# ABLAL: Adaptive Background Latent Space Adversarial Learning Algorithm for Hyperspectral Target Detection

Long Sun, Zongfang Ma, and Yi Zhang, *Member, IEEE*

***Abstract*—Hyperspectral images (HSIs) are challenging for hyperspectral object detection (HTD) due to their complex background information and the limited prior knowledge of the target. This article proposes an adaptive background latent space adversarial learning algorithm for hyperspectral target detection (ABLAL). We begin by using a coarse screening method to select pseudobackground and pseudotarget sample sets, addressing the issues caused by insufficient prior target information and complicated background information, which result in low detection accuracy. Next, we utilize an Adversarial Autoencoder (AAE) based backbone network to extract the background latent spatial information of the HSI. It should be noted that we adaptively constrain the accuracy of the extracted information through the pseudotarget dataset, accounting for the impact of potential targets in the pseudobackground dataset. Furthermore, we fully utilize the information extracted by AAE and employ a strategy combining multiple output results of AAE. Specifically, we use the distance between the target latent space vector and the background latent space vector, and the HSI reconstruction difference to suppress the background. Finally, extensive experiments are conducted on real datasets to demonstrate the effectiveness of the proposed method.**

***Index Terms*—Adversarial learning, hyperspectral image (HSI), latent space, target detection, weak supervised learning.**

## I. Introduction

HYPERSPECTRAL image (HSI) is a powerful technology that can simultaneously capture the morphological and spectral information of objects in a specific area. Morphological information highlights the external features of object samples, including size and shape. Meanwhile, different components absorb spectra uniquely, generating variations in reflections for different objects in the image at specific wavelengths. The spectral information therefore reflects the differences in the internal structure and composition of target samples. These distinct features make hyperspectral imaging technology applicable in various fields, including mineral detection [1], disease diagnosis [2], military reconnaissance [3], and environmental detection [4].

Hyperspectral target detection (HTD) is the process of classifying all image pixels in the HSI into either target or background classes. However, the target areas in HSI are usually limited to a small part of the region, with the remainder marked as "background." This means that the targets are sparse [5], and detecting them is the key to successful HTD in the presence of complex and diverse background categories in HSI.

HTD is achieved by utilizing the difference between the spectral signature of the targets and that of other land covers. For example, spectral angle mapper (SAM) [6]. Adaptive cosine estimator (ACE) [7], and matched filter (MF) [8] define the target detection task as a binary hypothesis test based on Gaussian distributions. Constrained energy minimization (CEM) [9] is used to design a linear filter that minimizes total energy while keeping the target output constant at 1, therefore, suppressing the background while detecting the target. Although many studies have been conducted on CEM and its variants, such as hCEM [9] and ECEM [10], their performance is limited due to the redundancy of spectral information and the limitations of imaging technology. In order to effectively tackle the intricate contextual issues in HSIs, Chen and Chang [11] adopted a comprehensive approach by integrating CEM and OSP techniques. As a result, they proposed a novel method called BKG-annihilated TCIMF method. In order to overcome the challenge of distinguishing targets and backgrounds in HSIs, which arises due to spectral variability and noise, researchers have introduced a technique known as DCEM. This technique transforms the linear solving process of CEM into a nonlinear one by utilizing a fully connected neural network at a pixel-level or a convolutional neural network at a cube-level. The primary objective of this transformation is to enhance detection performance [12].

The field of computer vision has experienced a surge in interest in HSI processing, due to the success of machine learning (ML) and deep learning (DL) algorithms. A range of ML methods have been utilized in HSI, including kernel-based learning, sparse and collaborative representation. Kernel OSP, a semisupervised method, can effectively resolve the linear inseparability problem by mapping the original space to the

Manuscript received 15 May 2023; revised 26 July 2023, 26 August 2023, and 29 September 2023; accepted 30 October 2023. Date of publication 3 November 2023; date of current version 23 November 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62276207, in part by the Technology Innovation Leading Program of Shaanxi under Grant 2023GXLH-055, in part by the China Postdoctoral Science Foundation under Grant 2018M643592, in part by the Natural Science Basic Research Plan in Shaanxi Province of China under Grant 2018JM5127, and in part by the Science and Technology Program Project of Ministry of Housing and Urban-Rural Development of China under Grant 2017-K2-014. *(Corresponding author: Yi Zhang.)*

The authors are with the College of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an 710311, China (e-mail: sunlongxauat@outlook.com; zongfangma@xauat.edu.cn; zhangyi@xauat.edu.cn).

Digital Object Identifier 10.1109/JSTARS.2023.3329771

kernel space [13]. However, the challenge of designing an appropriate kernel remains unresolved. Sparse and collaborative representation methods, such as combined sparse and collaborative representation (CSCR) [14], require labeled target information for dictionary construction and can have reduced performance if insufficient prior target information is available [15].

DL-based methods provide data-driven, nonlinear, and deep features to approximate complex functions, which has led to extensive research interest in various applications. For instance, within the domain of HSI classification, a network named DFINet, designed for depthwise feature interaction, is introduced in [16]. This network's objective is to extract autocorrelation and cross-correlation from pairs of features originating from multiple sources, utilizing deep cross-attention modules. This extraction process enhances significant complementary information, contributing to the classification of both hyperspectral and multispectral images. Gao et al. [17] introduced the concept of a transformer-based model named CSMFormer (cross-scale mixing attention transformer). This model is designed for the fusion and classification of multisource remote sensing data, exhibiting remarkable performance in both aspects. The FADCNN framework, a spatial-spectral dense CNN framework with a feedback attention mechanism, was proposed for HSI classification in the work by Yu et al.[18]. The SLA-NET method performs tree species classification by extracting fine-grained morphological structures from HSIs [19]. Yu et al. [20] introduced a multiperspective calibration prototype learning framework that enhances the robustness of prototypes in the embedding space through calibration strategies. It also improves the accuracy of HSI classification by transferring statistical knowledge to eliminate the local bias in the test phase. Typically, DL-based HTD methods can be categorized into two major types: supervised learning and unsupervised learning. In supervised learning, target samples are synthesized and expanded and an end-to-end detector is then constructed by training on a large number of pixel pairs [15]. However, synthesizing and expanding samples that are not present in the data "exhausts" computing resources and wastes time matching the real situation. Unsupervised networks, as opposed to supervised networks, do not need to address the challenge of inadequate training samples [21]. It is customary to exploit unsupervised networks with specific constraints to improve target recognition capabilities, subsequently employing a straightforward matching strategy for target detection [15].

Although DL-based methods generally exhibit better detection performance compared to traditional methods, DL-based approaches often have numerous trainable parameters and require a large amount of labeled data to achieve optimal detection performance. In practice, it is challenging to obtain a large number of accurately labeled hyperspectral samples.

Adversarial learning, which is an advanced deep learning mechanism, has exhibited promising results in hyperspectral tasks such as classification [22] and anomaly detection [23]. For instance, Gao et al. [24] employed adversarial complementary learning (ACL) strategies grounded in adversarial learning to extract complementary information from various data sources. The author further creates primary land cover representations and supplementary representations for HSI classification in the context of multisource remote sensing data.

Recently, adversarial learning has been incorporated into two methods in the field of hyperspectral target detection (HTD): the Spectral-Spatial Target Detector (SSTD) [25] and the Background Learning Based on Target Suppression Constraint (BLTSC) [26].

SSTD involves dimensionality reduction of high-dimensional hyperspectral data by utilizing band selection through adversarial learning. This is followed by spectral detection on the reduced data and spatial detection on the selected data. The results from both detections are then fused to obtain the final detection outcome. However, it should be noted that SSTD does not effectively differentiate between background and target information. This limitation may introduce inaccuracies in the dimensionality reduction process, subsequently affecting the overall detection performance.

In BLTSC, an Adversarial Autoencoder (AAE) is employed for extracting background information through adversarial learning. This extracted background information is then reconstructed. However, it is important to highlight that inaccurate initial screening in BLTSC may cause deviations from the original background distribution, leading to imprecise detection results.

The scarcity of target samples in HSIs presents a challenge as it hampers the model's ability to understand the distribution of target spectra effectively. In contrast, there is an abundance of background samples that contain valuable latent information. By extracting and leveraging this latent information, we can effectively highlight the differences between the target and background, which facilitates background suppression and target detection.

The process of extracting features from the background can be considered as an unsupervised learning process. However, it is crucial to address the challenge of relying on a dataset constructed with limited prior target spectral information, which can potentially result in inaccuracies.

To extract the underlying distribution information from HSI with complex backgrounds and alleviate the impact of inaccurately labeled samples, we propose an adaptive adversarial learning framework for HTD. This framework consists of following three fundamental steps.

1) *Sample Selection for Adaptive Adversarial Learning:* In this step, we employ simple detectors to identify high-probability target and background instances. These selected instances are used to create a pseudodataset for adversarial learning. The objective is to achieve a more accurate estimation of the background distribution.
2) *Adaptive Adversarial Learning on the Background Latent Space:* This step involves conducting adversarial learning on the latent space of the background. The goal is to assess the distribution of the background and acquire its latent spatial feature distribution. Adaptive distance constraints are applied to improve the accuracy of background reconstruction. These constraints ensure that the background spectrum is reliably reconstructed while discouraging the accurate reconstruction of target spectra. During training,

the encoder is trained to align the latent space representation of the background with the prior distribution. This alignment enables the discriminator to distinguish the background latent space from other latent spaces. The residual difference between the original and reconstructed spectra is then analyzed to determine the probability of these spectra being targets.

3) *Target Detection with a Combination Strategy:* The final step involves target detection using a combination strategy. This strategy aims to effectively constrain the background for precise target detection and minimize false positives. It combines the differences between targets and backgrounds in the latent space with the residual difference between the reconstructed and original spectra. This combination results in highly accurate target detection.

In summary, our proposed method makes the following key contributions.

1) *Adaptive AAE Construction:* We construct an AAE with adaptive constraints to highlight the target and capture the background distribution and reconstruction in the latent space. To address the issue of limited training samples, we selectively choose training data based on the results of rough detection.
2) *Accurate Background Information Extraction:* By utilizing adaptive constraints, we extract precise background information. During the training process, we reconstruct the background spectrum while incorporating prior target information and a constructed pseudotarget dataset to create an average target spectrum. This average target spectrum, along with adaptive constraints, guides the reconstruction of the background spectrum to closely resemble the actual background.
3) *Utilizing Extracted Information for Precise Detection:* We leverage the extracted information to achieve accurate detection results. By analyzing the information in the latent space, we detect targets, while the reconstructed spectra enable effective background suppression. The joint operation of these two components leads to stable and precise detection outcomes.

## II. Related Work

An HSI image is essentially a three-dimensional matrix, with the first two dimensions representing spatial information and the third dimension representing spectral information. The matrix is of size $M \times N \times L$, where $M$ and $N$ correspond to the dimensions of the image along the $x$ and $y$ axes, respectively, and $L$ corresponds to the number of spectral bands captured by the sensor. Each pixel in the matrix, denoted by $x_{ij}$, represents the spectral reflectance vector for a particular material or surface property at that location. Spectral reflectance refers to the amount of light reflected by a surface at each wavelength in the electromagnetic spectrum. Thus, the HSI matrix contains valuable information about the material composition and intrinsic properties of objects in the scene.

$$\mathrm{X} = \begin{bmatrix} x_{1,1} & x_{2,1} & \cdots & x_{N,1} \\ x_{1,2} & x_{2,2} & \cdots & x_{N,2} \\ \cdots & \cdots & \cdots & \cdots \\ x_{1,M} & x_{2,M} & \cdots & x_{N,M} \end{bmatrix}. \tag{1}$$

In HTD tasks, it is usually necessary to flatten the HSI into a two-dimensional matrix form as follows:

$$\mathrm{X} = [x_1, x_2, \ldots, x_{M\times N}] \tag{2}$$

where $x_i$ represents the spectral vector of each pixel, in the following form:

$$x_i^{1\times L} = \left[x_i^1, x_i^2, \ldots, x_i^L\right]^\top. \tag{3}$$

The prior target spectral information $d$ has the same shape as $x_i$.

### A. CEM Detector

The goal of the CEM filter is to design a finite impulse response (FIR) linear filter, denoted by $w$, with the intention of minimizing the average output energy of the filter, represented by $E = w^\top Rw$. The autocorrelation matrix of the data samples is denoted by R. In addition it is desired to keep the output of the target prior spectral information, denoted by d, constant. Specifically, $d^\top w = 1$. By constructing such a filter, the optimal filter vector $w^*$ can be obtained

$$w^* = \frac{\mathbf{R}^{-1}d}{d^\top \mathbf{R}^{-1}d} \tag{4}$$

where *R* represents the autocorrelation matrix

$$\mathbf{R} = \frac{1}{M \times N} \sum_{i=1}^{M\times N} x_i x_i^\top. \tag{5}$$

The final detection result of CEM is

$$y = (w^*)^\top X. \tag{6}$$

### B. Adversarial Autoencoder

The AAE is a method that combines the structure of an autoencoder (AE) with the adversarial concept from generative adversarial networks (GAN) [27]. The AAE consists of following three main components.

1) *Encoder ($z = E(x)$):* Similar to the generator ($G$) in GAN, the encoder transforms the input ($x$) into a latent space vector ($z$), and the distribution of $z$ is denoted as $q(z)$.
2) *Decoder ($x' = De(z)$):* The decoder takes the latent vector ($z$) as input and reconstructs the original input ($x$). The entire process from encoding to decoding is represented as $x' = De(E(x))$
3) *Discriminator ($y = Di(z)$):* The discriminator takes a latent vector ($z$) as input and classifies it as belonging to either the latent space or being sampled from the prior distribution ($p(z)$). The discriminator's output ($y$) is a scalar value between 0 and 1, indicating the probability.

