspectral target detection," *Pattern Recognit.*, vol. 48, no. 12, pp. 3904–3916, Dec. 2015.

[12] S. K. Roy, S. Manna, T. Song, and L. Bruzzone, "Attention-based adaptive spectral–spatial kernel ResNet for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7831–7843, Sep. 2021.

[13] W. Xie, J. Lei, J. Yang, Y. Li, Q. Du, and Z. Li, "Deep latent spectral representation learning-based hyperspectral band selection for target detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 58, no. 3, pp. 2015–2026, Mar. 2020.

[14] B. Palsson, M. O. Ulfarsson, and J. R. Sveinsson, "Convolutional autoencoder for spectral–spatial hyperspectral unmixing," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 1, pp. 535–549, Jan. 2020.

[15] W. Wei, J. Nie, Y. Li, L. Zhang, and Y. Zhang, "Deep recursive network for hyperspectral image super-resolution," *IEEE Trans. Comput. Imag.*, vol. 6, pp. 1233–1244, 2020.

[16] W. Li, G. Wu, and Q. Du, "Transferred deep learning for hyperspectral target detection," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Jul. 2017, pp. 5177–5180.

[17] Y. Shi, J. Li, Y. Li, and Q. Du, "Sensor-independent hyperspectral target detection with semisupervised domain adaptive few-shot learning," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 8, pp. 6894–6906, Oct. 2021.

[18] G. Zhang, S. Zhao, W. Li, Q. Du, Q. Ran, and R. Tao, "HTD-Net: A deep convolutional neural network for target detection in hyperspectral imagery," *Remote Sens.*, vol. 12, no. 9, p. 1489, May 2020.

[19] Y. Gao, Y. Feng, and X. Yu, "Hyperspectral target detection with an auxiliary generative adversarial network," *Remote Sens.*, vol. 13, no. 21, p. 4454, Nov. 2021.

[20] D. Zhu, B. Du, and L. Zhang, "Two-stream convolutional networks for hyperspectral target detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 8, pp. 6907–6921, Aug. 2021.

[21] W. Xie, X. Zhang, Y. Li, K. Wang, and Q. Du, "Background learning based on target suppression constraint for hyperspectral target detection," *IEEE J. Sel. Top. Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5887–5897, 2020.

[22] Y. Shi, J. Li, Y. Yin, B. Xi, and Y. Li, "Hyperspectral target detection with macro-micro feature extracted by 3-D residual autoencoder," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 12, pp. 4907–4919, Dec. 2019.

[23] Y. Shi, J. Li, Y. Zheng, B. Xi, and Y. Li, "Hyperspectral target detection with RoI feature transformation and multiscale spectral attention," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 5071–5084, Jun. 2021.

[24] B. Xi, J. Li, Y. Li, R. Song, and Q. Du, "Deep prototypical networks with hybrid residual attention for hyperspectral image classification," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 3683–3700, 2020.

[25] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Jun. 2020.

[26] B. Liu, A. Yu, X. Yu, R. Wang, K. Gao, and W. Guo, "Deep multiview learning for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 9, pp. 7758–7772, Sep. 2021.

[27] C.-I. Chang, "Band sampling for hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2021, Art. no. 5514024.
[28] Z. Cao, X. Li, Y. Feng, S. Chen, C. Xia, and L. Zhao, "ContrastNet: Unsupervised feature learning by autoencoder and prototypical contrastive learning for hyperspectral imagery classification," *Neurocomputing*, vol. 460, pp. 71–83, Oct. 2021.
[29] Y. Li, P. Hu, Z. Liu, D. Peng, J. T. Zhou, and X. Peng, "Contrastive clustering," in *Proc. AAAI Conf. Artif. Intell.*, 2021, pp. 8547–8555.
[30] X. Kang, X. Zhang, S. Li, K. Li, J. Li, and J. A. Benediktsson, "Hyperspectral anomaly detection with attribute and edge-preserving filters," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 10, pp. 5600–5611, Oct. 2017.
[31] A. Z. P. Gader, R. Close, J. Aitken, and G. Tuell, "Muufl gulfport hyperspectral and LiDAR airborne data set," Dept. Elect. Comput. Eng., Univ. Florida, Gainesville, FL, USA, Tech. Rep. REP-2013-570, 2013.
[32] X. D. A. A. Zare, "Technical report: Scene label ground truth map for MUUFL Gulfport data set," Dept. Elect. Comput. Eng., Univ. Florida, Gainesville, FL, USA, Tech. Rep. 20170417, 2017. [Online]. Available: http://ufdc.ufl.edu/IR00009711/00001
[33] C.-I. Chang, "An effective evaluation tool for hyperspectral target detection: 3D receiver operating characteristic curve analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 5131–5153, Jun. 2020.
[34] T. Cheng and B. Wang, "Decomposition model with background dictionary learning for hyperspectral target detection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 14, pp. 1872–1884, 2021.
[35] L. Zhang and B. Cheng, "Fractional Fourier transform and transferred CNN based on tensor for hyperspectral anomaly detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 19, pp. 1–5, 2022.

Yulei Wang (Member, IEEE) was born in Yantai, Shandong Province, China, in 1986. She received the B.S. and Ph.D. degrees in signal and information processing from Harbin Engineering University, Harbin, China, in 2009 and 2015, respectively.

She was a joint Ph.D. Student with the Remote Sensing Signal and Image Processing Laboratory, University of Maryland, College Park, MD, USA, from 2011 to 2013. She is currently an Associate Professor and a Doctoral Supervisor with the Center of Hyperspectral Imaging in Remote Sensing (CHIRS), Information Science and Technology College, Dalian Maritime University, Dalian, China. Her research interests include hyperspectral image processing and vital signs signal processing.

Xi Chen was born in Kuitun, Xinjiang Uygur Autonomous Region, China, in 2000. He received the B.E. degree in electronic information engineering from Dalian Maritime University, Dalian, China, in 2020, where he is currently pursuing the M.S. degree in information and communication engineering with the Information Science and Technology College.

His research interests include hyperspectral target detection and deep learning.

Enyu Zhao received the Ph.D. degree in cartography and geographic information system from the College of Resources and Environment, University of Chinese Academy of Sciences, Beijing, China, in 2017.

He was a joint Ph.D. Student with the Engineering Science, Computer Science and Imaging Laboratory, University of Strasbourg, Strasbourg, France, from 2014 to 2016. He is currently an Associate Professor with the College of Information Science and Technology, Dalian Maritime University, Dalian, China.

Meiping Song received the Ph.D. degree from the College of Computer Science and Technology, Harbin Engineering University, Harbin, China, in 2006.

From 2013 to 2014, she was a Visiting Associate Research Scholar with the Remote Sensing Signal and Image Processing Laboratory, University of Maryland, College Park, MD, USA. She is currently a Professor and a Doctoral Supervisor with the Center of Hyperspectral Imaging in Remote Sensing (CHIRS), Information Science and Technology College, Dalian Maritime University, Dalian, China. Her research interests include remote sensing and hyperspectral image processing.