Considering $q(z|x)$ as the encoding distribution and $p(x|z)$ as the decoding distribution, and assuming $p_d(x)$ represents the

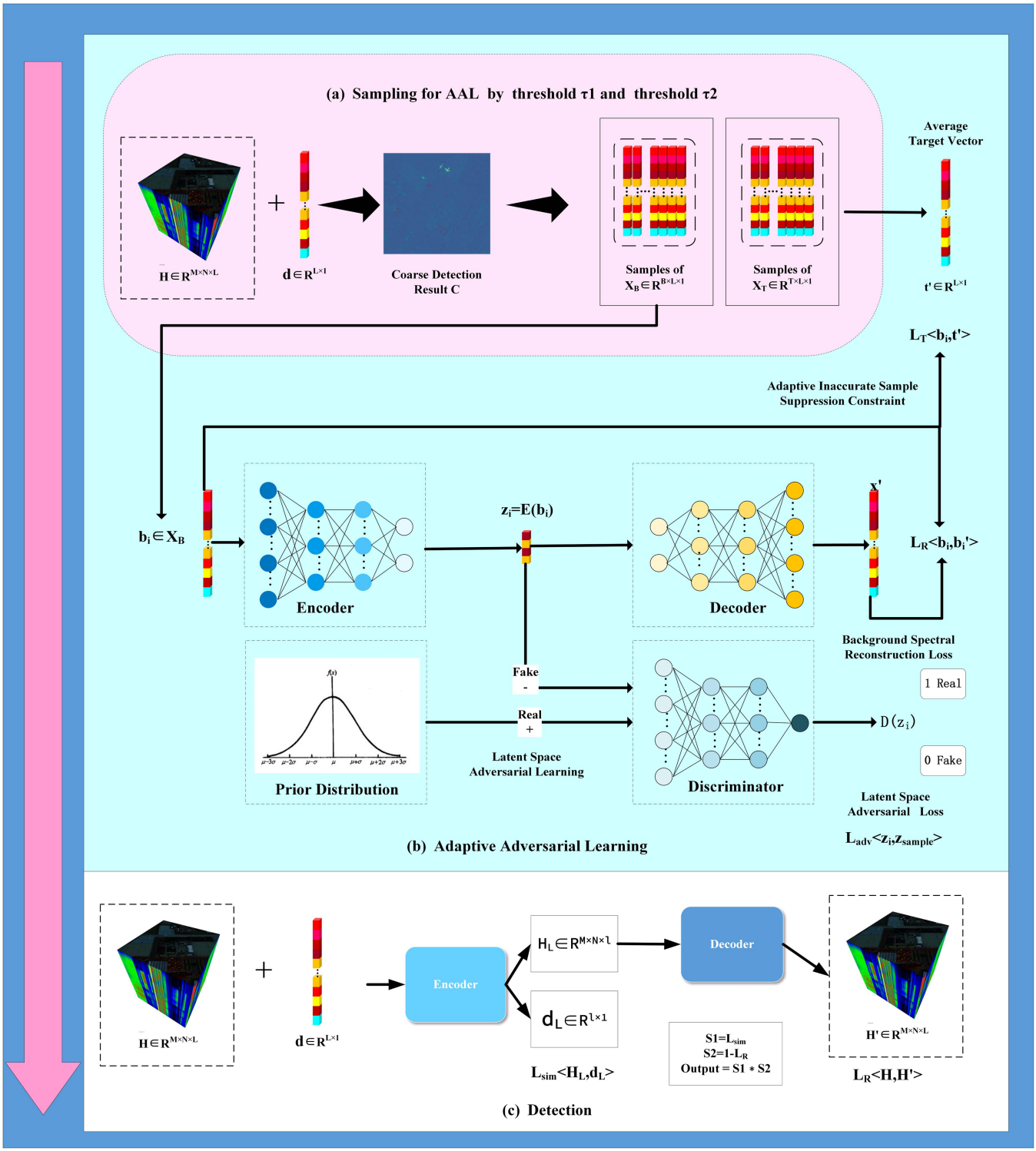



Fig. 1. Schematic of the proposed ABLAL-based HTD method.

true data distribution, the aggregated posterior distribution of $q(z)$ in the latent space can be expressed as follows:

$$q(z) = \int_x q(z|x)p_d(x)\mathrm{d}x. \tag{7}$$

The AAE aims to minimize the Jensen–Shannon (JS) divergence between the prior distribution $p(z)$ and the aggregated posterior distribution of $q(z)$. During the training process, two main errors are minimized: the reconstruction error and the adversarial training error. Adversarial training consists of two steps: Deceptive Step: In this step, the Encoder $E$ (or Generator

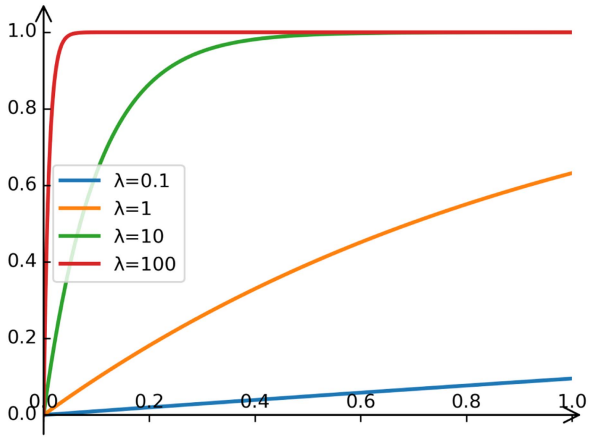
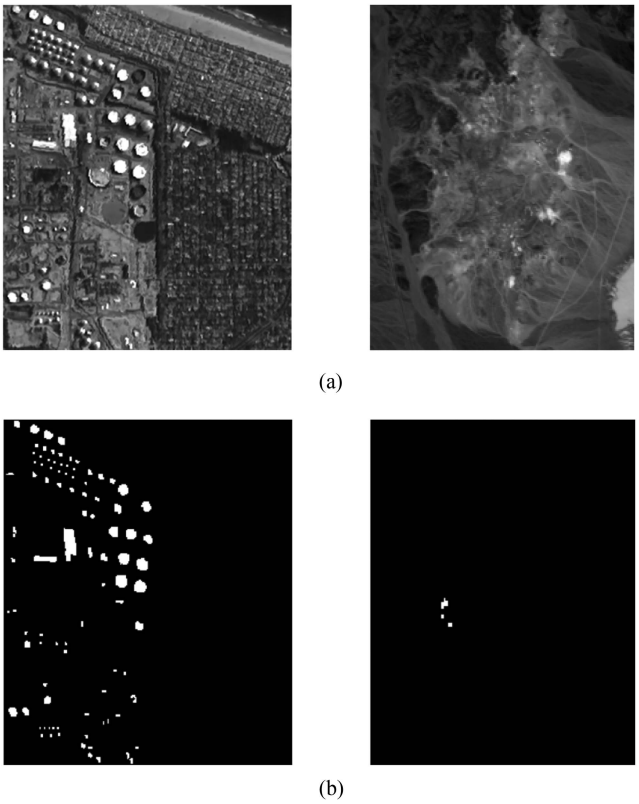


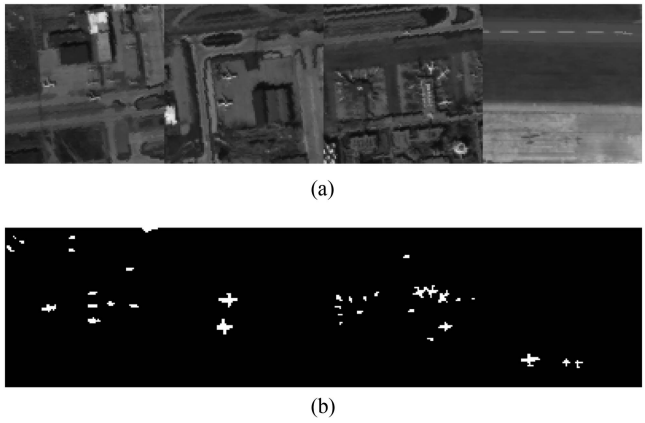

Fig. 2. Function curve with different values of $\gamma$.



(a)

(b)

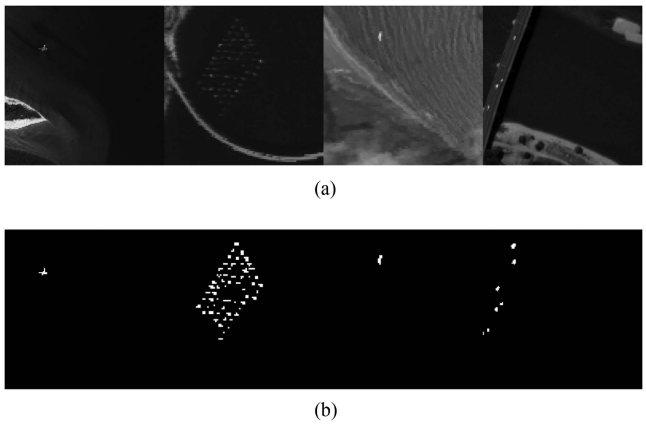
Fig. 3. 30th band images, GT map of the targets of El Segundo and Cuprite. (a) Band 30. (b) GT.



(a)

(b)

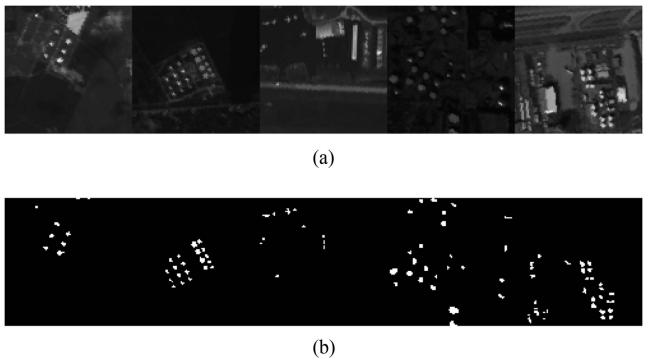
Fig. 4. 30th band images, GT map of the targets of ABU-Airport. (a) Band 30. (b) GT.



(a)

(b)

Fig. 5. 30th band images, GT map of the targets of ABU-Beach. (a) Band 30. (b) GT.



(a)

(b)

Fig. 6. 30th band images, GT map of the targets of ABU-Urban. (a) Band 30. (b) GT.

$G$) aims to deceive the Discriminator $Di$ by generating samples $z$ from the distribution $q(z)$. Discriminative Step: In this step, the Discriminator $Di$ learns to differentiate between samples $z$ from the aggregated posterior distribution $q(z)$ and samples $z$ from the prior distribution $p(z)$. By establishing an adversarial relationship between the Encoder $E$ and the Discriminator $Di$, the AAE encourages the Encoder $E$ to generate latent representations $z$ that can successfully fool the Discriminator $Di$. The adversarial relationship between $E$ and $Di$ can be represented as

$$\min_{E}\max_{Di}\mathbb{E}_{z\sim p(z)}\left[\log Di(z)\right] + \mathbb{E}_{z\sim q(z)}\left[\log\left(1-Di(z)\right)\right]. \tag{8}$$

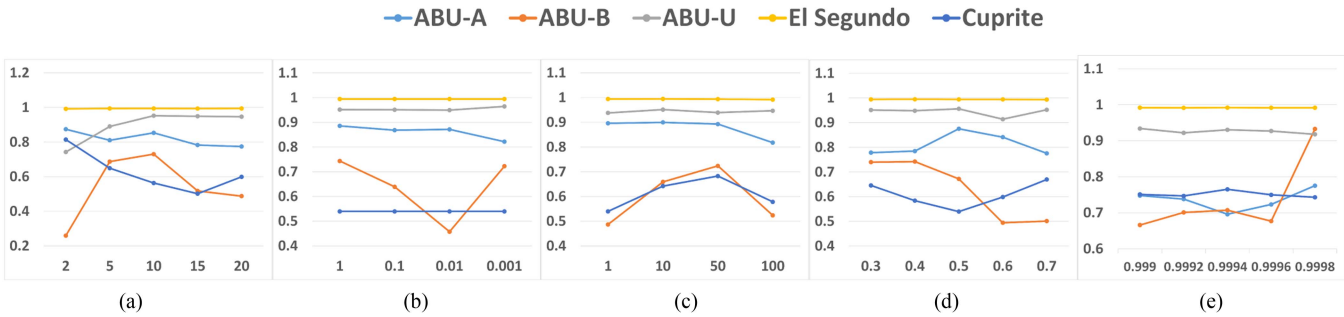



Fig. 7. Parameter sensitivity analysis of influence on different datasets. (a) Number of latent space nodes. (b) $\beta$. (c) $\gamma$. (d) $\tau_1$. (e) $\tau_2$.

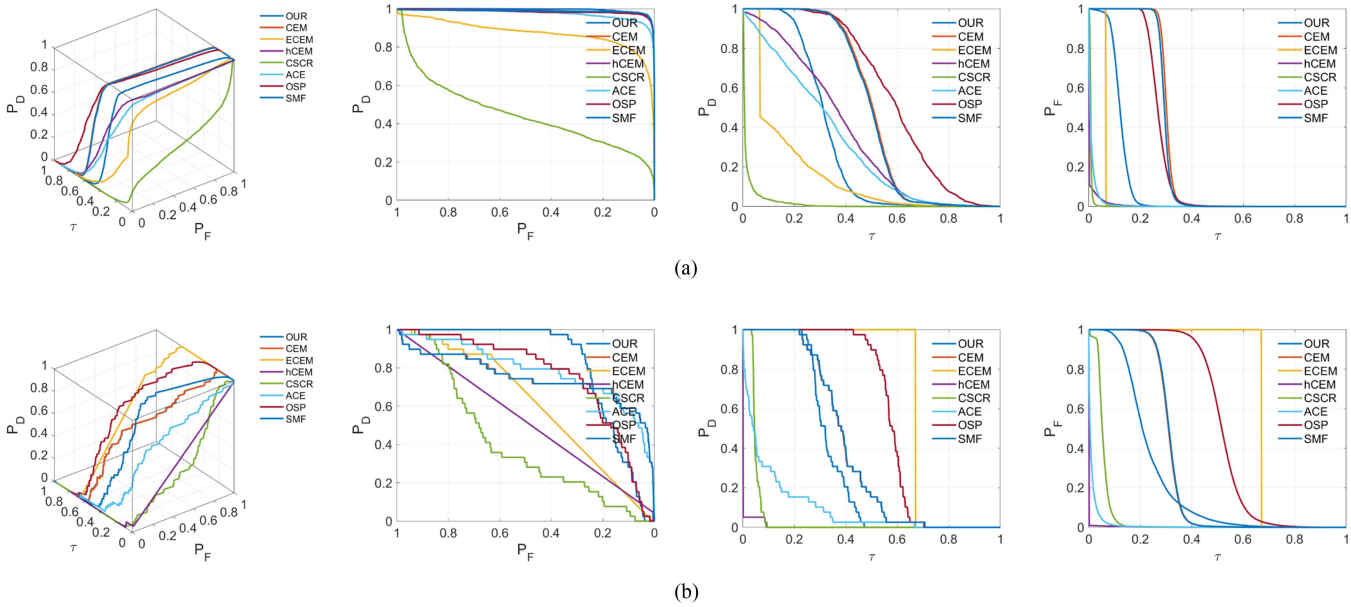



Fig. 8. 3D-ROC and three corresponding 2D-ROC curves of different comparing methods on El Segundo HSI and Cuprite HSI. (Left to right) 3D-ROC curve, 2D-ROC curve of $(P_d, P_f)$, 2-D ROC curves of $(P_d, \tau)$, 2-D ROC curves of $(P_f, \tau)$. (a) El Segundo. (b) Cuprite.

This objective function drives the Encoder E to generate latent vectors z that can maximize the Discriminator's probability of correctly distinguishing between samples from q(z) and samples from p(z).

In this article, we focus on determining the background distribution in HSIs using the AAE approach. In HSIs, the background distribution typically exhibits characteristics of a Gaussian distribution or can be approximated by a Gaussian distribution. To facilitate convenient calculations and analysis, we aim to ensure that the distribution in the latent space obtained by inputting the background data into the Encoder $E$ also follows a Gaussian distribution. Consequently, we select the Gaussian distribution as the prior distribution $p(z)$ for our AAE model.

In our approach, we conduct joint training for the two components of the AAE: the Discriminator $Di$ and the Encoder $E$. The Discriminator $Di$ is responsible for discriminating whether a given sample $z$ originates from the samples generated by the Encoder $E$ or from pseudosamples sampled from the prior distribution $p(z)$. Simultaneously, the Encoder $E$ aims to confuse the

TABLE I
DATASET INFO AND LOCATION OF THE PRIOR TARGET SPECTRAL D

| HSIs | Parameter | | |
|---|---|---|---|
| | *Targets num* | *size* | *d location* |
| El Segundo | 2048 | 300*250 | original |
| Cuprite | 39 | 250*191 | original |
| ABU-Airport-1 | 144 | 100*100 | (51,28) |
| ABU-Airport-2 | 87 | 100*100 | (45,42) |
| ABU-Airport-3 | 170 | 100*100 | (41,68) |
| ABU-Airport-4 | 60 | 100*100 | (83,30) |
| ABU-Beach-1 | 19 | 150*150 | (41,38) |
| ABU-Beach-2 | 202 | 100*100 | (19,52) |
| ABU-Beach-3 | 11 | 100*100 | (20,36) |
| ABU-Beach-4 | 68 | 100*100 | (16,29) |
| ABU-Urban-1 | 67 | 100*100 | (44,44) |
| ABU-Urban-2 | 155 | 100*100 | (42,52) |
| ABU-Urban-3 | 52 | 100*100 | (11,23) |
| ABU-Urban-4 | 272 | 100*100 | (58,6) |
| ABU-Urban-5 | 235 | 100*100 | (55,14) |

TABLE II
EVALUATE AUC SCORES OF THE MAIN COMPONENTS $S1$ AND $S2$ OF OUR METHOD AND THE AUC SCORE OF THE FINAL OUTPUT RESULT

| HSIs | AUC values | | |
|---|---|---|---|
| | *S1* | *S2* | *Output* |
| El Segundo | 0.99252 | 0.55498 | **0.99358** |
| Cuprite | 0.80378 | 0.71296 | **0.90381** |
| ABU-Airport-1 | 0.70549 | 0.85258 | **0.88013** |
| ABU-Airport-2 | 0.68292 | **0.94251** | 0.90702 |
| ABU-Airport-3 | 0.94623 | 0.72263 | **0.95974** |
| ABU-Airport-4 | 0.93536 | 0.92556 | **0.95097** |
| ABU-Beach-1 | 0.95555 | **0.96784** | 0.95645 |
| ABU-Beach-2 | 0.91325 | 0.89335 | **0.91842** |
| ABU-Beach-3 | **0.91167** | 0.76061 | 0.90753 |
| ABU-Beach-4 | 0.89681 | 0.83775 | **0.89843** |
| ABU-Urban-1 | **0.9715** | 0.40192 | 0.95238 |
| ABU-Urban-2 | **0.99927** | 0.30989 | 0.99916 |
| ABU-Urban-3 | 0.9183 | 0.62581 | **0.95089** |
| ABU-Urban-4 | 0.99631 | 0.6454 | **0.99634** |
| ABU-Urban-5 | 0.96193 | 0.72019 | **0.96781** |

The bold entities represent the best performance in each row.

TABLE III
AUC VALUES DERIVED OF EL SEGUNDO AND CUPRITE HSIS FROM THE THREE 2-D ROC CURVES PRESENTED IN FIGS. 8 AND 9

| HSIs | AUCs | Detector | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | OUR | CEM | ECEM | hCEM | CSCR | ACE | OSP | SMF |
| El Segundo | $AUC_{(D,F)}$ | **0.99358** | 0.98636 | 0.873679 | 0.987763 | 0.448939 | 0.973986 | 0.984076 | 0.986481 |
| | $AUC_{(D,\tau)}$ | 0.314092 | 0.499611 | 0.159717 | 0.357185 | 0.017261 | 0.309939 | **0.59255** | 0.494255 |
| | $AUC_{(F,\tau)}$ | 0.122251 | 0.303243 | 0.066383 | **0.00518** | 0.007934 | 0.010792 | 0.271335 | 0.295275 |
| | $AUC_{OD}$ | 1.185422 | 1.182728 | 0.967013 | **1.33977** | 0.458266 | 1.273133 | 1.305291 | 1.185461 |
| Cuprite | $AUC_{(D,F)}$ | **0.82364** | 0.749049 | 0.608536 | 0.520331 | 0.403325 | 0.78759 | 0.760867 | 0.748264 |
| | $AUC_{(D,\tau)}$ | 0.330802 | 0.388157 | **0.669989** | 0.004575 | 0.05049 | 0.095016 | 0.575884 | 0.389501 |
| | $AUC_{(F,\tau)}$ | 0.234166 | 0.310688 | 0.66994 | **0.001372** | 0.05648 | 0.01274 | 0.519379 | 0.312569 |
| | $AUC_{OD}$ | **0.92028** | 0.826518 | 0.608585 | 0.523534 | 0.397335 | 0.869865 | 0.817372 | 0.825196 |

The bold entities represent the best performance in each row.

TABLE IV
AUC VALUES DERIVED OF ABU-AIRPORT HSIS FROM THE THREE 2-D ROC CURVES PRESENTED IN FIGS. 10 AND 11

| HSIs | AUCs | Detector | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | OUR | CEM | ECEM | hCEM | CSCR | ACE | OSP | SMF |
| ABU-Airport-1 | $AUC_{(D,F)}$ | **0.88013** | 0.780684 | 0.659492 | 0.737468 | 0.865334 | 0.813756 | 0.778067 | 0.782239 |
| | $AUC_{(D,\tau)}$ | 0.420254 | 0.22361 | 0.012251 | 0.144143 | 0.093326 | 0.05269 | **0.70341** | 0.224044 |
| | $AUC_{(F,\tau)}$ | 0.30269 | 0.129402 | **0.00184** | 0.015099 | 0.035334 | 0.002996 | 0.563459 | 0.129718 |
| | $AUC_{OD}$ | **0.99769** | 0.874892 | 0.669902 | 0.866512 | 0.923326 | 0.86345 | 0.918021 | 0.876565 |
| ABU-Airport-2 | $AUC_{(D,F)}$ | **0.90702** | 0.401452 | 0.474378 | 0.516192 | 0.855277 | 0.473004 | 0.201741 | 0.392418 |
| | $AUC_{(D,\tau)}$ | **0.44903** | 0.156146 | 0.001251 | 0.098306 | 0.147944 | 0.003789 | 0.369565 | 0.153059 |
| | $AUC_{(F,\tau)}$ | 0.289131 | 0.172774 | **0.0014** | 0.086408 | 0.055427 | 0.005027 | 0.502731 | 0.17143 |
| | $AUC_{OD}$ | **1.06692** | 0.384824 | 0.474227 | 0.52809 | 0.947794 | 0.471766 | 0.068575 | 0.374047 |
| ABU-Airport-3 | $AUC_{(D,F)}$ | **0.95974** | 0.494383 | 0.537728 | 0.20793 | 0.894586 | 0.485475 | 0.053737 | 0.49128 |
| | $AUC_{(D,\tau)}$ | 0.468995 | 0.271595 | 0.002125 | 0.020853 | 0.04316 | 0.004357 | **0.49697** | 0.269139 |
| | $AUC_{(F,\tau)}$ | 0.274141 | 0.269733 | **0.00228** | 0.04479 | 0.012142 | 0.005151 | 0.733736 | 0.268123 |
| | $AUC_{OD}$ | **1.1546** | 0.496245 | 0.537577 | 0.183993 | 0.925604 | 0.484681 | -0.18303 | 0.492296 |
| ABU-Airport-4 | $AUC_{(D,F)}$ | **0.95097** | 0.950917 | 0.735929 | 0.775 | 0.934878 | 0.945672 | 0.516343 | 0.950434 |
| | $AUC_{(D,\tau)}$ | 0.4653 | 0.384126 | 0.026808 | 0.232947 | 0.146381 | 0.194715 | **0.53723** | 0.369714 |
| | $AUC_{(F,\tau)}$ | 0.154741 | 0.122403 | **0.00183** | 0.009633 | 0.067203 | 0.002404 | 0.520792 | 0.101233 |
| | $AUC_{OD}$ | **1.26153** | 1.21264 | 0.760904 | 0.998314 | 1.014056 | 1.137983 | 0.532782 | 1.218915 |

The bold entities represent the best performance in each row.

Discriminator $Di$ by encoding the background sample results. The Encoder $E$ minimizes the loss specified in (8), while the discriminator $Di$ aims to maximize it. Consequently, we define the adversarial training objective as $\min_E \max_{Di} L_{\text{adv}}$, and it can be expressed as follows:

$$L_{\text{adv}} = \mathbb{E}\left[\log Di(N(0,1))\right] + \mathbb{E}\left[\log\left(1 - Di(z)\right)\right] \quad (9)$$

TABLE V
AUC VALUES DERIVED OF ABU-BEACH HSIS FROM THE THREE 2-D ROC CURVES PRESENTED IN FIGS. 12 AND 13

| HSIs | AUCs | Detector | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | OUR | CEM | ECEM | hCEM | CSCR | ACE | OSP | SMF |
| ABU-Beach-1 | $AUC_{(D,F)}$ | 0.951662 | 0.470762 | 0.794131 | 0.405013 | **0.98746** | 0.299298 | 0.132212 | 0.498058 |
| | $AUC_{(D,\tau)}$ | 0.355373 | 0.332853 | 0.005207 | 0.421743 | 0.390708 | 0.001568 | **0.58187** | 0.333944 |
| | $AUC_{(F,\tau)}$ | 0.071356 | 0.351189 | **0.00527** | 0.512099 | 0.040655 | 0.005877 | 0.827097 | 0.344458 |
| | $AUC_{OD}$ | 1.235679 | 0.452426 | 0.794065 | 0.314657 | **1.33751** | 0.294989 | -0.11302 | 0.487544 |
| ABU-Beach-2 | $AUC_{(D,F)}$ | 0.901965 | 0.526783 | 0.640602 | 0.18327 | **0.97286** | 0.478377 | 0.077337 | 0.53196 |
| | $AUC_{(D,\tau)}$ | 0.116749 | 0.351027 | 0.003273 | 0.182672 | 0.161611 | 0.004602 | **0.73387** | 0.346492 |
| | $AUC_{(F,\tau)}$ | 0.061004 | 0.344591 | **0.00353** | 0.426755 | 0.051914 | 0.005718 | 0.878103 | 0.338976 |
| | $AUC_{OD}$ | 0.95771 | 0.533219 | 0.640349 | -0.06081 | **1.08256** | 0.477261 | -0.0669 | 0.539476 |
| ABU-Beach-3 | $AUC_{(D,F)}$ | 0.943156 | 0.516204 | 0.621957 | 0.92894 | **0.99963** | 0.474203 | 0.492314 | 0.531758 |
| | $AUC_{(D,\tau)}$ | **0.5925** | 0.265749 | 0.001322 | 0.342568 | 0.587584 | 0.003337 | 0.45914 | 0.254823 |
| | $AUC_{(F,\tau)}$ | 0.189006 | 0.257895 | **0.00157** | 0.022678 | 0.050486 | 0.005265 | 0.428423 | 0.242 |
| | $AUC_{OD}$ | 1.346654 | 0.524058 | 0.621706 | 1.24883 | **1.53673** | 0.472275 | 0.523031 | 0.544581 |
| ABU-Beach-4 | $AUC_{(D,F)}$ | **0.91263** | 0.668471 | 0.664524 | 0.711722 | 0.516235 | 0.567464 | 0.369415 | 0.644205 |
| | $AUC_{(D,\tau)}$ | 0.387601 | 0.377634 | 0.006101 | 0.119354 | 0.05072 | 0.013483 | **0.45125** | 0.376457 |
| | $AUC_{(F,\tau)}$ | 0.116854 | 0.303783 | 0.00621 | **0.00501** | 0.050823 | 0.009818 | 0.512107 | 0.315949 |
| | $AUC_{OD}$ | **1.18338** | 0.742322 | 0.664415 | 0.826071 | 0.516132 | 0.571129 | 0.308557 | 0.704713 |

The bold entities represent the best performance in each row.

TABLE VI
AUC VALUES DERIVED OF ABU-URBAN HSIS FROM THE THREE 2-D ROC CURVES PRESENTED IN FIGS. 14 AND 15

| HSIs | AUCs | Detector | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | OUR | CEM | ECEM | hCEM | CSCR | ACE | OSP | SMF |
| ABU-Urban-1 | $AUC_{(D,F)}$ | 0.952383 | 0.522596 | 0.772714 | 0.409059 | **0.99288** | 0.414954 | 0.235127 | 0.530626 |
| | $AUC_{(D,\tau)}$ | **0.51258** | 0.288778 | 0.001813 | 0.002121 | 0.26683 | 0.002489 | 0.442966 | 0.283295 |
| | $AUC_{(F,\tau)}$ | 0.132048 | 0.284862 | **0.00196** | 0.04054 | 0.038206 | 0.005278 | 0.564133 | 0.277925 |
| | $AUC_{OD}$ | **1.33292** | 0.526512 | 0.772565 | 0.37064 | 1.221508 | 0.412165 | 0.11396 | 0.535996 |
| ABU-Urban-2 | $AUC_{(D,F)}$ | **0.99916** | 0.471311 | 0.57439 | 0.574487 | 0.747304 | 0.412509 | 0.209987 | 0.467284 |
| | $AUC_{(D,\tau)}$ | **0.43372** | 0.297284 | 0.002617 | 0.249539 | 0.080027 | 0.002994 | 0.37278 | 0.278022 |
| | $AUC_{(F,\tau)}$ | 0.108416 | 0.305196 | **0.00276** | 0.181732 | 0.018338 | 0.00555 | 0.58864 | 0.288625 |
| | $AUC_{OD}$ | **1.32446** | 0.463399 | 0.574251 | 0.642294 | 0.808993 | 0.409953 | -0.00587 | 0.456681 |
| ABU-Urban-3 | $AUC_{(D,F)}$ | **0.95089** | 0.418148 | 0.69544 | 0.328307 | 0.904904 | 0.285377 | 0.387688 | 0.416395 |
| | $AUC_{(D,\tau)}$ | 0.274165 | 0.240993 | 0.002747 | 0 | 0.1426 | 0.001664 | **0.36796** | 0.239979 |
| | $AUC_{(F,\tau)}$ | 0.100265 | 0.257702 | **0.0029** | 0.076863 | 0.026433 | 0.005607 | 0.4925 | 0.256861 |
| | $AUC_{OD}$ | **1.12479** | 0.401439 | 0.695291 | 0.251444 | 1.021071 | 0.281434 | 0.263147 | 0.399513 |
| ABU-Urban-4 | $AUC_{(D,F)}$ | **0.99634** | 0.464297 | 0.585031 | 0.190108 | 0.669074 | 0.412405 | 0.118031 | 0.466481 |
| | $AUC_{(D,\tau)}$ | 0.340691 | **0.43861** | 0.00841 | 0.004535 | 0.077098 | 0.003191 | 0.289892 | 0.437517 |
| | $AUC_{(F,\tau)}$ | 0.060591 | 0.444134 | 0.008563 | 0.161734 | 0.017446 | **0.00552** | 0.448157 | 0.442662 |
| | $AUC_{OD}$ | **1.27644** | 0.458771 | 0.584878 | 0.032909 | 0.728726 | 0.410075 | -0.04023 | 0.461336 |
| ABU-Urban-5 | $AUC_{(D,F)}$ | **0.96791** | 0.452576 | 0.602853 | 0.41829 | 0.699951 | 0.457041 | 0.323777 | 0.441913 |
| | $AUC_{(D,\tau)}$ | **0.57246** | 0.21376 | 0.002112 | 0.024825 | 0.074523 | 0.003469 | 0.396645 | 0.207001 |
| | $AUC_{(F,\tau)}$ | 0.290105 | 0.223815 | **0.00227** | 0.063007 | 0.040576 | 0.005028 | 0.446991 | 0.218836 |
| | $AUC_{OD}$ | **1.25027** | 0.442521 | 0.602699 | 0.380108 | 0.733898 | 0.455482 | 0.273431 | 0.430078 |

The bold entities represent the best performance in each row.

where $z$ is the latent representation of the background sample $x$ obtained by inputting Encoder $E$, $N(0,1)$ represents a prior Gaussian distribution $p(z)$. By minimizing the adversarial loss $L_{\text{adv}}$, we aim to bring the distribution of the background latent space closer to the desired Gaussian distribution. This optimization process encourages the encoder $E$ to output latent representations that follow the Gaussian distribution, aligning the latent space with the prior distribution.

To complement the adversarial training, we also incorporate a reconstruction loss into the autoencoder part of the AAE. This additional loss term ensures that the latent space vector obtained by encoding the input data through the encoder can be decoded by the decoder, resulting in the restoration of the original data. The reconstruction loss $L_R$ aims to minimize the error between the input data $b_i \in H_B$ and its corresponding reconstructed data $b'_i \in H'_B$. By minimizing this reconstruction loss, we encourage the encoder–decoder pair to learn a representation in the latent space that captures the essential information of the input data, allowing for accurate reconstruction.

$$L_R = \sum_i \|b_i - b'_i\|_2. \quad (10)$$

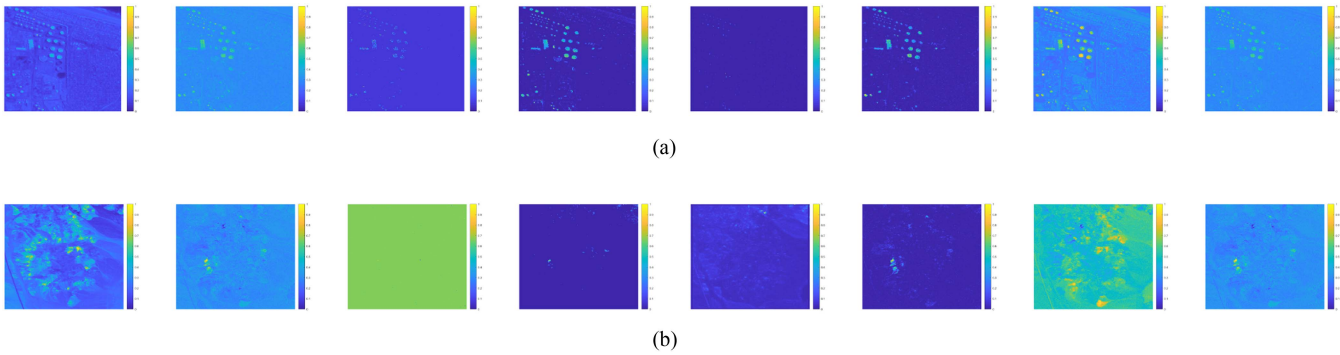


Fig. 9. Detection results of different comparing methods on El Segundo HSI and Cuprite HSI. (Left to right) OUR, CEM, ECEM, hCEM, CSCR, ACE, OSP, SMF. (a) El Segundo. (b) Cuprite.

### C. Spectral Similarity Measurement

In the context of HTD, dimensionality reduction techniques are often employed as a preliminary step or during the HTD process to handle the high dimensionality of HSIs. One such technique is the use of autoencoders, which can serve as a dimensionality reduction method. When the dimensionality of the latent space in an autoencoder is smaller than the dimensionality of the input data, the autoencoder effectively reduces the dimensionality of the data. In this way, autoencoders can be viewed as a form of dimensionality reduction technique in the HTD process [28]. Furthermore, autoencoders have the advantage of enabling nonlinear dimensionality reduction by selecting appropriate activation functions. This contrasts with classical techniques like Principal Component Analysis (PCA), which only support linear dimensionality reduction.

In our model, we intentionally choose a lower dimensionality for the latent features compared to the input data dimensionality. As a result, the information obtained in the latent space can be considered as a form of dimensionality reduction for the input data. In the context of our approach, we employ the SAM [29] to discriminate between targets and backgrounds specifically in the latent space. SAM is a commonly used spectral matching technique that compares the spectral angles between reference spectra and the spectra of pixels or samples. By applying SAM in the latent space, we leverage the reduced dimensionality and the discriminative power of the spectral angles to identify and differentiate between target and background samples

$$\theta(x_1, x_2) = \cos^{-1}\left(\frac{x_1^\top x_2}{\|x_1\|\,\|x_2\|}\right) = \cos^{-1}\left(\frac{x_1^\top x_2}{\sqrt{x_1^\top x_1}\sqrt{x_2^\top x_2}}\right). \tag{11}$$

To ensure that the computed results are within the range of [0, 1] and to facilitate subsequent calculations, we have applied normalization to the SAM values

$$d_s = 1 - \frac{1}{\pi}\cos^{-1}\left(\frac{x_1^\top x_2}{\sqrt{x_1^\top x_1}\sqrt{x_2^\top x_2}}\right). \tag{12}$$

TABLE VII
AVERAGE COMPUTING TIME (IN SECONDS) OF THE COMPARED METHODS

| HSIs | Computing Time (in seconds) | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | OUR | CEM | ECEM | hCEM | CSCR | ACE | OSP | SMF |
| EL Segundo | 6.369885683 | **0.46875** | 25.70503139 | 3.328125 | 8208.4375 | 4.9375 | 2.1875 | 2.921875 |
| Cuprite | 3.843737602 | **0.21875** | 11.08897018 | 1.828125 | 3939.03125 | 2.578125 | 1.25 | 1.625 |
| ABU-Airport-1 | 0.98752594 | **0.15625** | 4.789309978 | 0.78125 | 1082.859375 | 0.9375 | 0.3125 | 0.484375 |
| ABU-Airport-2 | 0.96365428 | **0.16124** | 5.147999287 | 1.125 | 1079.703125 | 0.640625 | 0.3125 | 0.328125 |
| ABU-Airport-3 | 1.004427671 | **0.17021** | 5.073733807 | 0.984375 | 1014.828125 | 0.78125 | 0.3125 | 0.34375 |
| ABU-Airport-4 | 1.009914875 | **0.171875** | 4.34210372 | 0.703125 | 845.234375 | 0.484375 | 0.3125 | 0.40625 |
| ABU-Beach-1 | 1.886020899 | **0.15625** | 6.354000807 | 0.578125 | 1788.953125 | 1.28125 | 0.46875 | 0.8125 |
| ABU-Beach-2 | 1.004435539 | **0.171875** | 5.298732758 | 0.34375 | 857.40625 | 0.484375 | 0.3125 | 0.328125 |
| ABU-Beach-3 | 1.057825089 | **0.16981** | 4.575999975 | 1.3125 | 821.765625 | 0.515625 | 0.3125 | 0.328125 |
| ABU-Beach-4 | 1.920382977 | **0.16893** | 3.417998552 | 0.65625 | 821.828125 | 0.078125 | 0.3125 | 0.046875 |
| ABU-Urban-1 | 1.137099743 | **0.18011** | 5.271927118 | 1.96875 | 942.03125 | 0.5 | 0.3125 | 0.46875 |
| ABU-Urban-2 | 0.978586912 | **0.1875** | 5.033922434 | 0.359375 | 960.921875 | 0.640625 | 0.3125 | 0.34375 |
| ABU-Urban-3 | 0.941962481 | **0.1875** | 4.822415352 | 1.296875 | 853.125 | 0.546875 | 0.3125 | 0.515625 |
| ABU-Urban-4 | 0.956938744 | **0.1801** | 5.092991114 | 0.875 | 977.015625 | 0.640625 | 0.3125 | 0.328125 |
| ABU-Urban-5 | 0.942192793 | **0.171875** | 4.691999912 | 0.96875 | 947.984375 | 0.640625 | 0.3125 | 0.34375 |
| Average | 1.666972748 | **0.194735** | 6.713809093 | 1.140625 | 1676.075 | 1.045833 | 0.510417 | 0.641667 |

The bold entities represent the best performance in each row.

In our model, we represent the latent space outputs of the input data $x$ and the target $d$ as $x_L$ and $d_L$, respectively. The similarity between these two latent space representations can be expressed using a value $L_{\text{sim}}$, which falls within the range of [0, 1]. A higher value of $L_{\text{sim}}$ suggests a greater similarity between the two spectra

$$L_{\text{sim}} = d_s\langle x_L, d_L\rangle. \tag{13}$$

## III. PROPOSED METHOD

Our proposed method for HTD is composed of three stages. In the first stage, we extract adaptive adversarial learning samples by applying pseudotarget set and pseudobackground set thresholds to the coarse detection results. In the second stage, we use these samples to train an AAE with three types of constraints. These constraints aim to improve the background reconstruction performance of the AAE while maintaining the separation of background and targets in the latent space. Finally, the trained AAE is used for target detection. Fig. 1 depicts the general scheme of our method.

### A. Sample Selection for Adaptive Adversarial Learning

In this study, the Adaptive Adversarial Learning sample selection process outlined in [30] is utilized to provide adequate

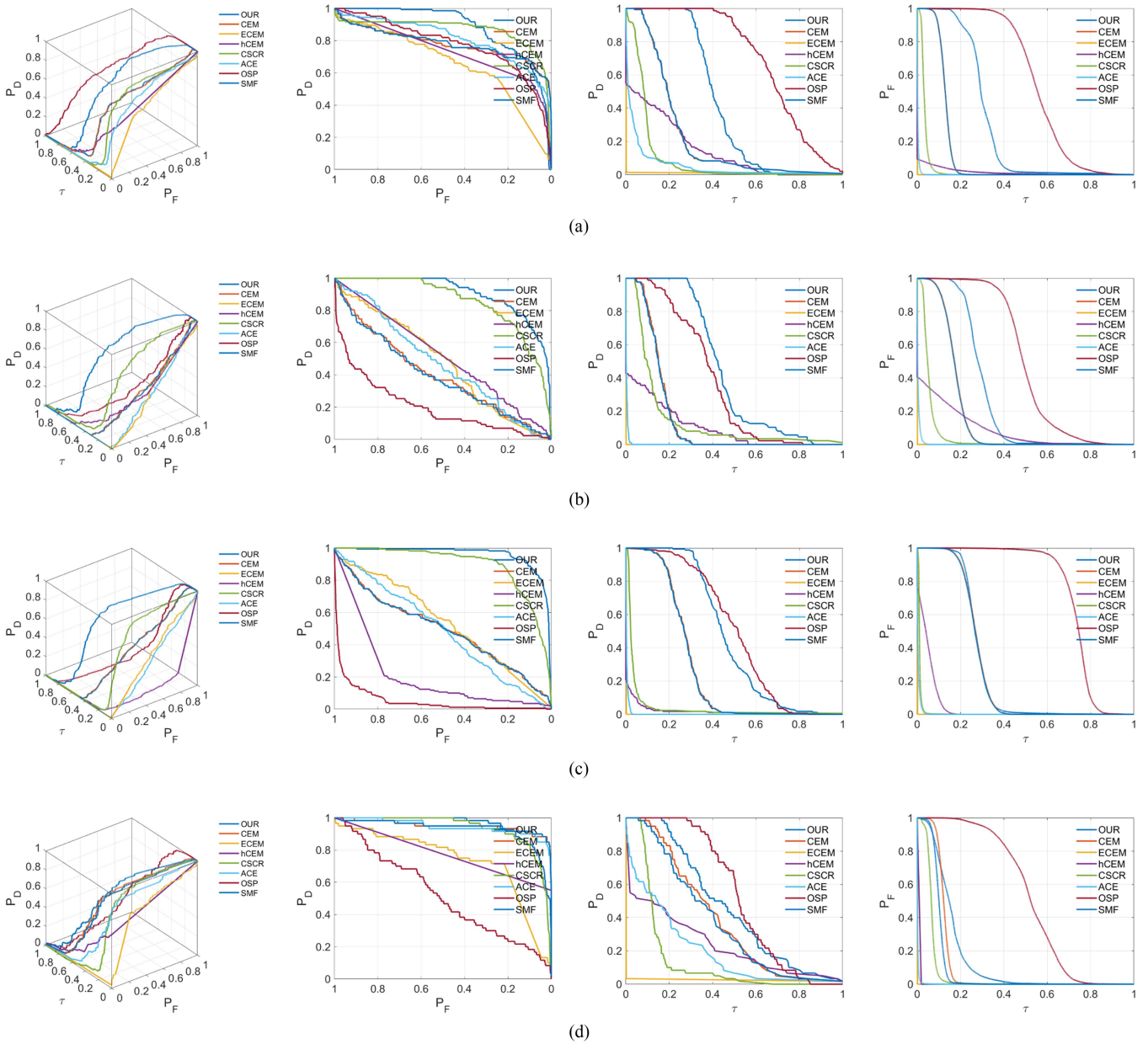



Fig. 10. 3D-ROC and three corresponding 2D-ROC curves of different comparing methods on ABU-Airport HSIs. (Left to right) 3D-ROC curve, 2D-ROC curve of $(P_d, P_f)$, 2-D ROC curves of $(P_d, \tau)$, 2-D ROC curves of $(P_f, \tau)$. (a) ABU-Airport-1. (b) ABU-Airport-2. (c) ABU-Airport-3. (d) ABU-Airport-4.

TABLE VIII
TIME COMPLEXITY FOR THE COMPARED METHODS

| Compared methods | Time complexity |
|---|---|
| OUR | $O(M \times N \times L^2 \times b)$ |
| CEM | $O(M \times N \times L^2)$ |
| ECEM | $O(ncem \times nlayer \times M \times N \times L^2)$ |
| hCEM | $O(iter \times M \times N \times L^2)$ |
| CSCR | $O(M \times N \times (w_{out}^2 - w_{in}^2)^2 \times L)$ |
| ACE | $O(M \times N \times L^2)$ |
| OSP | $O(M \times N \times L^2)$ |
| SMF | $O(M \times N \times L^2)$ |

training samples for Adaptive Adversarial Learning (AAL). As the coarse detector, CEM is employed to construct a pseudo-dataset that comprises sufficient samples for training purposes. It is worth noting that the coarse detector employed in this study may be substituted with other commonly used detectors. By using (6), the CEM detector can obtain a preliminary detection outcome C, where each pixel value indicates the probability of a particular spectrum becoming a target sample. In reference to the distribution of real target and background pixels in the actual HSI, we sort the pixels of X based on the corresponding values in spatial map C, as shown in Fig. 1. Specifically, we select samples with smaller values at the top $\tau_1$ to form a pseudobackground dataset $X_B \in R^{B \times L}$ and select samples with the largest values at the top $\tau_2$ to form a pseudotarget dataset $X_T \in R^{T \times L}$. It is

noteworthy that the ratio of selecting pseudobackground pixels and pseudotarget pixels is not fixed, and for each specific dataset, there exists a suitable ratio range.

Due to factors such as imaging equipment, targets in HSI may possess spectra that differ. Moreover, the CEM algorithm can only maintain the output of prior spectral information $d$, which implies that targets that vary significantly from d cannot be detected, while background spectra that are similar to $d$ may be detected as targets. In other words, the labels of the pseudotraining dataset constructed by dividing the limited prior target spectra $d$ are incomplete and inaccurate. Therefore, the proposed method can be categorized as an inaccurate supervision method, which is a form of adaptive adversarial learning.

### B. Adversarial Learning on the Latent Space of Background

As mentioned above, our goal is to learn the accurate distribution of background latent space and to separate the background latent space vector from the target latent vector. To achieve this goal, we use three methods to ensure that the extracted background latent space information is accurate, complete, and distinguishable from the target latent space. First, we use an adversarial approach to make the distribution of the background latent space conform to a prior Gaussian distribution. Second, we use background reconstruction loss to measure the loss between the input spectrum and the reconstructed spectrum, ensuring that the background spectrum can be effectively reconstructed. Finally, we use the pseudotarget set suppression constraint to separate the latent targets from the pseudobackground data and ensure good reconstruction performance of the real background spectra.

*1) Background Latent Space Adversarial Learning:* AAE is a variant of GAN that regularizes the model during training. We assume that the latent space distribution of the real background conforms to a Gaussian distribution $p(z)$. We use the encoder E to extract the latent space vector $z$ of the background, making its distribution $q(z)$ conform to $p(z)$, and making the latent space vector $z$ able to be well restored to the background spectrum through the decoder De. The input of the discriminator Di either comes from the latent space vector extracted by the encoder or is a vector sampled from the prior distribution $p(z)$. The output of discriminator Di is a single scalar value, which can be interpreted as the probability that the input vector comes from the latent space vector encoded from the background data or from a sampled vector of the prior background latent space distribution. In the training process, minimizing $L_{\text{adv}}$ (9) ensures that the latent space distribution of the background is accurately extracted and conforms to the prior distribution.

*2) Background Spectral Reconstruction:* The background spectrum $b_i$ sampled from the pseudobackground dataset is encoded by the encoder to obtain its latent space vector $z_i$, which is then input to the decoder De to obtain the reconstructed background spectrum $b_i'$. In order to obtain an accurate reconstructed spectrum, we measure the reconstruction loss between the input sample $x$ and $De(E(b_i))$ by minimizing the MSE (Mean Squared Error), i.e., $L_R$(10). After training is completed, the latent space $z_i$ is obtained from the trained encoder, and then decoder reconstructs the background instances through the latent space. At the same time, instances with large residuals after reconstruction, that is, instances with high MSE values, can be considered as detected targets.

*3) Adaptive Inaccurate Sample Suppression Constraint Based on Pseudotarget Set:* As described above, the MSE between spectral vectors and their reconstructed versions can be used as a similarity metric for identifying targets. However, minimizing MSE alone can occasionally result in unrealistic reconstructions. In addition, due to the limitations of the coarse detector CEM, it is possible that target pixels may be misclassified as background pixels in order to construct a pseudobackground dataset $X_B$. To improve the discrimination between targets and backgrounds, we use an adaptive constraint based on Euclidean distance. The constraint fully utilizes the limited prior information $d$ and the pseudotarget dataset $X_T \in R^{T\times L}$.

The constraint maximizes the distance between the spectral vector bi in the pseudobackground dataset $X_b$ and the average target vector $\bar{t}'$ obtained from $d$ and $X_T$, which enhances the discrimination of spectral reconstruction by optimizing the loss function

$$L_T = -\sum_{i=1}^{B} \alpha_i \left\| b_i - \bar{t}' \right\|_2^2 \tag{14}$$

where $\alpha_i$ is the weight coefficient assigned to each training sample. $\alpha_i$ and $\bar{t}'$ can be expressed as

$$\alpha_i = 1 - \exp\left(-\gamma \left\| b_i - \bar{t}' \right\|^2\right) \tag{15}$$

$$\bar{t}' = -\frac{1}{2}\left(d + t'\right) = \frac{1}{2}\left(d + \left(\frac{\sum_{j=1}^{T} t_j}{T}\right)\right). \tag{16}$$

In the equation, $\gamma$ is a tunable parameter and $t_j \in X_T$. The value of $\gamma$ affects the optimization strength of the model when there is a change in the gap between $b_i$ and $\bar{t}'$. The curves of the function for different values of $\gamma$ are shown in Fig. 2.

When the distance between an input spectral vector $b_i$ and the target average vector $\bar{t}'$ is small, we can infer that this inaccurate sample bi is likely to be a target pixel rather than a background pixel. At the same time, the weighting factor $\alpha_i$ of the sample tends to be 0, suppressing the latent interference of inaccurate sample target spectra on background spectrum representation and reconstruction. Conversely, when the distance is large, $\alpha_i$ tends to be 1, maintaining the normal optimization of the model.

Therefore, the final loss function of the training process can be summarized as

$$L = L_{\text{adv}} + L_R + \beta L_T. \tag{17}$$

The parameter $\beta$ is used to balance the impact of the reconstruction authenticity and target distance constraints.

This objective ensures effective reconstruction of the background, while suppressing interference from targets during the training process, alleviating the problem of ineffective separation between background and targets when the supervision is weak.

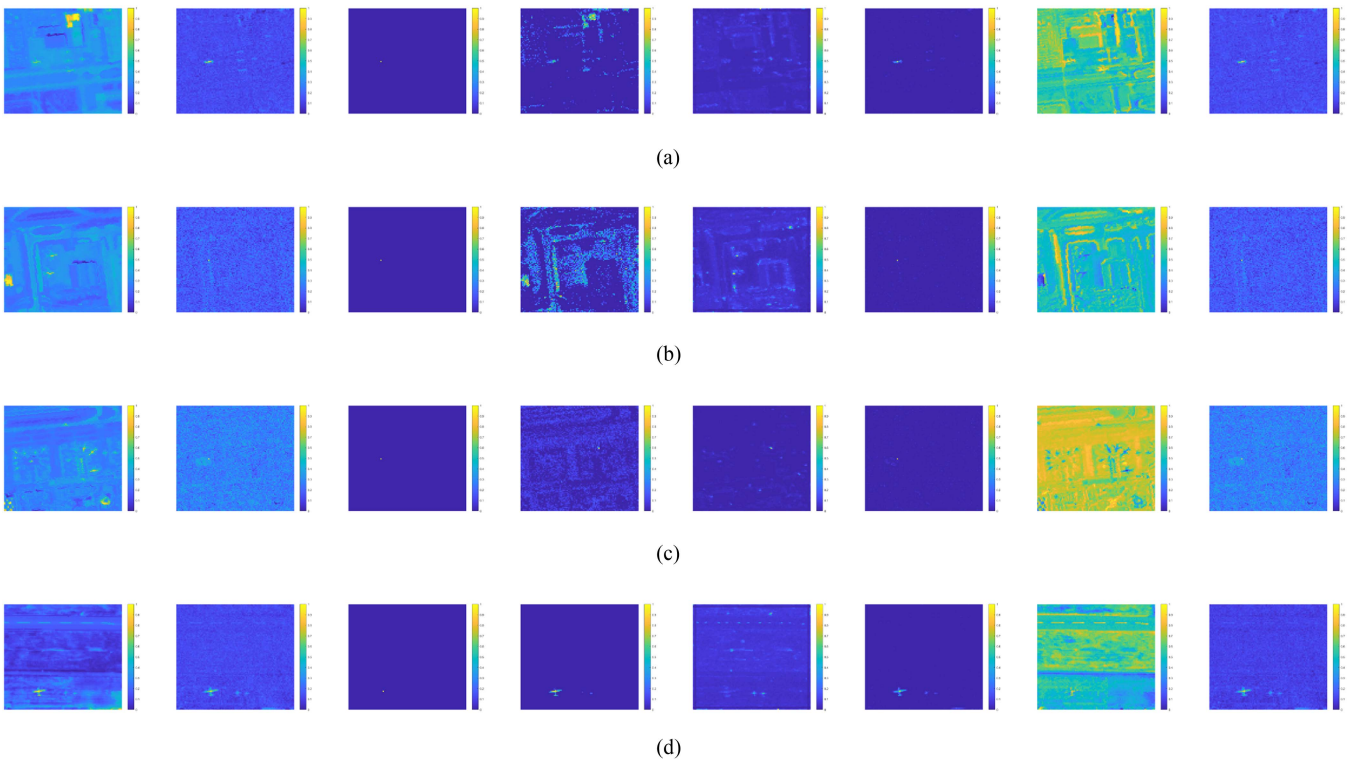



Fig. 11. Detection results of different comparing methods on ABU-Airport HSIs. (Left to right) OUR, CEM, ECEM, hCEM, CSCR, ACE, OSP, SMF. (a) ABU-Airport-1. (b) ABU-Airport-2. (c) ABU-Airport-3. (d) ABU-Airport-4.

### C. Target Detection

After the training is completed, as shown in Fig. 1(c), all spectral vectors $H \in R^{M\times N\times L}$ and prior target information $d$ in the original HSI are input into the trained model to obtain the similarity $L_{\text{sim}}$ between the latent space $H_L$ and the target latent space vector $d_L$ through (13), which represents the likelihood of each pixel in $H_L$ becoming a target. In addition, the reconstruction $H'$ and the $H$ obtained through (10) further illustrate the difference between targets and background. Therefore, our final detection result is

$$\text{Output} = L_{\text{sim}} * (1 - L_R). \tag{18}$$

We summarize the algorithm we proposed in Algorithm 1.

**Algorithm 1:** Adaptive Background Latent Space Adversarial Learning Algorithm for Hyperspectral Target Detection.

**Input** HSI $X$, prior known target spectrum $d$, threshold of dividing the pseudo-target set $\tau_1$, threshold of dividing the pseudo-background set $\tau_2$.
**Step:**
Obtain the pseudo-background set $\mathrm{X}_B$ and the pseudo-target set $\mathrm{X}_T$ using (6) and thresholds $\tau_1$ and $\tau_2$;
Establish the background latent space learning model;
**for each epoch do**
  minimize final loss $L_T$ in (17);
**end**
Reconstruct $X' = De(E(d))$, latent spase $X_L = E(X)$ with the trained model;
Latent spase $d_L = E(d)$ with the trained model;
Calculate the latent spase similarity $L_{sim}$ with (13);
Calculate the reconstruct distance $L_R$ with (10);
Detect target from $L_{sim}$ and $L_R$ with (18);
**Output** final detection result Output.

## IV. Experiments

In this section, we designed and executed experiments and conducted an analysis of the impact of parameter changes on the results. We utilized three publicly available hyperspectral datasets to evaluate the performance of our proposed detection method.

### A. Datasets Description

*1) The El Segundo Dataset:* The first column of Fig. 3 show the 30th band image and GT map in the El Segundo dataset. This dataset was captured by the AVIRIS sensor, which has 224 spectral channels ranging from 366 to 2496 nm. This urban scene covers the El Segundo area of California, USA, with a spatial size of 250 × 191 × 300. The ground resolution of each pixel is 7.1 m. The image dataset mainly consists of refinery areas, several residential areas, parks, and a school area. The building area of the refinery, such as oil tanks and towers, reaches 2048 pixels, and is therefore considered as attack targets.

*2) Cuprite Dataset:* The second column of Fig. 3 shows the 30th band image and GT map in the Cuprite dataset. This dataset

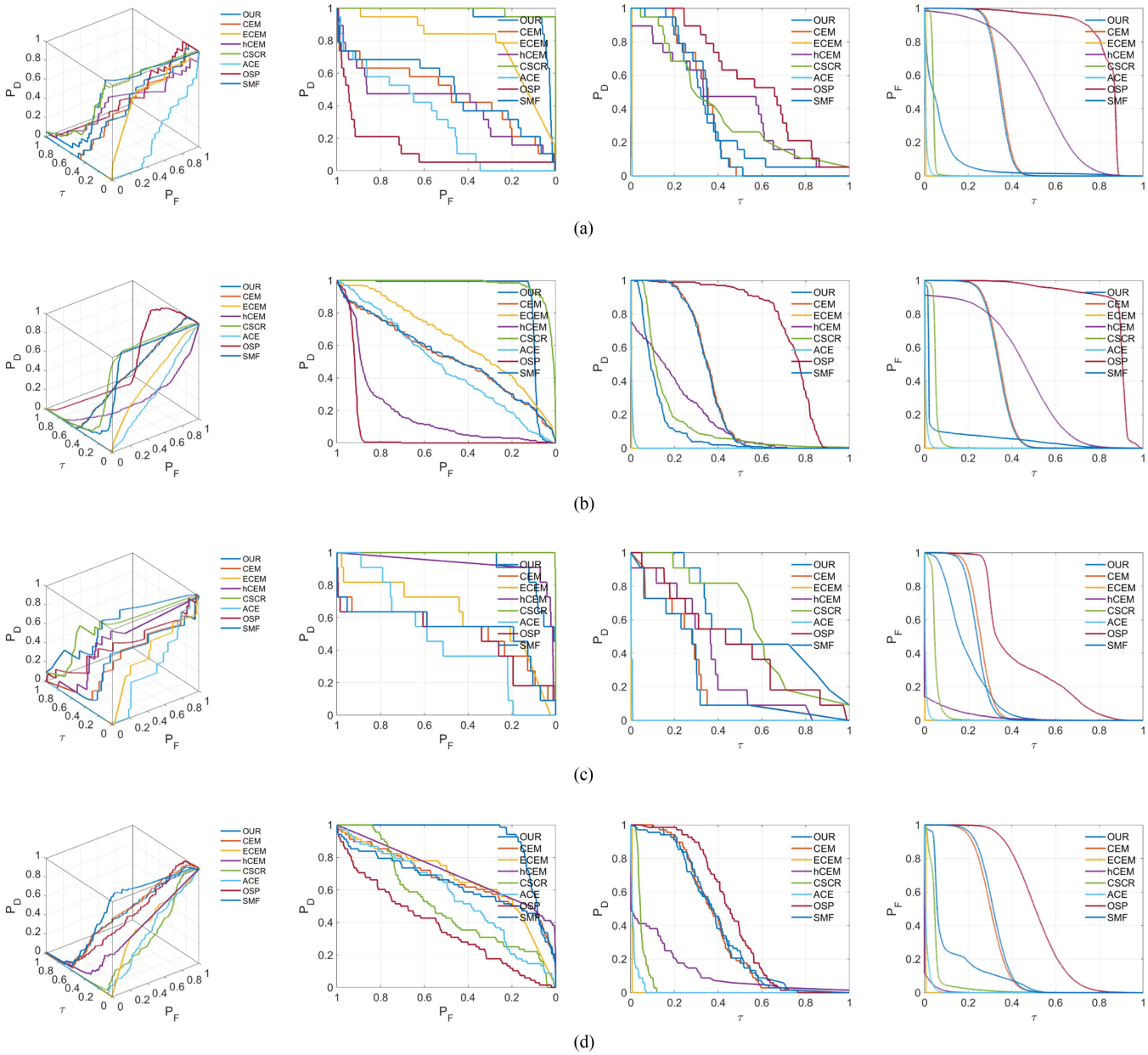



Fig. 12. 3D-ROC and three corresponding 2D-ROC curves of different comparing methods on ABU-Beach HSIs. (Left to right) 3D-ROC curve, 2D-ROC curve of $(P_d, P_f)$, 2-D ROC curves of $(P_d, \tau)$, 2-D ROC curves of $(P_f, \tau)$. (a) ABU-Beach-1. (b) ABU-Beach-2. (c) ABU-Beach-3. (d) ABU-Beach-4.

was collected by the AVIRIS sensor in the Cuprite mining area of Nevada, USA, in 1997, and contains approximately 14 types of minerals, including orthoclase, iron clay, kaolinite, and others. Here, we use a subset of the Cuprite data for experimentation, which consists of 250 $\times$ 191 pixels and 188 bands (excluding low signal-to-noise and water absorption bands). In this dataset, buddingtonite (orthoclase) is treated as the target to be detected. The Cuprite database was made available online: http://aviris.jpl.nasa.gov/html/aviris.freedata.html

*3) The ABU(Airport-Beach-Urban) Database:* The Figs. 4–6 show the 30th band images and GT maps in the ABU dataset. The database was obtained in detail in [31]. These sample images contain 100 $\times$ 100 or 150 $\times$ 150 pixels, and have corresponding references. The airplane and urban scenes were captured by the AVIRIS sensor, and most of the beach scenes were also captured by the AVIRIS sensor. One of the beach scenes was captured by the Reflective Optics System Imaging Spectrometer (ROSSI-03) sensor. The ABU database was made available online: http://xudongkang.weebly.com/

In the case of the ABU dataset, which is frequently utilized for evaluating anomaly detection models [31], there is no readily available prior target information $d$. To overcome this challenge and provide prior information for HTD, target pixels were randomly selected from the Ground Truth (GT) and designated as the prior information $d$. To provide transparency and facilitate further analysis, the coordinates of these selected target pixels

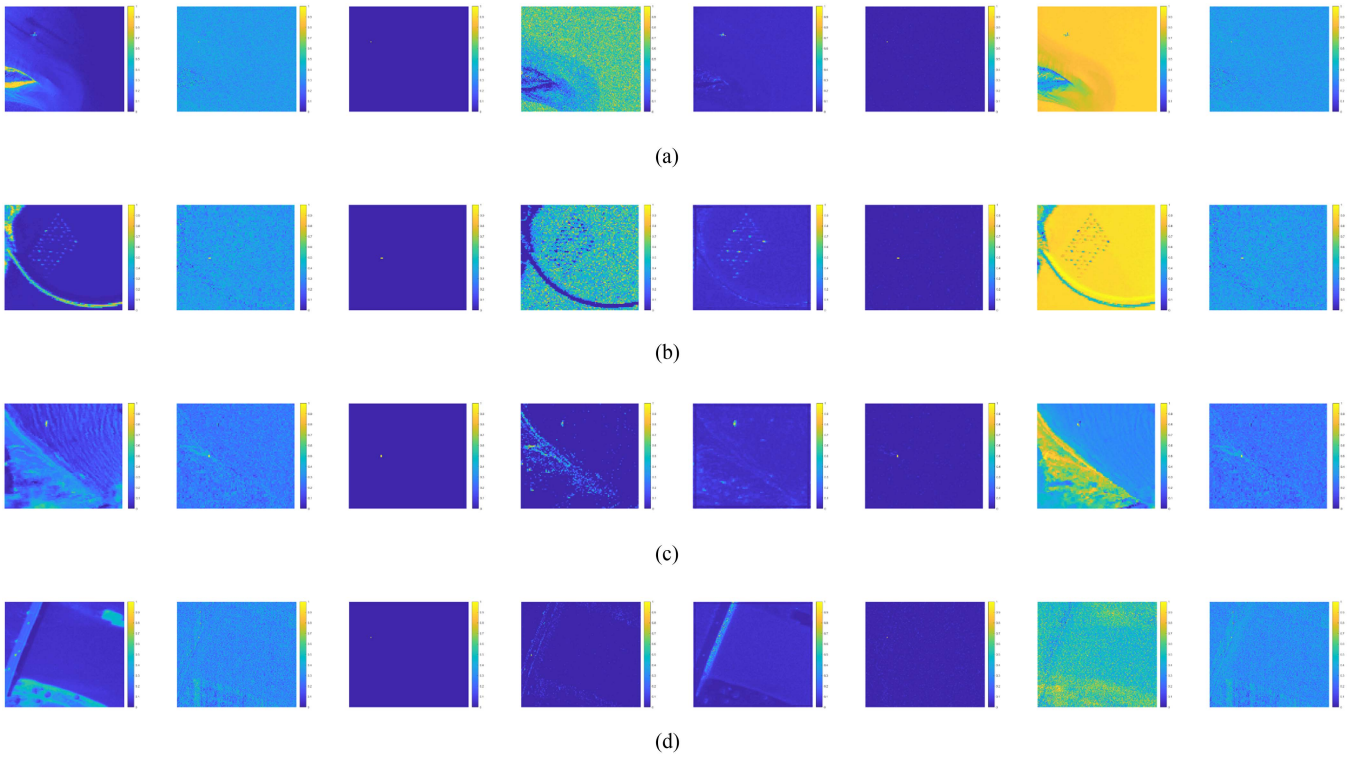



Fig. 13. Detection results of different comparing methods on ABU-Beach HSIs. (Left to right) OUR, CEM, ECEM, hCEM, CSCR, ACE, OSP, SMF. (a) ABU-Beach-1. (b) ABU-Beach-2. (c) ABU-Beach-3. (d) ABU-Beach-4.

can be found in Table I. In addition, Table I provides additional information such as the size of the targets and the number of targets present in each dataset.

This section utilizes seven commonly used or performance-leading HTD methods, as described in literature. These methods include ACE [7], SMF [8], [32], hCEM [9], E-CEM [10], OSP [13], [33], and CSCR [14]. ACE, OSP, and CEM are traditional HTD algorithms. In contrast, hCEM, E-CEM, and CSCR are popular extensions of CEM that have been widely applied, with CSCR being a representative based on ML models.

Chang [34] demonstrated that the traditional 2-D ROC curve obtained by $P_D$ and $P_F$ is inadequate or ineffective for comprehensive performance evaluation of detectors. This is because both $P_D$ and $P_F$ are determined by the same threshold, which can only be used to assess the effectiveness of detectors and not target detectability (TD) and also not background suppressibility (BS). Therefore, Chang introduced a three-dimensional ROC curve and derived three 2-D ROC curves: the 2-D ROC curve ($P_D$, $P_F$), the 2-D ROC curve ($P_D$, $\tau$), and the 2-D ROC curve ($P_F$, $\tau$) [34]. In this article, we will adopt this approach to evaluate the performance of our detector.

It is worth noting that for the 2-D ROC curve ($P_D$, $P_F$), a larger value of $\text{AUC}_{(D,F)}$ indicates a better detector. Similarly, for the 2-D ROC curve ($P_D$, $\tau$), a larger value of $\text{AUC}_{(D,\tau)}$ indicates a better detector. Conversely, for the 2-D ROC curve ($P_F$, $\tau$), a smaller value of $\text{AUC}_{(F,\tau)}$ indicates a better detector. Therefore, we will use the $\text{AUC}_{OD}$ from [34] as a comprehensive measure to evaluate the performance of our detector.

$$\text{AUC}_{OD} = \text{AUC}_{(D,F)} + \text{AUC}_{(D,\tau)} - \text{AUC}_{(F,\tau)}. \tag{19}$$

The code we used is provided by [34] and can be found at the following online address: https://wiki.umbc.edu/display/rssipl/10.+Download

### B. Parameter Sensitivity Analysis

As mentioned earlier, the dimensionality of the latent space refers to the number of nodes in the deepest encoding layer. This parameter determines the quality of the learned background spectral distribution features. Setting it too small would result in insufficient latent space information to reconstruct the background spectra and accurately discriminate between the backgrounds and targets in the latent space. On the other hand, setting it too large would lead to inadequate dimensionality reduction, potentially including irrelevant information that could affect reconstruction and differentiation between targets and backgrounds.

Fig. 7(a) illustrates the effect of varying latent node numbers on different datasets. Except for Cuprite, where the detection performance decreases as the number of latent nodes increases, the remaining datasets generally exhibit an increasing-then-decreasing trend. The best detection results are achieved when the latent nodes are set to 10.

During the training process, the parameter $\beta$ balances the influence of reconstruction fidelity and target distance constraints. If it is set too large, the target constraint will affect the normal reconstruction of background spectra, which is unfavorable for detection. Conversely, if it is set too small, the target distance constraint will not be effective, resulting in target reconstruction.

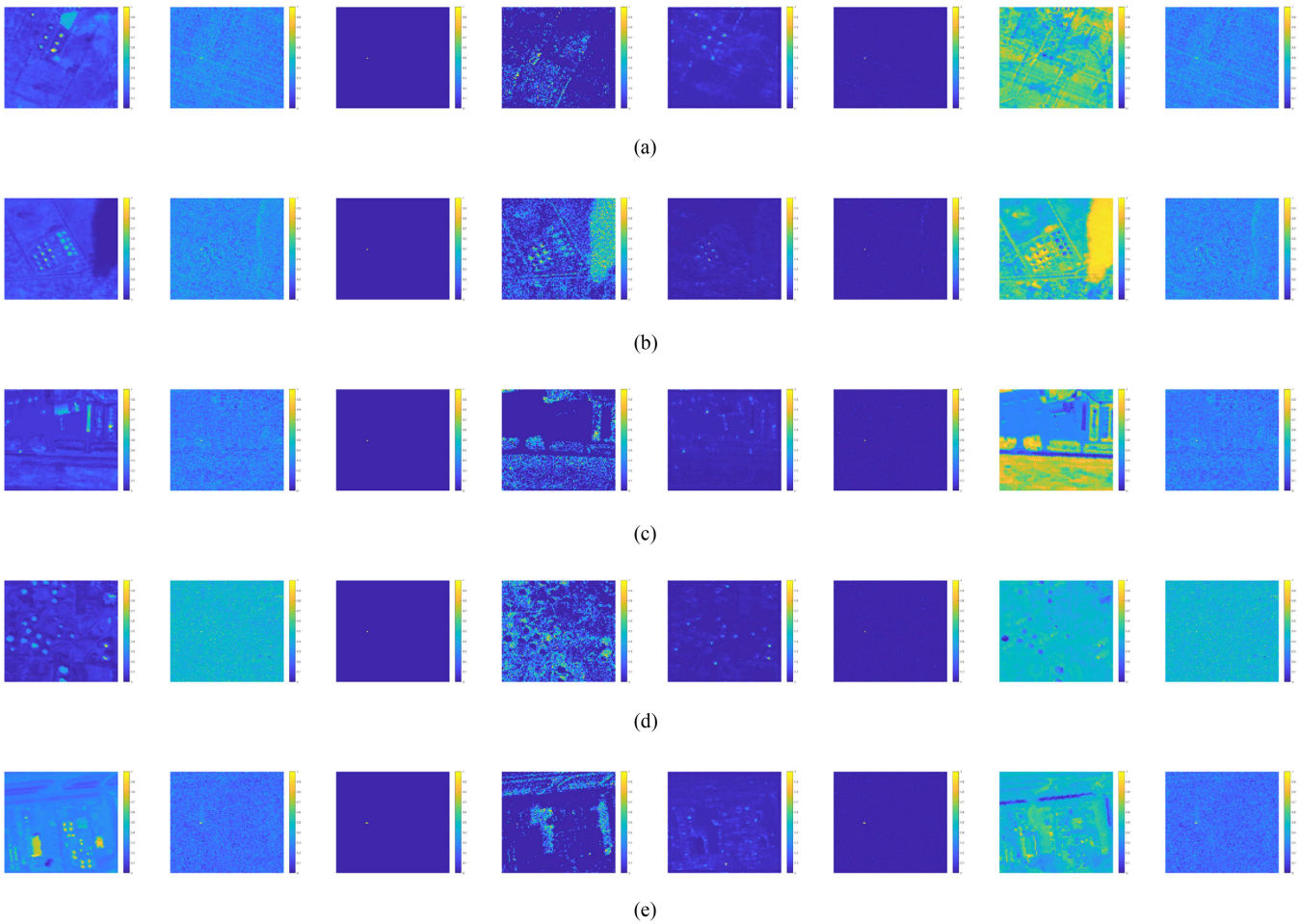



Fig. 14. Detection results of different comparing methods on ABU-Urban HSIs. (Left to right) OUR, CEM, ECEM, hCEM, CSCR, ACE, OSP, SMF. (a) ABU-Urban-1. (b) ABU-Urban-2. (c) ABU-Urban-3. (d) ABU-Urban-4. (e) ABU-Urban-5.

Fig. 7(b) illustrates the effects of different $\beta$ values on various datasets. Except for ABU-Beach, the remaining datasets exhibit the best performance when $\beta$ is set to 0.1. The anomaly observed in ABU-Beach may be due to inaccurate prior target $d$ selection, which was confirmed by the subsequent experimental detection results.

The parameter $\gamma$ determines the distance constraint between targets and backgrounds and is responsible for distinguishing inaccurate samples in the pseudobackground dataset. If $\gamma$ is set too small, it will not effectively differentiate inaccurate samples in the pseudobackground dataset. Conversely, if it is set too large, accurate samples may be identified as inaccurate samples.

As shown in Fig. 7(c), the best detection results are achieved when $\gamma$ is set to 50.

Regarding the selection thresholds for the pseudobackground dataset ($\tau_1$) and pseudotarget dataset ($\tau_2$), they determine whether the background information is sufficiently selected and whether the pseudotargets are accurate. If $\tau_1$ is set too small, it may not include all the background information, while setting it too large may result in including target information in the pseudobackground dataset. The setting of $\tau_2$ depends on the number of targets in the dataset. If the targets are sparse, $\tau_2$ should be set as large as possible to ensure the accuracy of the pseudotarget dataset. Appropriately, setting these parameters can yield good detection performance. Fig. 7 illustrates the impact of these parameters on the detection AUC values. In general, using 10 latent space nodes, a $\beta$ value of 0.1, a $\gamma$ value of 50, a $\tau_1$ value of 0.5, and a $\tau_2$ value of 0.9994 perform well on most datasets. However, the number of targets varies across different datasets. For example, in datasets with fewer targets like ABU Beach and Cuprite, better detection results can be achieved when $\tau_2$ is set to a higher value.

### C. Component Analysis

Table II displays the AUC values for the two components of the final detection results and the overall detection performance. In the first part of our method, we use (13) to calculate the dissimilarity between targets and backgrounds in the latent space. By reducing the dimensionality of the original spectral data, we obtain discriminative information, which can be seen as target detection. In the second part, the reconstruction information from the AAE is utilized to determine whether the input data is from the background, thus acting as a background suppression.

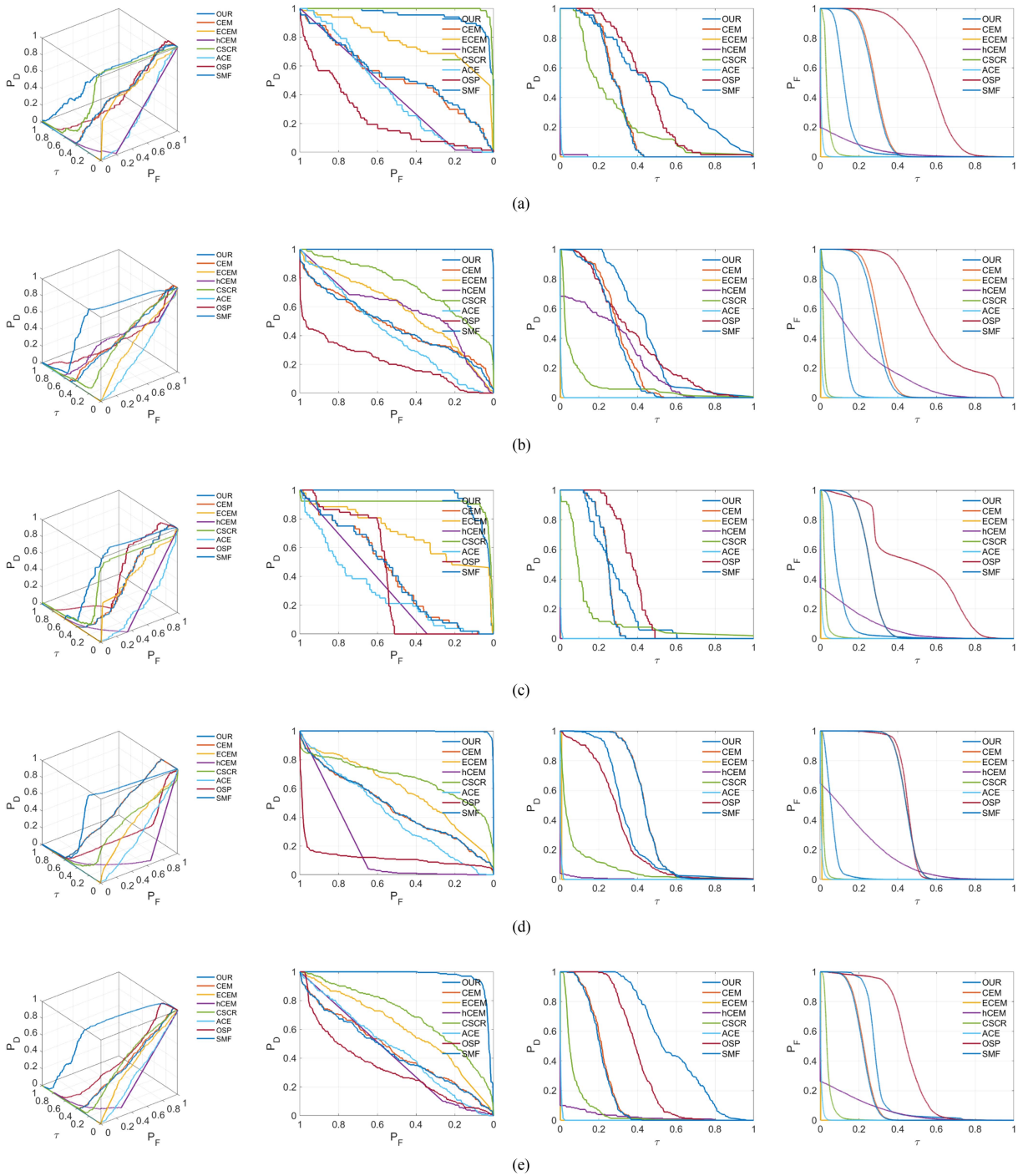



Fig. 15. 3D-ROC and three corresponding 2D-ROC curves of different comparing methods on ABU-Urban HSIs. (Left to right) 3D-ROC curve, 2D-ROC curve of $(P_d, P_f)$, 2-D ROC curves of $(P_d, \tau)$, 2-D ROC curves of $(P_f, \tau)$. (a) ABU-Urban-1. (b) ABU-Urban-2. (c) ABU-Urban-3. (d) ABU-Urban-4. (e) ABU-Urban-5.

While the two parts of the detection results may be superior to the final detection result in specific cases, as demonstrated above, they may be unstable. On the other hand, combining the two results produces a more stable final output.

### D. Detection Result

Tables III–VI present the AUC values and $AUC_{\text{OD}}$ of our method and other comparison methods on different HSIs across three different 2D-ROC curves. Figs. 8–14 display the corresponding 3D-ROC curves, three corresponding 2D-ROC curves, and detection results.

*1) El Segundo and Cuprite:* On El Segundo, although our algorithm surpasses other algorithms in terms of $AUC_{(D,F)}$, it does not exhibit superior performance in terms of $AUC_{(D,\tau)}$ compared to OSP. Moreover, it does not outperform hCEM in terms of $AUC_{(F,\tau)}$ and $AUC_{OD}$, as hCEM effectively suppresses the background through multiple detections, thereby

highlighting the targets. In this complex background scenario, our algorithm fails to effectively suppress the background, resulting in inadequate detection results.

On Cuprite, our algorithm outperforms other algorithms in terms of $AUC_{(D,F)}$ and $AUC_{OD}$. However, it does not perform as well as ECEM in terms of $AUC_{(D,\tau)}$. Nonetheless, from Fig. 9, we observe that its detection performance is not satisfactory, as it exhibits the worst performance in terms of $AUC_{(F,\tau)}$.

*2) Abu-Airport:* On ABU-Airport, our algorithm outperforms other algorithms in terms of $AUC_{OD}$. However, it does not perform as well as OSP in terms of $AUC_{(D,\tau)}$, whereas ECEM performs better in terms of $AUC_{(F,\tau)}$. Particularly, in Fig. 11(a) and (c), although our initial algorithm CEM does not perform well in delineating the pseudotarget set, we still achieved good detection results.

*3) Abu-Beach:* Although our detection algorithm outperforms the best-performing algorithm CSCR on ABU-Beach-4, it does not surpass CSCR's performance on other datasets. Due to the relatively simple background information of ABU Beach 1-3, the dictionary construction information obtained by CSCR happens to have good target detection ability. It is worth noting that in terms of overall performance, our algorithm is relatively superior to other algorithms on this dataset, except for CSCR.

*4) Abu-Urban:* On ABU-Urban, our algorithm outperforms other algorithms in terms of $AUC_{OD}$. However, it does not surpass ECEM in terms of $AUC_{(F,\tau)}$. In terms of $AUC_{(D,F)}$, our algorithm is the best among other HSIs except for ABU-Urban-1, where it does not surpass CSCR.

Although the proposed algorithm in this article does not surpass ECEM and hCEM in terms of the evaluation metric $AUC_{(F,\tau)}$, these variants of the CEM algorithm exhibit high capability in background suppression, especially in complex background scenarios. However, compared to our methods on other datasets, they lack stability.

CSCR performs better than other algorithms in small target detection and inaccurate prior target spectra tasks, and it outperforms our proposed method on ABU-Beach-1, ABU-Beach-2, ABU-Beach-3, and ABU-Urban-1. However, compared to these methods, our proposed method shows robust detection performance for targets of different scales in all scenarios. Additionally, the AUC values of the three 2-DROC curves and the joint result of the three AUC values obtained support the observation on the El Segundo, Cuprite, and ABU datasets. Therefore, it can be concluded that the proposed method achieves excellent performance and high stability on most datasets.

In addition, the computation times of different methods are shown in Table VII. All experiments were conducted on a machine with an Intel Core(TM) i9-10850 K CPU and 32 GB RAM. Our proposed method and ECEM were implemented using Python 3.8.0 and TensorFlow 2.2, while the comparison methods were implemented using MATLAB R2019a. The runtime of detection was measured after the training phase and recorded in seconds, as modeling the background latent space takes several minutes due to the introduction of the network. Table VII demonstrates that our model has lower complexity compared to CSCR and ECEM. Although CEM has a shorter runtime, its detection performance is inferior to our method. Our proposed method achieves excellent detection performance with a tolerable runtime. We have also computed the time complexity of all the compared methods and summarized them in Table VIII. The input image size for all methods is $M \times N \times L$. In our method, $b$ denotes the number of nodes in the latent space, while $iter$ represents the number of layers in hCEM, which is on average 10. For CSCR, $(w_out, w_in)$ form the sliding double windows and are set to (11, 3). The values of $ncem$ and $nlayer$ represent the number of CEM detectors used per layer and the number of layers in ECEM, respectively, with values of 6 and 10. Our method, hCEM, ACE, OSP, and SMF have similar time complexities, with hCEM achieving better performance than CEM at the cost of higher computation time. CSCR, on the other hand, has significantly higher time complexity compared to our method, although it may outperform our method on certain datasets. These results are consistent with the computation times listed in Table VII.

## V. Conclusion

In this article, we propose a novel approach for HTD by leveraging adaptive adversarial learning techniques in the background latent space. Our method incorporates an AAE to extract the background data distribution from high spectral images and capture essential spectral information in the latent space. Through the process of acquiring the spectral distribution, we utilize adversarial learning to align the latent space distribution of the background spectra with a prior distribution. This adversarial learning step helps bridge the gap between the background latent space and the desired prior distribution. To enhance the accuracy of target reconstruction and effectively suppress background spectra, we introduce a pseudotarget set that restricts the framework's ability to fully reconstruct targets. By constraining the reconstruction capabilities, our method achieves more precise target reconstruction and better suppression of background spectra. Our experimental results demonstrate that our proposed method outperforms existing approaches, especially when dealing with smaller datasets. We conducted experiments on three real hyperspectral datasets, and the results highlight the effectiveness of our approach.

## References

[1] Y. Hou, W. Zhu, and E. Wang, "Hyperspectral mineral target detection based on density peak," *Intell. Automat. Soft Comput.*, vol. 25, no. 4, pp. 805–814, Dec. 2019.

[2] R. Martin, B. Thies, and A. O. Gerstner, "Hyperspectral hybrid method classification for detecting altered mucosa of the human larynx," *Int. J. Health Geographics*, vol. 11, pp. 21–29, 2012. [Online]. Available: https: MEDLINE:22720905

[3] F. Mei, C. Li, and Y. Zhang, "Application of spectral imaging technology in maritime target detection," *Chin. Opt.*, vol. 10, no. 6, pp. 708–718, 2017. [Online]. Available: https:CSCD:6131854

[4] R. M. Kudela, S. L. Palacios, D. C. Austerberry, E. K. Accorsi, L. S. Guild, and J. Torres-Perez, "Application of hyperspectral remote sensing to cyanobacterial blooms in inland waters," *Remote Sens. Environ.*, vol. 167, pp. 196–205, 2015. [Online]. Available: https:WOS:000360510800017

[5] D. Manolakis and G. Shaw, "Detection algorithms for hyperspectral imaging applications," *IEEE Signal Process. Mag.*, vol. 19, no. 1, pp. 29–43, Jan. 2002.

[6] C.-I. Chang, "An information-theoretic approach to spectral variability, similarity, and discrimination for hyperspectral image analysis," *IEEE Trans. Inf. Theory*, vol. 46, no. 5, pp. 1927–1932, Aug. 2000, doi: 10.1109/18.857802.
[7] S. Kraut, L. Scharf, and L. McWhorter, "Adaptive subspace detectors," *IEEE Trans. Signal Process.*, vol. 49, no. 1, pp. 1–16, Jan. 2001.
[8] F. C. Robey, D. R. Fuhrmann, E. J. Kelly, and R. Nitzberg, "A CFAR adaptive matched filter detector," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 28, no. 1, pp. 208–216, Jan. 1992, doi: 10.1109/7.135446.
[9] 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.
[10] 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, 2019, Art. no. 1310. [Online]. Available: https:www.mdpi.com/2072-4292/11/11/1310
[11] J. Chen and C.-I. Chang, "Background-annihilated target-constrained interference-minimized filter (TCIMF) for hyperspectral target detection," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Sep. 2022, Art. no. 5540224.
[12] X. Yang, M. Zhao, S. Shi, and J. Chen, "Deep constrained energy minimization for hyperspectral target detection," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 15, pp. 8049–8063, Sep. 2022.
[13] L. Capobianco, A. Garzelli, and G. Camps-Valls, "Target detection with semisupervised kernel orthogonal subspace projection," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 11, pp. 3822–3833, Nov. 2009, doi: 10.1109/TGRS.2009.2020910.
[14] 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, 2015. [Online]. Available: https:www.sciencedirect.com/science/article/pii/S0031320315002034
[15] Y. Li, Y. Shi, K. Wang, B. Xi, J. Li, and P. Gamba, "Target detection with unconstrained linear mixture model and hierarchical denoising autoencoder in hyperspectral imagery," *IEEE Trans. Image Process.*, vol. 31, pp. 1418–1432, Jan. 2022.
[16] Y. Gao et al., "Hyperspectral and multispectral classification for coastal wetland using depthwise feature interaction network," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5512615.
[17] Y. Gao, M. Zhang, J. Wang, and W. Li, "Cross-scale mixing attention for multisource remote sensing data fusion and classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, 2023, Art. no. 5507815.
[18] C. Yu, R. Han, M. Song, C. Liu, and C.-I. Chang, "Feedback attention-based dense CNN for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, 2022, Art. no. 5501916.
[19] M. Zhang, W. Li, X. Zhao, H. Liu, R. Tao, and Q. Du, "Morphological transformation and spatial-logical aggregation for tree species classification using hyperspectral imagery," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Jan. 2023, Art. no. 5501212.
[20] C. Yu, B. Gong, M. Song, E. Zhao, and C.-I. Chang, "Multiview calibrated prototype learning for few-shot hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 60, Dec. 2022, Art. no. 5544713.
[21] Y. Shi, J. Lei, Y. Yin, K. Cao, Y. Li, and C.-I. Chang, "Discriminative feature learning with distance constrained stacked sparse autoencoder for hyperspectral target detection," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 9, pp. 1462–1466, Sep. 2019.
[22] L. Zhu, Y. Chen, P. Ghamisi, and J. A. Benediktsson, "Generative adversarial networks for hyperspectral image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 56, no. 9, pp. 5046–5063, Sep. 2018.
[23] W. Xie, J. Lei, S. Fang, Y. Li, X. Jia, and M. Li, "Dual feature extraction network for hyperspectral image analysis," *Pattern Recognit.*, vol. 118, 2021, Art. no. 107992. [Online]. Available: https:www.sciencedirect.com/science/article/pii/S0031320321001795
[24] Y. Gao, M. Zhang, W. Li, X. Song, X. Jiang, and Y. Ma, "Adversarial complementary learning for multisource remote sensing classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 61, Mar. 2023, Art. no. 5505613.
[25] W. Xie, J. Zhang, J. Lei, Y. Li, and X. Jia, "Self-spectral learning with GAN based spectral–spatial target detection for hyperspectral image," *Neural Netw.*, vol. 142, pp. 375–387, 2021. [Online]. Available: https:www.sciencedirect.com/science/article/pii/S0893608021002252
[26] 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. Topics Appl. Earth Observ. Remote Sens.*, vol. 13, pp. 5887–5897, Sep. 2020.
[27] A. Makhzani, J. Shlens, N. Jaitly, and I. J. Goodfellow, "Adversarial autoencoders," 2015, doi: 10.48550/arXiv.1511.05644.
[28] U. Michelucci, "An introduction to autoencoders," 2022, doi: 10.48550/arXiv.2201.03898.
[29] F. Kruse et al., "The spectral image processing system (sips)–interactive visualization and analysis of imaging spectrometer data," *Remote Sens. Environ.*, vol. 44, no. 2, pp. 145–163, 1993. [Online]. Available: https:www.sciencedirect.com/science/article/pii/003442579390013N
[30] H. Qin, W. Xie, Y. Li, K. Jiang, J. Lei, and Q. Du, "Weakly supervised adversarial learning via latent space for hyperspectral target detection," *Pattern Recognit.*, vol. 135, 2023, Art. no. 109125. [Online]. Available: https:www.sciencedirect.com/science/article/pii/S0031320322006057
[31] 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.
[32] J. C. Harsanyi, "Detection and classification of subpixel spectral signatures in hyperspectral image sequences." Ph.D. dissertation, University of Maryland Baltimore County, 1993.
[33] J. C. Harsanyi and C. I. Chang, "Hyperspectral image classification and dimensionality reduction: An orthogonal subspace projection approach," *IEEE Trans. Geosci. Remote Sens.*, vol. 32, no. 4, pp. 779–785, Jul. 1994.
[34] C.-I. Chang, "An effective evaluation tool for hyperspectral target detection: 3 D receiver operating characteristic curve analysis," *IEEE Trans. Geosci. Remote Sens.*, vol. 59, no. 6, pp. 5131–5153, Jun. 2021.

**Long Sun** received the B.S. degree in information management and information systems from the Henan Polytechnic University, Jiaozuo, China, in 2021. He is currently working toward the M.S. degree in computer science and technology from the College of Information and Control Engineering, Xi'an University of Architecture and Technology, Xi'an, China.

His research interests include machine learning and hyperspectral target detection.

**Zongfang Ma** was born in Anhui, China, in 1980. He received the bachelor's degree in communication engineering and master's degree in computer application technology from the Xi'an University of Architecture and Technology, Xi'an, China, in 2002 and 2006, respectively, and the Ph.D. degree in pattern recognition and intelligent systems from Northwestern Polytechnical University, Xi'an, China, in 2011.

He is currently a Professor with the School of Information and Control Engineering, XAUAT. His current research interests include machine vision and pattern recognition.

**Yi Zhang** (Member, IEEE) received the B.S. degree in communication engineering, in 2002, the M.S. degree in computer application, in 2009, and Ph.D. in digital architecture, in 2016, all from Xi'an University of Architecture and Technology, Xi'an, China.

He is currently a Teacher with the School of Information and Control Engineering, Xi'an University of Architecture and Technology. His current research interests include image analysis and understanding, data mining, and intelligent computing.